

On the Theories Behind Hard Negative Sampling for Recommendation

Wentao Shi
shiwentao123@mail.ustc.edu.cn
University of Science and
Technology of China
Hefei, China

Jiawei Chen*
sleepyhunt@zju.edu.cn
Zhejiang University
Hangzhou, China

Fuli Feng
fulifeng93@gmail.com
University of Science and
Technology of China
Hefei, China

Jizhi Zhang
cdzhangjizhi@mail.ustc.edu.cn
University of Science and
Technology of China
Hefei, China

Junkang Wu
jkwu0909@gmail.com
University of Science and
Technology of China
Hefei, China

Chongming Gao
chongming.gao@gmail.com
University of Science and
Technology of China
Hefei, China

Xiangnan He*
xiangnanhe@gmail.com
University of Science and
Technology of China
Hefei, China

ABSTRACT

Negative sampling has been heavily used to train recommender models on large-scale data, wherein sampling hard examples usually not only accelerates the convergence but also improves the model accuracy. Nevertheless, the reasons for the effectiveness of Hard Negative Sampling (HNS) have not been revealed yet. In this work, we fill the research gap by conducting thorough theoretical analyses on HNS. Firstly, we prove that employing HNS on the Bayesian Personalized Ranking (BPR) learner is equivalent to optimizing One-way Partial AUC (OPAUC). Concretely, the BPR equipped with Dynamic Negative Sampling (DNS) is an exact estimator, while with softmax-based sampling is a soft estimator. Secondly, we prove that OPAUC has a stronger connection with Top- K evaluation metrics than AUC and verify it with simulation experiments. These analyses establish the theoretical foundation of HNS in optimizing Top- K recommendation performance for the first time. On these bases, we offer two insightful guidelines for effective usage of HNS: 1) the sampling hardness should be controllable, e.g., via pre-defined hyper-parameters, to adapt to different Top- K metrics and datasets; 2) the smaller the K we emphasize in Top- K evaluation metrics, the harder the negative samples we should draw. Extensive experiments on three real-world benchmarks verify the two guidelines.

CCS CONCEPTS

• Information systems → Recommender systems.

KEYWORDS

One-way Partial AUC, Negative Sampling, Distributionally Robust Optimization, Implicit Feedback, Recommender Systems

*Corresponding author

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.

WWW '23, May 1–5, 2023, Austin, TX, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9416-1/23/04...\$15.00
<https://doi.org/10.1145/3543507.3583223>

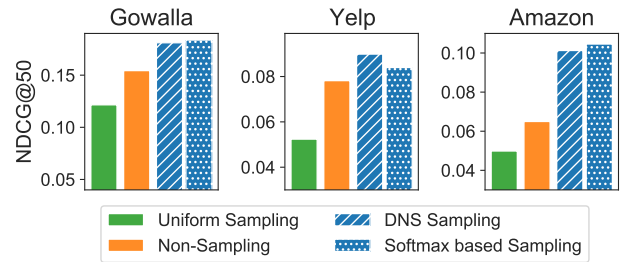


Figure 1: The recommendation performance on three widely used datasets with different sampling strategies.

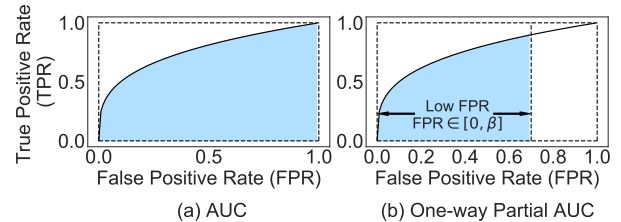


Figure 2: (a) AUC, measures the entire area of the ROC curve. (b) OPAUC, measures partial area within an FPR range of $[0, \beta]$. AUC is a special case of OPAUC with $\beta = 1$.

1 INTRODUCTION

Recommendation systems are essential in addressing information overload by filtering unintended information and have benefited many high-tech companies [7]. Bayesian Personalized Ranking (BPR) [29] is a common choice for learning recommender models from implicit feedback, which randomly draws negative items for the sake of efficiency and approximately optimizes the AUC metric. However, uniformly sampled negative items may not be informative, contributing little to the gradients and the convergence [28, 40]. To overcome this obstacle, researchers have proposed many Hard Negative Sampling (HNS) methods, such as Dynamic Negative Sampling (DNS) [40] and Softmax-based Sampling methods [9, 21, 33]. Superior to uniform sampling, HNS methods oversample

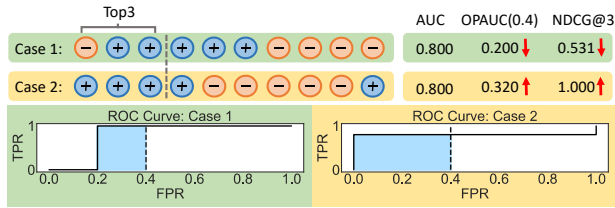


Figure 3: Two simple cases have the same overall ranking performance but different top-ranking performance. The ROC curves of two cases have the same AUC but different OPAUC($\beta=0.4$).

high-scored negative items, which are more informative with large gradients and thus accelerate the convergence [8].

While existing work usually attributes the superior performance of HNS to its better convergence, we find that the merits of HNS are beyond this thought. To validate it, we conduct empirical analyses on three widely used datasets in Figure 1. We compare two HNS strategies with a strong baseline named Non-Sampling¹ [4] that computes the gradient over the whole data (including all negative items). As such, the Non-Sampling strategy is supposed to converge to a better optimum more stably [1, 5, 6, 17]. Nevertheless, to our surprise, both HNS strategies substantially outperform the Non-Sampling strategy. It indicates that fast convergence may not be the only justification for the impressive performance of HNS. There must be other reasons for its superior performance, which motivates us to delve into HNS and explore its theoretical foundation. Our findings are twofold:

- **Optimizing the BPR loss equipped with HNS is equivalent to optimizing the One-way Partial AUC (OPAUC)**, whereas the original BPR loss only optimizes AUC. OPAUC puts a restriction on the range of false positive rate (FPR) $\in [0, \beta]$ [13], as shown in Figure 2(b), which emphasizes the ranking of top-ranked negative items. In contrast, AUC is a special case of OPAUC(β) with $\beta = 1$, which considers the whole ranking list. Our proof of the equivalence is based on the *Distributionally Robust Optimization* (DRO) framework [27] (cf. Section 3).
- **Compared to AUC, OPAUC has a stronger connection with Top-K metrics.** To illustrate it, we conduct simulation studies with randomly generated ranking lists, showing that OPAUC exhibits a much higher correlation with Top-K metrics like Recall, Precision and NDCG by tuning β (cf. Figure 6). This is because both OPAUC and Top-K metrics care more about the ranking of top-ranked items, as shown in Figure 3. Furthermore, we confirm the correlation through theoretical analysis that Recall@K and Precision@K metrics could be higher and lower bounded with a function of specific OPAUC(β), respectively.

In short, our analyses reveal that equipping BPR with HNS is equivalent to optimizing the OPAUC, leading to better Top-K recommendation performance (cf. Figure 4). Our analyses not only explain the impressive performance of HNS but also shed light on how to perform HNS in recommendation. Given the correspondence between Top-K evaluation metrics and OPAUC(β), we offer two instructive guidelines to ensure the practical effectiveness of

¹All compared methods optimize the same loss of BPR.

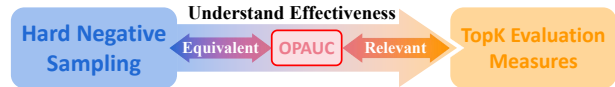


Figure 4: The relationship among HNS, OPAUC measure, and Top-K evaluation metrics.

HNS. First, the sampling hardness should be controllable, e.g., via pre-defined hyper-parameters, to adapt to different Top-K metrics and datasets. Second, the smaller the K we emphasize in Top-K evaluation metrics, the harder the negative samples we should draw.

The main contributions of this paper are summarized as follows:

- We are the first to establish the theoretical foundations for HNS: equipping BPR with DNS is an exact estimator of the OPAUC objective, and with softmax-based sampling is a soft estimator.
- We conduct theoretical analyses, simulation studies, and real-world experiments, to justify the connection between OPAUC and Top-K metrics and explain the performance gain of HNS.
- We provide two crucial guidelines on how to perform HNS and adjust sampling hardness. The experiments on real-world datasets validate the rationality of the guidelines.

2 BACKGROUND

This section provides the necessary background of Implicit Feedback, Hard Negative Sampling Strategies, One-way Partial Area Under ROC Curve (OPAUC), and Distributionally Robust Optimization (DRO) [27]. DRO is a robust learning framework that we will use in subsequent sections.

2.1 Implicit Feedback

The goal of a recommender is to learn a score function $r(c, i|\theta)$ to predict scores of unobserved item i in context c and recommend the top-ranked items [1]. A larger predicted score reflects a higher preference for the item $i \in \mathcal{I}$ in a context $c \in \mathcal{C}^2$. In the implicit feedback setting, we can only observe positive class $\mathcal{I}_c^+ \subseteq \mathcal{I}$ in the context c . The remaining $\mathcal{I}_c^- = \mathcal{I} \setminus \mathcal{I}_c^+$ are usually considered as negative items in the context c . In personalized ranking algorithms with BPR loss, the objective functions can be formulated as follows:

$$\min_{\theta} \sum_{c \in \mathcal{C}} \sum_{i \in \mathcal{I}_c^+} E_{j \sim P_{ns}(j|c)} [\ell(r(c, i|\theta) - r(c, j|\theta))], \quad (1)$$

where θ are the model parameters, $\ell(t)$ is the loss function which is often defined as $\log(1 + \exp(-t))$. $P_{ns}(j|c)$ denotes the negative sampling probability that a negative item $j \in \mathcal{I}_c^-$ in the context c is drawn. In BPR [29], each negative item is assigned an equal sampling probability. For HNS strategies, a negative item with a larger predicted score will have a higher sampling probability. For ease of understanding, we refer to [12] and define the “hardness” of a negative sample as its predicted score, i.e., a negative sample is “harder” than another when its score is larger. In what follows, $r(c, i|\theta)$ is abbreviated as r_{ci} for short.

²In collaborative filtering setting, a context c denotes a user $u \in \mathcal{U}$; in sequential recommendation setting, c can be a historical interaction sequence.

2.2 Hard Negative Sampling Strategies

Different from static sampling like uniform and popularity-aware strategy [10], HNS strategies are adaptive both to context and recommender models during the training. Here we review two widely-used HNS strategies.

DNS [40] ranks the negative items and oversamples the high-ranked items³. The sampling probability of DNS is defined as:

$$P_{ns}^{DNS}(j|c) = \begin{cases} \frac{1}{M}, & j \in S_{\mathcal{I}_c^-}^\downarrow[1, M] \\ 0, & j \in \text{others} \end{cases}, \quad (2)$$

where $S_{\mathcal{I}_c^-}^\downarrow[1, M] \subset \mathcal{I}_c^-$ denotes the subset of the top-ranked M negative items, i.e., the negative samples with top- M largest predicted scores. **Remark that the smaller the M is, the harder the negative samples will be drawn.**

Softmax-based sampling is widely used in adversarial learning [26, 33] and importance sampling [9, 21], where they refer to softmax distribution to assign higher sampling probability to higher scored items. The negative sampling probability can be defined as:

$$\begin{aligned} P_{ns}^{Softmax}(j|c) &= \frac{\exp(r_{cj}/\tau)}{\sum_{k \in \mathcal{I}_c^-} \exp(r_{ck}/\tau)} \\ &= \frac{\exp((r_{cj} - r_{ci})/\tau)}{\sum_{k \in \mathcal{I}_c^-} \exp((r_{ck} - r_{ci})/\tau)}, \end{aligned} \quad (3)$$

where τ is a temperature parameter. **It is noteworthy that the smaller the τ is, the harder the samples will be drawn.**

2.3 One-way Partial AUC

For each context c , we can define true positive rates (TPR) and false positive rates (FPR) as

$$TPR_{c,\theta}(t) = \Pr(r_{ci} > t | i \in \mathcal{I}_c^+), \quad (4)$$

$$FPR_{c,\theta}(t) = \Pr(r_{cj} > t | j \in \mathcal{I}_c^-). \quad (5)$$

Then, for a given $s \in [0, 1]$, let $TPR_{c,\theta}^{-1}(s) = \inf\{t \in \mathbb{R}, TPR_{c,\theta}(t) < s\}$ and $FPR_{c,\theta}^{-1}(s) = \inf\{t \in \mathbb{R}, FPR_{c,\theta}(t) < s\}$. Based on these, the AUC can be formulated as

$$\text{AUC}(\theta) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \int_0^1 TPR_{c,\theta} \left[FPR_{c,\theta}^{-1}(s) \right] ds. \quad (6)$$

As shown in Figure 2, One-way Partial AUC (OPAUC) only cares about the performance within a given false positive rate (FPR) range $[\alpha, \beta]$. Non-normalized OPAUC [13] is equal to

$$\text{OPAUC}(\theta, \alpha, \beta) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \int_\alpha^\beta TPR_{c,\theta} \left[FPR_{c,\theta}^{-1}(s) \right] ds. \quad (7)$$

In this paper, we consider the special case of OPAUC with $\alpha = 0$, which is denoted as $\text{OPAUC}(\beta)$ for short. Based on the definition in Eq. (7), we can have the following non-parametric estimator of $\text{OPAUC}(\beta)$:

$$\widehat{\text{OPAUC}}(\beta) = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{n_+} \frac{1}{n_-} \sum_{i \in \mathcal{I}_c^+} \sum_{j \in S_{\mathcal{I}_c^-}^\downarrow[1, n_- \cdot \beta]} \mathbb{I}(r_{ci} > r_{cj}), \quad (8)$$

³Without loss of generality, we consider a special case of DNS (Algorithm 2 in [40]) that set n to $|\mathcal{I}_c^-|$, set $\beta_1, \dots, \beta_{M-1}$ to 1 and set β_M, \dots, β_N to 0. Our analysis can generalize to the arbitrary multi-nomial distribution of β_k .

where $n_+ = |\mathcal{I}_c^+|$ and $n_- = |\mathcal{I}_c^-|$, and $\mathbb{I}(\cdot)$ is an indicator function. For simplicity, we assume $n_- \cdot \beta$ is a positive integer.

Since the OPAUC estimator in Eq. (8) is non-continuous and non-differentiable, we usually replace the indicator function with a continuous surrogate loss $L(c, i, j) = \ell(r_{ci} - r_{cj})$. With suitable surrogate loss $\ell(\cdot)$, maximizing $\widehat{\text{OPAUC}}(\beta)$ in Eq. (8) is equivalent to the following problem:

$$\min_{\theta} \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{n_+} \sum_{i \in \mathcal{I}_c^+} \frac{1}{n_- \cdot \beta} \sum_{j \in S_{\mathcal{I}_c^-}^\downarrow[1, n_- \cdot \beta]} L(c, i, j). \quad (9)$$

Remark that the objective is divided by a fixed constant β for proof, which does not affect the properties of the objective function. For surrogate loss $\ell(\cdot)$, [16] proposes a sufficient condition to ensure it consistent for OPAUC maximization, where the widely used logistic loss $\ell(t) = \log(1 + \exp(-t))$ satisfies the properties.

Additionally, for comparison among different β , we define normalized OPAUC(β) following [24],

$$\text{OPAUC}_{\text{norm}}(\beta) = \text{Trans}(\text{OPAUC}(\beta)), \quad (10)$$

where the normalized transformation is defined as:

$$\text{Trans}(A) = \frac{1}{2} \left[1 + \frac{A - \min_{\theta} A}{\max_{\theta} A - \min_{\theta} A} \right]. \quad (11)$$

2.4 Distributionally Robust Optimization

Given a divergence D_{ϕ} between two distributions P and Q , Distributionally Robust Optimization (DRO) aims to minimize the expected risk over the worst-case distribution Q [19, 22, 27], where Q is in a divergence ball around training distribution P . Formally, it can be defined as:

$$\begin{aligned} \min_{\theta} \sup_Q E_Q [\mathcal{L}(f_{\theta}(\mathbf{x}), y)] \\ \text{s.t. } D_{\phi}(Q||P) \leq \rho, \end{aligned} \quad (12)$$

where the hyperparameter ρ modulates the distributional shift, \mathcal{L} is the loss function. In this paper, we will focus on two special divergence metrics, i.e. the KL divergence $D_{KL}(Q||P) = \int \log(\frac{dQ}{dP}) dQ$ [18] and the CVaR divergence $D_{CVaR}(Q||P) = \sup \log(\frac{dQ}{dP})$ [14].

3 HARD NEGATIVE SAMPLING MEETS OPAUC

In this section, we prove that the BPR loss equipped with HNS optimizes $\text{OPAUC}(\beta)$, which is the first step to understanding the effectiveness of HNS.

We achieve the proof based on the DRO objective and present the proof outline in Figure 5. Following the theorems proposed in [41], we first show the connection between the OPAUC objective and the DRO-based objective. Then we prove that the personalized ranking problem (Eq. (1)) equipped with HNS is equivalent to the DRO-based objective in our theorems.

Following [41], we define the DRO-based objective as:

$$\begin{aligned} \min_{\theta} \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{n_+} \sum_{i \in \mathcal{I}_c^+} \max_Q E_Q [L(c, i, j)] \\ \text{s.t. } D_{\phi}(Q||P_0) \leq \rho, \end{aligned} \quad (13)$$

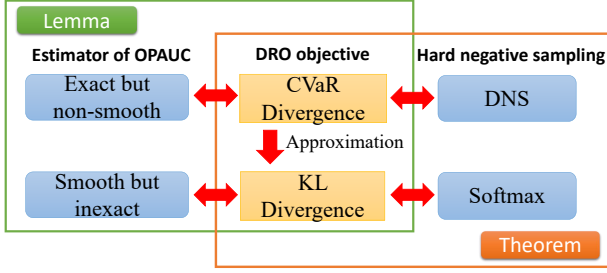


Figure 5: The Lemma 1 shows the equivalence between the OPAUC estimator and DRO objective. Based on DRO objective, we prove the equivalence between HNS and OPAUC in Theorem 1 and Theorem 2.

where P_0 denotes uniform distribution over \mathcal{I}_c^- , the hyperparameter ρ modulates the degree of distributional shift, D_ϕ is the divergence measure between distributions.

Then we show the connection between the OPAUC objective and the DRO-based objective through the following lemma:

LEMMA 1 (THEOREM 1 OF [41]). *By choosing CVaR divergence $D_\phi = D_{CVaR}(Q||P_0) = \sup \log(\frac{dQ}{dP_0})$ and setting $\beta = e^{-\rho}$, the DRO-based objective (Eq. (13)) is equivalent to the OPAUC(β) objective (Eq. (9)).*

Based on the above lemma, we prove the equivalence between the OPAUC objective and the HNS based objective.

THEOREM 1. *By choosing $P_{ns} = P_{ns}^{DNS}$,*

$$M = n_- \cdot \beta, \quad (14)$$

the DNS based problem (Eq. (1)) is equivalent to the OPAUC(β) objective (Eq. (9)).

PROOF. Given Lemma 1, we just need to show that DNS sampling based problem (Eq. (1)) is equivalent to the DRO-based objective (Eq. (13)).

By choosing CVaR divergence, then the DRO-based objective (Eq. (13)) reduces to [41] (using strong duality and Theorem 4 in [30])

$$\min_{\theta} \min_{\eta \geq 0} \frac{1}{|C|} \sum_{c \in C} \frac{1}{n_+} \sum_{i \in \mathcal{I}_c^+} \left\{ \frac{1}{e^{-\rho}} \cdot E_{j \sim P_0} [(L(c, i, j) - \eta_i)_+] + \eta_i \right\}, \quad (15)$$

where P_0 denotes uniform distribution over \mathcal{I}_c^- . Following [39], it's easy to see that the optimal η_i is the $e^{-\rho}$ -quantile of $L(c, i, j)$, which is defined as:

$$\eta_i^* = \inf_{\eta_i} \{P_{j \sim P_0} [L(c, i, j) > \eta_i] < e^{-\rho}\}. \quad (16)$$

Substitute η_i with η_i^* in Eq. (15) and replace $e^{-\rho}$ with $\frac{M}{n_-}$, then we obtain the equivalence between DNS sampling based problem (Eq. (1)) and DRO-based objective (Eq. (13)). Recall the conclusion in Lemma 1, then we complete the proof by setting $M = n_- \cdot \beta$. \square

Remark: The DNS based problem is an exact but non-smooth estimator of OPAUC(β), which is consistent for OPAUC(β) maximization. **The hyperparameter M in DNS strategy directly determines β in the OPAUC objective.**

THEOREM 2. *By choosing $P_{ns} = P_{ns}^{Softmax}$,*

$$\tau = \sqrt{\frac{\text{Var}_j(L(c, i, j))}{-2 \log \beta}}, \quad (17)$$

$$\text{Var}_j(L(c, i, j)) = E_{j \sim P_0} [(L(c, i, j) - E_{j \sim P_0} [L(c, i, j)])^2], \quad (18)$$

then problem (Eq. (1)) equipped with softmax-based sampling strategy is a surrogate version of the OPAUC(β) objective (Eq. (9)).

The proof process is similar to Theorem 1. Substitute CVaR divergence with KL divergence but remain the same ρ , then we get a soft estimator of OPAUC(β). We prove that the soft estimator is equivalent to softmax-based sampling problem (Eq. (1)). The precise relationship between τ and β is complex and hard to compute. Hence we get an approximate version via the Taylor expansion. The detailed proof can be found in Appendix A.

Remark: The BPR loss equipped with softmax-based sampling is a smooth but inexact estimator of OPAUC(β). **The hyperparameter τ in softmax-based sampling directly determines β in OPAUC objective.**

4 OPAUC MEETS TOP-K METRICS

In this section, we investigate the connection between OPAUC(β) and Top-K evaluation metrics, which is the second step to understanding the effectiveness of HNS. We propose two arguments to declare their relationship:

- (1) **Compared to AUC, OPAUC(β) has a stronger correlation with Top-K evaluation metrics.**
- (2) **A smaller K in Top-K evaluation metrics has a stronger correlation with a smaller β in OPAUC(β).**

We conduct theoretical analysis and simulation experiments to verify our proposals as follows.

4.1 Theoretical Analysis

In this subsection, we analyze the connection between OPAUC(β) and Top-K metrics from a theoretical perspective. To be concrete, we prove that given K , Precision@ K and Recall@ K are higher bounded and lower bounded by the functions of specific OPAUC(β).

THEOREM 3. *Suppose there are N_+ positive items and N_- negative items, where $N_+ > K$ and $N_- > K$. For any permutation of all items in descending order, we have*

$$\frac{1}{N_+} \left[\frac{N_+ + K - \sqrt{(N_+ + K)^2 - 4N_+N_- \times \text{OPAUC}(\beta)}}{2} \right] \leq \text{Recall}@K \leq \frac{1}{N_+} \left[\sqrt{N_+N_- \times \text{OPAUC}(\beta)} \right], \quad (19)$$

$$\frac{1}{K} \left[\frac{N_+ + K - \sqrt{(N_+ + K)^2 - 4N_+N_- \times \text{OPAUC}(\beta)}}{2} \right] \leq \text{Precision}@K \leq \frac{1}{K} \left[\sqrt{N_+N_- \times \text{OPAUC}(\beta)} \right], \quad (20)$$

where $\beta = \frac{K}{N_-}$.

Remark: From above, we get the following inspirations:

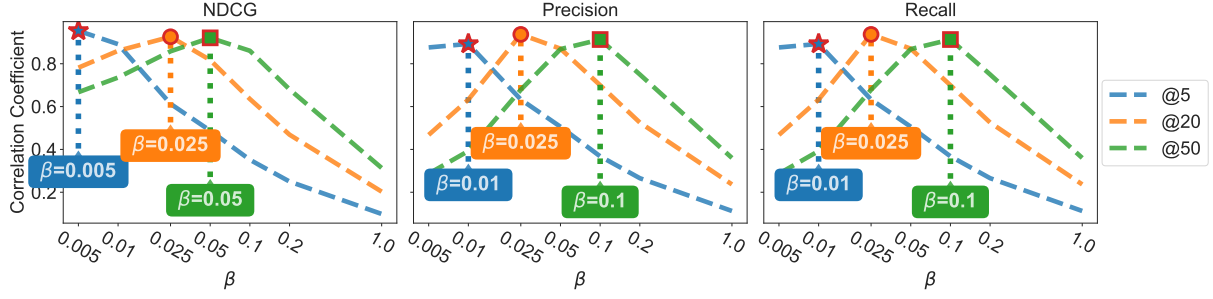


Figure 6: The estimated correlation coefficient between Top- K evaluation metrics and $\text{OPAUC}_{norm}(\beta)$ under Monte Carlo sampling experiments, where $N_+ = 200$ and $N_- = 800$. We highlight the value of β when each curve reaches its maximum correlation coefficient. Remark that AUC is also a special case of $\text{OPAUC}_{norm}(\beta)$ with $\beta = 1$.

- (1) The Top- K metrics like Precision@ K and Recall@ K have a strong connection with specific $\text{OPAUC}(\beta)$, where $\beta = \frac{K}{N_-}$. However, such a connection does not exist for AUC, which confirms our first argument. Hence, maximizing specific $\text{OPAUC}(\beta)$ approximately optimizes specific Precision@ K and Recall@ K .
- (2) The smaller the K is, the smaller the β ($= \frac{K}{N_-}$) should be considered. A smaller K has a stronger connection with a smaller β , which effectively verifies our second argument.

4.2 Simulation Experiments

In this subsection, we conduct Monte Carlo sampling experiments to analyze the connection between $\text{OPAUC}(\beta)$ and Top- K evaluation metrics. For comparison among different β , we use normalized OPAUC defined in Eq. (10) here. Suppose there are N_+ positive items and N_- negative items in the item set \mathcal{I} . Due to the vast scale of the entire permutation space of items, it is impossible to enumerate all cases for analyses directly. Hence, we make a Monte-Carlo approximation and uniformly sample permutations from the space as simulated ranking lists 100000 times. Then we calculate the evaluation metrics (Top- K metrics and $\text{OPAUC}_{norm}(\beta)$) for these simulated ranking lists. Afterward, we estimate the correlation coefficient between Top- K metrics and $\text{OPAUC}_{norm}(\beta)$ and report them in Figure 6. We report β of $\text{OPAUC}_{norm}(\beta)$ in logarithmic scale. Furthermore, we highlight the value of β when each curve reaches its maximum correlation coefficient. Remark that $\text{OPAUC}_{norm}(1)$ is equal to AUC.

As shown in Figure 6, we have the following observations:

- (1) The correlation coefficient of the highest point of the curve is much larger than the correlation coefficient when β is equal to 1. That means most Top- K evaluation metrics have higher correlation coefficients with specific $\text{OPAUC}_{norm}(\beta)$ (above 0.8) than AUC (under 0.4), which clearly verifies our first argument.
- (2) Given a specific K in Top- K metrics, the correlation coefficient with $\text{OPAUC}_{norm}(\beta)$ gets the maximum value at a specific β . Both too large and too small β will degrade the correlation with specific Top- K metrics.
- (3) For different K , the peak of the curve varies according to β . The smaller the K in the Top- K evaluation metrics, the smaller the β that takes the maximum value of the correlation coefficient. This effectively confirms our second argument.

- (4) On the left side of the peak of the curve, we find that the correlation coefficient of NDCG@ K descends more slowly than the other two metrics. This is because NDCG@ K pays more attention to top-ranked items in Top- K items.

5 DEEP UNDERSTANDING OF HNS

Based on the arguments discussed above, we gain a deeper theoretical understanding of HNS. The BPR loss equipped with HNS optimizes $\text{OPAUC}(\beta)$, which has a stronger connection with Top- K metrics. In this sense, we derive the following corollary:

COROLLARY 1. *The BPR loss equipped with HNS approximately optimizes Top- K evaluation metrics, where the level of sampling hardness controls the value of K .*

Moreover, we take a step further and propose two instructive guidelines for effective usage of HNS.

- (1) **The sampling hardness should be controllable, e.g., via pre-defined hyper-parameters, to adapt to different Top- K metrics and datasets.**
- (2) **The smaller the K we emphasize in Top- K evaluation metrics, the harder the negative samples we should draw.**

Motivated by these, we generalize the DNS and softmax-based sampling to two controllable algorithms DNS(M, N) and Softmax- $v(\rho, N)$, as shown in Algorithm 1 and Algorithm 2 respectively.

- In DNS(M, N), we utilize hyperparameter M to control sampling hardness, where the original DNS is a special case with $M = 1$.
- In Softmax- $v(\rho, N)$, we propose to use an adaptive τ in Eq. (17), instead of a fixed τ in Eq. (3). Hyperparameter ρ controls the sampling hardness. This ensures that β of the optimization objective $\text{OPAUC}(\beta)$ remains the same during training.

As discussed, the hyperparameters M and ρ affect how hard the negative samples we will draw. Besides, the size of the sampling pool N also affects the actual sampling probability of negative items. We conduct simulation experiments to investigate the difference of the sampling distribution under different parameter settings. We choose the user embeddings and items embeddings from the well-trained model on the Gowalla dataset and keep them fixed. Then, we randomly pick a (user, positive item) pair (c, i) and then simulate the sampling process 10000 times to estimate the actual sampling probability. The average value of p_{cij} over the sampling process

Algorithm 1 DNS (M, N)

```

1: Initialize  $\theta$ 
2: for  $t = 1, \dots, T$  do
3:   Sample a mini-batch  $\mathcal{B} \in \mathcal{D}$ 
4:   for  $(c, i) \in \mathcal{B}$  do
5:     Uniformly sample a mini-batch  $\mathcal{B}'_c \in \mathcal{I}_c^-, |\mathcal{B}'_c| = N$ .
6:     Let  $p_{cij} = \begin{cases} \frac{1}{M}, & j \in \mathcal{S}_{\mathcal{B}'_c}^\downarrow[1, M] \\ 0, & j \in \text{others.} \end{cases}$ 
7:   end for
8:   Compute a gradient estimator  $\nabla_t$  by

```

$$\nabla_t = \frac{1}{|\mathcal{B}|} \sum_{(c,i) \in \mathcal{B}} \sum_{j \in \mathcal{I}_c^-} p_{cij} \nabla_{\theta} L(c, i, j).$$

```

9:   Update  $\theta_{t+1} = \theta_t - \eta \nabla_t$ .
10: end for

```

Algorithm 2 Softmax-v (ρ, N)

```

1: Initialize  $\theta$ 
2: for  $t = 1, \dots, T$  do
3:   Sample a mini-batch  $\mathcal{B} \in \mathcal{D}$ 
4:   for  $(c, i) \in \mathcal{B}$  do
5:     Uniformly sample a mini-batch  $\mathcal{B}'_c \in \mathcal{I}_c^-, |\mathcal{B}'_c| = N$ .
6:     Let  $p_{cij} = \begin{cases} \frac{e^{\tau(r_{ci} - r_{cj})}}{\sum_{k \in \mathcal{B}'_c} e^{\tau(r_{ci} - r_{ck})}}, & j \in \mathcal{B}'_c \\ 0, & j \in \text{others,} \end{cases}$ 

```

where $\tau = \sqrt{\frac{\text{Var}_j(L(c, i, j))}{2\rho}}$.

```

7:   end for
8:   Compute a gradient estimator  $\nabla_t$  by

```

$$\nabla_t = \frac{1}{|\mathcal{B}|} \sum_{(c,i) \in \mathcal{B}} \sum_{j \in \mathcal{I}_c^-} p_{cij} \nabla_{\theta} L(c, i, j).$$

```

9:   Update  $\theta_{t+1} = \theta_t - \eta \nabla_t$ .
10: end for

```

is approximated as the actual sampling probability that negative item j is chosen by pair (c, i) for training. We report the cumulative probability distribution under different parameter settings in Figure 7. The negative items are in descending order w.r.t. their scores.

Since items are in descending order, we conclude that the faster the curve rises, the higher the sampling probability the top-ranked items are drawn with. Easily, we have the following observations:

- Smaller M in DNS(M, N) means higher sampling hardness.
- Larger N in DNS(M, N) means higher sampling hardness.
- Larger ρ in Softmax-v(ρ, N) means higher sampling hardness.

6 EXPERIMENTS

In this section, we evaluate the models on three public datasets to figure out the following questions:

- (Q1) How do DNS(M, N) and Softmax-v(ρ, N) perform compared to state-of-the-art HNS methods? Is it beneficial to control sampling hardness with pre-defined hyperparameters?
- (Q2) Can experiment results validate our second guideline on adjusting sampling hardness according to K in Top- K metrics?

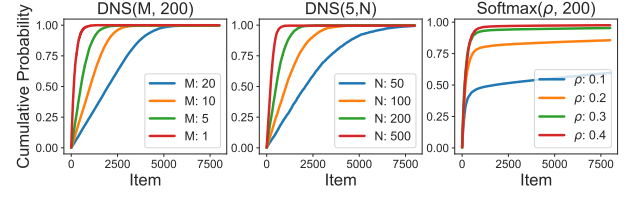


Figure 7: Approximated distributions under different parameter settings. The faster the curve rises, the higher the sampling probability the top-ranked items are drawn with.

Table 1: The Statistics of Datasets

Dataset	User	Item	Train	Test	Sparsity
Gowalla	29,858	40,988	822,358	205,106	99.9160%
Yelp	77,277	45,638	1,684,846	419,049	99.9403%
Amazon	130,380	128,939	1,934,404	481,246	99.9856%

Dataset. The Statistics of three public datasets are shown in Table 1, which vary in scale and sparsity. The Gowalla dataset is the collection of user check-in histories. The Yelp dataset is a subset of Yelp’s businesses, reviews, and user data. The Amazon dataset is a subset of customers’ ratings for Amazon books. Considering the ratings are integers ranging from 1 to 5, the ratings above 4 are regarded as positive. Following [9, 21], we leverage the routine strategy – 5-core setting to preprocess the dataset.

For each user, we randomly select 80% of items to form the training set and 20% of items to form the test set. 10% of the training set is used for validation. The models are built on the training set and evaluated on the test set.

Metrics. When evaluating the models, we filter out positive items in the training set and utilize widely-used metrics Recall@ K and NDCG@ K to evaluate the recommendation performance. The detailed definitions are shown in Appendix C.

6.1 Baselines

To verify the effectiveness of DNS(M, N) and Softmax-v(ρ, N) methods, we compare our algorithms with the following baselines.

- **BPR** [29] is a classical method for implicit feedback. It utilizes pairwise logit loss and randomly samples negative items.
- **AOBPR** [28] improves BPR through adaptively oversampling top-ranked negative items.
- **WARP** [35] uses the Weighted Approximate-Rank Pairwise loss function for implicit feedback.
- **IRGAN** [33] utilizes a minimax game to optimize the generative and discriminative network simultaneously. The negative items are drawn based on softmax distribution.
- **DNS** [40] is a dynamic negative sampler, which is a special case of DNS(M, N) with $M = 1$.
- **Kernel** [2] is an efficient sampling method that approximates the softmax distribution with non-negative quadratic kernel.
- **PRIS** [21] utilizes importance sampling for training, where importance weights are based on softmax distribution. They adopt

Table 2: Performance comparison on three datasets. The best results are in bold and the second best are underlined. The baselines are taken from [9], as we completely follow their experiment settings. “*” denote the improvement is significant with t-test with $p < 0.05$.**

Method	Gowalla		Yelp		Amazon	
	NDCG@50	Recall@50	NDCG@50	Recall@50	NDCG@50	Recall@50
BPR	0.1216	0.2048	0.0524	0.1083	0.0499	0.1171
AOBPR	0.1385	0.2417	0.0677	0.1346	0.0563	0.1303
WARP	0.1248	0.2240	0.0636	0.1332	0.0542	0.1267
IRGAN	0.1443	0.2242	0.0695	0.1367	0.0627	0.1395
Kernel	0.1399	0.2264	0.0658	0.1315	0.0700	0.1495
DNS	0.1412	0.1839	0.0693	0.1425	0.0615	0.1378
PRIS(U)	0.1334	0.2217	0.0639	0.1273	0.0607	0.1377
PRIS(P)	0.1385	0.2282	0.0673	0.1342	0.0697	0.1463
AdaSIR(U)	0.1489	0.2500	0.0732	0.1523	0.0731	0.1505
AdaSIR(P)	0.1519	0.2516	0.0731	0.1525	0.0740	0.1534
DNS(M, N)	<u>0.1811</u> **	<u>0.2989</u> **	0.0899 **	0.1774 **	<u>0.1014</u> **	<u>0.1833</u> **
Softmax- $v(\rho, N)$	0.1837 **	0.2993 **	<u>0.0840</u> **	<u>0.1690</u> **	0.1046 **	0.1937 **

the uniform and popularity-based distribution to construct the sampling pool, denoted as PRIS(U) and PRIS(P), respectively.

- **AdaSIR** [9] is a two-stage method that maintains a fixed size contextualized sample pool with importance resampling. The importance weights are based on softmax distribution. They adopt the uniform and popularity-based distribution to construct the sampling pool, denoted as AdaSIR(U) and AdaSIR(P), respectively.

6.2 Implementation Details

The algorithms are implemented based on PyTorch. We completely follow the experiments setting in [9, 21]. We utilize Matrix Factorization (MF) as the recommender model for our model. We utilize Adam optimizer to optimize all parameters. The dimension of user and item embedding is set to 32. The batch size is fixed to 4096, and the learning rate is set to 0.001 by default. The number of training epochs is set to 200 for all methods. We utilize grid search to find the best with $\text{weight_decay} \in \{0.1, 0.01, 0.001, 0.0001\}$. The hyperparameter M in DNS(M, N) is tuned over $\{1, 2, 3, 4, 5, 10, 20\}$ and the hyperparameter ρ of Softmax- $v(\rho, N)$ is tuned over $\{0.01, 0.1, 1, 10, 100\}$ for all datasets. Due to the efficiency limit, the sample pool size N for each user is set to 200, 200, and 500 for Gowalla, Yelp, and Amazon. The maximum number of negative samples per positive pair (c, i) is the sample pool size. The baseline results are directly taken from [9], as we completely follow their experiment setting. Code is available at https://github.com/swt-user/WWW_2023_code.

6.3 (RQ1) Performance Comparison

Table 2 shows the performance of DNS(M, N), Softmax- $v(\rho, N)$, and baselines. From them, we have the following key findings:

- Compared to the uniform negative sampling method BPR, most HNS methods perform much better, especially DNS(M, N) and Softmax- $v(\rho, N)$. This clearly verifies the effectiveness of HNS.
- Benefiting from the adjustable sampling hardness, DNS(M, N) significantly outperform its original version on average 40%. Meanwhile, the two methods also present a huge performance boost

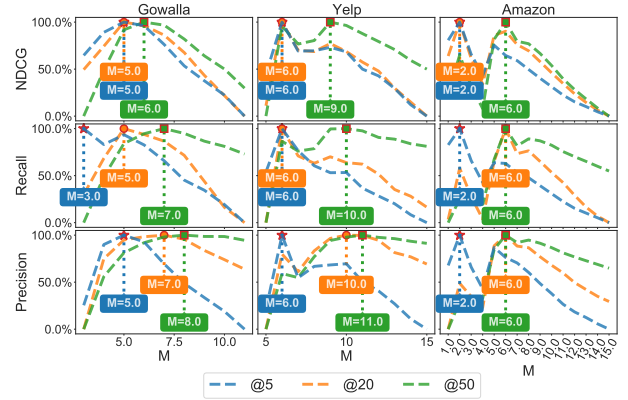


Figure 8: The effect of M in DNS(M, N), where N is set to 200, 200, 500 for Gowalla, Yelp and Amazon respectively.

over other HNS methods. These findings demonstrate the extreme importance of our first guideline in Section 5.

6.4 (RQ2) Performance with Different Sampling Distributions

This subsection investigates how Top-K metrics will change under different sampling distributions on real-world datasets. As the sampling distribution is affected by hyperparameters, see Section 5, we investigate the performance under different hyperparameter settings.

We report the performance results on three public dataset under different M in DNS(M, N), different N in DNS(M, N) and different ρ in Softmax- $v(\rho, N)$ in Figure 8, Figure 9 and Figure 10 respectively. We only care about the relative magnitude of Top-K metrics, so we report the relative value of Top-K evaluation metrics for better visualization. We highlight the value of hyperparameters when each curve reaches its maximum result. For each result, we tune

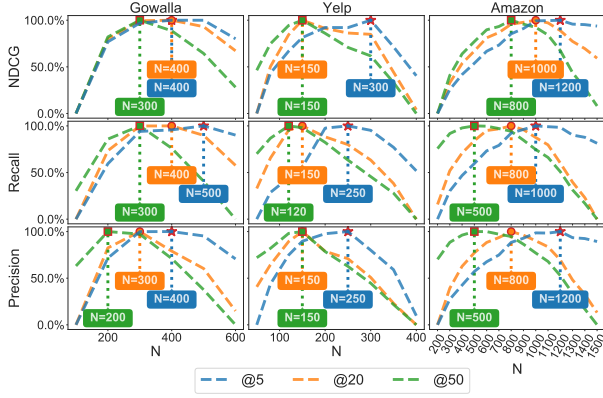


Figure 9: The effect of N in $\text{DNS}(M, N)$, where M is set to 5 for all three datasets.

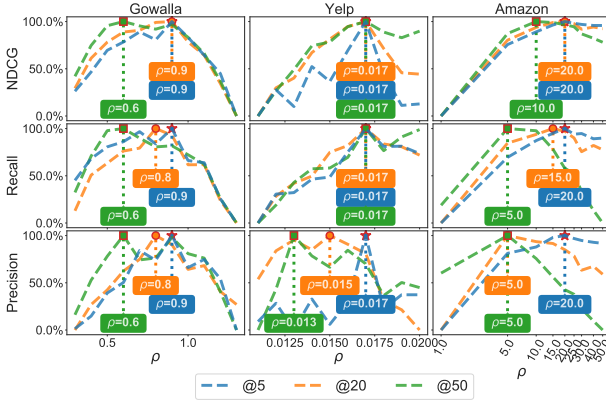


Figure 10: The effect of ρ in $\text{Softmax-v}(\rho, N)$, where N is set to 200, 200, 500 for Gowalla, Yelp and Amazon respectively.

the learning rate $\in \{0.01, 0.001\}$ and weight_decay $\in \{0.01, 0.001, 0.0001\}$ to find the best.

- From Figure 8, we observe that for all datasets and all metrics, the lower the K in Top- K metrics is, the smaller the M in $\text{DNS}(M, N)$ when the curve achieves its maximum performance.
- From Figure 9, we observe that for all datasets and all metrics, the lower the K in Top- K metrics is, the larger the N in $\text{DNS}(M, N)$ when the curve achieves its maximum performance.
- From Figure 10, we observe that for all datasets and all metrics, the lower the K in Top- K metrics is, the larger the ρ in $\text{Softmax-v}(\rho, N)$ when the curve achieves its maximum performance.

In some cases, the peak of the Top-20 curve coincides with the peak of the Top-50 curve or Top-5 curve. This can be attributed to the relatively small difference of K . With a larger difference of K , for example, Top-50 and Top-5, their curve always matches our observation. We conduct further experiments to investigate the performance across a wide range of K in Appendix D.

Recall that we have observed how hyperparameters (M, N, ρ) affect sampling hardness in Figure 7. Combining these two observations, we can easily conclude that **the smaller the K in Top- K metrics, the harder the negative samples we should draw**. These clearly verify our second guideline.

7 RELATED WORK

7.1 Negative Sampling for Recommendation

Early work sample items based on predefined distributions, e.g., uniform distribution [11, 29] and popularity-based distribution [3, 10]. These static samplers are independent of model status and unchanged for different users. Thus, the performance is limited. Later on, adaptive samplers are proposed, such as DNS [40] and softmax-based sampling methods. Softmax-based sampling is widely used in adversarial learning (e.g. IRGAN [33] and ADVIR [26]) and importance sampling (e.g. PRIS [21] and AdaSIR [9]). They assign high sampling probability to top-ranked negative items, accounting for model status. There are also some fine-grained negative sampling methods [23, 32, 34, 42]. Empirical experiments verify the effectiveness and efficiency of HNS. The efficiency problem has been studied in AOBPR [28]. They argue that HNS samples more informative high-scored items, which can contribute more to the gradients and accelerate the convergence. Nevertheless, the reasons for the effectiveness of HNS are not revealed yet. To the best of our knowledge, only DNS [40] provides clues of the connection between HNS and Top- K metrics. But unfortunately, they fail to give a theoretical foundation and deep analyses.

7.2 Partial AUC Maximization

Early work does not directly optimize the surrogate objective of Partial AUC, but instead, some other related objectives, e.g., p-norm push [31], infinite-push [20], and asymmetric SVM objective [36]. Nevertheless, these algorithms are not scalable and applicable to deep learning. More recently, [38] considers two-way partial AUC maximization and simplifies the optimizing problem for large scale optimization. [41] proposes new formulations of Partial AUC surrogate objectives using distributionally robust optimization (DRO). This work motivates our proof of the connection between OPAUC and HNS. A more comprehensive study of AUC can refer to [37].

8 CONCLUSION

In this paper, we reveal the theories behind HNS for recommendation. We prove that the BPR loss equipped with HNS strategies optimizes OPAUC. Meanwhile, we conduct theoretical analysis and simulation experiments to show the strong connection between OPAUC and Top- K evaluation metrics. On these bases, the effectiveness of HNS can be clearly explained. To take a step further, we propose two insightful guidelines for effective usage of HNS. In conclusion, the proposed theoretical understanding of HNS can not only explain effectiveness but also provide insightful guidelines for future study.

In the future, we will investigate the connection between Two-way Partial AUC and recommendation algorithms, which may bring more insights into recommendation systems. The effect of false negative items will also be an exciting research direction.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2022YFB3104701), the Starry Night Science

Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-001), the National Natural Science Foundation of China (61972372, 62121002, 62102382), and the CCCD Key Lab of Ministry of Culture and Tourism.

REFERENCES

- [1] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A Generic Coordinate Descent Framework for Learning from Implicit Feedback. In *WWW*. ACM, 1341–1350.
- [2] Guy Blanc and Steffen Rendle. 2018. Adaptive Sampled Softmax with Kernel Based Sampling. In *ICML*. 590–599.
- [3] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. 2018. Word2vec applied to recommendation: hyperparameters matter. In *RecSys*. 352–356.
- [4] Chong Chen, Weizhi Ma, Min Zhang, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2022. Revisiting Negative Sampling VS. Non-Sampling in Implicit Recommendation. *ACM Trans. Inf. Syst.* (2022).
- [5] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation. In *AAAI*.
- [6] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Trans. Inf. Syst.* 38 (2020).
- [7] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *CoRR* abs/2010.03240 (2020).
- [8] Jiawei Chen, Chengquan Jiang, Can Wang, Sheng Zhou, Yan Feng, Chun Chen, Martin Ester, and Xiangnan He. 2021. CoSam: An Efficient Collaborative Adaptive Sampler for Recommendation. *ACM Trans. Inf. Syst.* 39 (2021).
- [9] Jin Chen, Defu Lian, Binbin Jin, Kai Zheng, and Enhong Chen. 2022. Learning Recommenders for Implicit Feedback with Importance Resampling. In *WWW*. 1997–2005.
- [10] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On Sampling Strategies for Neural Network-based Collaborative Filtering. In *SIGKDD*.
- [11] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data. In *IJCAI*. 2230–2236.
- [12] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and Robustify Negative Sampling for Implicit Collaborative Filtering. In *NIPS*. 1094–1105.
- [13] Lori E Dodd and Margaret S Pepe. 2003. Partial AUC estimation and regression. *Biometrics* 59, 3 (2003), 614–623.
- [14] John C. Duchi and Hongseok Namkoong. 2018. Learning Models with Uniform Performance via Distributionally Robust Optimization. *CoRR* abs/1810.08750 (2018).
- [15] Louis Faury, Ugo Tanielian, Elvis Dohmatob, Elena Smirnova, and Flavian Vasile. 2020. Distributionally Robust Counterfactual Risk Minimization. In *AAAI*. 3850–3857.
- [16] Wei Gao and Zhi-Hua Zhou. 2015. On the Consistency of AUC Pairwise Optimization. In *IJCAI*. 939–945.
- [17] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *SIGIR*. ACM, 549–558.
- [18] Zhaolin Hu and L Jeff Hong. 2013. Kullback-Leibler divergence constrained distributionally robust optimization. *Available at Optimization Online* (2013), 1695–1724.
- [19] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. 2020. Large-Scale Methods for Distributionally Robust Optimization. In *NIPS*, Vol. 33. 8847–8860.
- [20] Nan Li, Rong Jin, and Zhi-Hua Zhou. 2014. Top Rank Optimization in Linear Time. In *NIPS*. 1502–1510.
- [21] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized Ranking with Importance Sampling. In *WWW*. 1093–1103.
- [22] Fengming Lin, Xiaolei Fang, and Zheming Gao. 2022. Distributionally Robust Optimization: A review on theory and applications. *Numerical Algebra, Control and Optimization* 12 (2022), 159–212.
- [23] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Boosting the Speed of Entity Alignment 10 ×: Dual Attention Matching Network with Normalized Hard Sample Mining. In *WWW*. 821–832.
- [24] Donna McClish. 1989. Analyzing a portion of the ROC Curve. *Medical decision making : an international journal of the Society for Medical Decision Making* 9 (1989), 190–5.
- [25] Khashayar Namdar, Masoom A. Haider, and Farzad Khalvati. 2021. A Modified AUC for Training Convolutional Neural Networks: Taking Confidence Into Account. *Frontiers Artif. Intell.* 4 (2021), 582928.
- [26] Dae Hoon Park and Yi Chang. 2019. Adversarial Sampling and Training for Semi-Supervised Information Retrieval. In *WWW*. 1443–1453.
- [27] Hamed Rahimian and Sanjay Mehrotra. 2019. Distributionally Robust Optimization: A Review. *CoRR* abs/1908.05659 (2019).
- [28] Steffen Rendle and Christoph Freudenthaler. 2014. Improving Pairwise Learning for Item Recommendation from Implicit Feedback. In *WSDM*. 273–282.
- [29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. 452–461.
- [30] R Tyrrell Rockafellar. 2017. Risk and utility in the duality framework of convex analysis. In *Jonathan M. Borwein Commemorative Conference*. 21–42.
- [31] Cynthia Rudin. 2009. The P-Norm Push: A Simple Convex Ranking Algorithm That Concentrates at the Top of the List. *J. Mach. Learn. Res.* 10 (2009), 2233–2271.
- [32] Qi Wan, Xiangnan He, Xiang Wang, Jiancan Wu, Wei Guo, and Ruiming Tang. 2022. Cross Pairwise Ranking for Unbiased Item Recommendation. In *WWW*. 2370–2378.
- [33] 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*. 515–524.
- [34] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *WWW*. 99–109.
- [35] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling up to Large Vocabulary Image Annotation. In *IJCAI*. 2764–2770.
- [36] Shan-Hung Wu, Keng-Pei Lin, Chung-Min Chen, and Ming-Syan Chen. 2008. Asymmetric support vector machines: low false-positive learning under the user tolerance. In *KDD*. ACM, 749–757.
- [37] Tianbao Yang and Yiming Ying. 2022. AUC Maximization in the Era of Big Data and AI: A Survey. *ACM Comput. Surv.* (jul 2022).
- [38] Zhiyong Yang, Qianqian Xu, Shilong Bao, Yuan He, Xiaochun Cao, and Qingming Huang. 2021. When All We Need is a Piece of the Pie: A Generic Framework for Optimizing Two-way Partial AUC. In *ICML*. 11820–11829.
- [39] Runtian Zhai, Chen Dan, J. Zico Kolter, and Pradeep Ravikumar. 2021. DORO: Distributional and Outlier Robust Optimization. In *ICML*. 12345–12355.
- [40] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *SIGIR*. 785–788.
- [41] Dixian Zhu, Gang Li, Bokun Wang, Xiaodong Wu, and Tianbao Yang. 2022. When AUC meets DRO: Optimizing Partial AUC for Deep Learning with Non-Convex Convergence Guarantee. In *ICML*. 27548–27573.
- [42] Qiannan Zhu, Haobo Zhang, Qing He, and Zhicheng Dou. 2022. A Gain-Tuning Dynamic Negative Sampler for Recommendation. In *WWW*. 277–285.

A PROOF OF THEOREM 2

PROOF. As shown in Lemma 1, the DRO-based objective (Eq. (13)) is equivalent to $OPAUC(\beta)$ (Eq. (9)). By replacing CVaR divergence with KL divergence $D_\phi = D_{KL}(Q||P_0) = \int \log(\frac{dQ}{dP})dQ$, then the DRO-based objective (Eq. (13)) reduces to

$$\min_{\theta} \min_{\lambda \geq 0} \frac{1}{|C|} \sum_{c \in C} \frac{1}{n_+} \sum_{i \in I_c^+} \{\lambda_i \cdot \log E_{j \sim P_0} \left[\exp \left(\frac{L(c, i, j)}{\lambda_i} \right) \right] + \lambda_i \cdot \rho\}. \quad (21)$$

The detailed derivation can be found in [18]. By setting $\beta = \exp(-\rho)$, we get a surrogate objective of $OPAUC(\beta)$. Next, we will show that it is equivalent to the softmax-based sampling problem.

Differentiate the objective respect to λ_i and set to 0, and then we find that the optimal λ_i is the solution to the fixed-point equation:

$$\lambda_i = E_{j \sim P_0} \left[\frac{e^{\frac{L(c, i, j)}{\lambda_i}} \cdot L(c, i, j)}{E_{j \sim P_0} \left[e^{\frac{L(c, i, j)}{\lambda_i}} \right]} \right] \cdot \frac{1}{\rho + \log E_{j \sim P_0} \left[e^{\frac{L(c, i, j)}{\lambda_i}} \right]}. \quad (22)$$

Replace the above value for λ_i in Eq. (21), and then we derive the following result:

$$\min_{\theta} \frac{1}{|C|} \sum_{c \in C} \frac{1}{n_+} \sum_{i \in I_c^+} \left\{ E_{j \sim P_0} \left[\frac{e^{\frac{L(c, i, j)}{\lambda_i}}}{E_{j \sim P_0} \left[e^{\frac{L(c, i, j)}{\lambda_i}} \right]} L(c, i, j) \right] \right\}, \quad (23)$$

where P_0 denotes uniform distribution over I_c^- . By setting $\lambda_i = \tau$, the analogy of Eq. (23) and the softmax sampling based problem (Eq. (1)) is obvious. The only difference is that the index term in Eq. (23) is $\frac{\ell(r_{ci} - r_{cj})}{\tau}$ but $\frac{r_{cj} - r_{ci}}{\tau}$ in Eq. (1). When choosing $\ell(t) = \log(1 + \exp(-t))$, it is consistent for optimization.

By now, we have proven the equivalence between softmax sampling based problem (Eq. (1)) and $OPAUC(\beta)$ objective (Eq. (9)). However, it is impossible to directly compute λ_i with Eq. (22). Hence, following [15], we propose approximation of optimal temperature parameter λ_i . A second-order Taylor expansion around 0 of Eq. (21) yield:

$$\min_{\theta} \min_{\lambda \geq 0} \frac{1}{|C|} \sum_{c \in C} \frac{1}{n_+} \sum_{i \in I_c^+} \left\{ \lambda_i \cdot \rho + E_{j \sim P_0} [L(c, i, j)] + \frac{\text{Var}_j(L(c, i, j))}{2\lambda_i} + o_{\infty} \left(\frac{1}{\lambda_i} \right) \right\}, \quad (24)$$

where $\text{Var}_j(L(c, i, j))$ is defined in Eq. (18). Solving the above equation yields approximated optimal temperature parameter λ_i :

$$\lambda_i \approx \sqrt{\frac{\text{Var}_j(L(c, i, j))}{2\rho}} = \sqrt{\frac{\text{Var}_j(L(c, i, j))}{-2 \log \beta}}. \quad (25)$$

□

B PROOF OF THEOREM 3

PROOF. Suppose there are i ($i < K$) positive items in Top- K items of the permutation, and then we have $Recall@K = i/K$. Under this condition, easily, we can find out the case which has the maximum value of $OPAUC(\beta)$, where $\beta = \frac{K}{N_-}$:

$$\underbrace{+\dots+}_{i} \underbrace{-\dots-}_{K-i} \mid \underbrace{+\dots+}_{N_+ - i} \underbrace{-\dots-}_{N_- - K + i}$$

Hence, the maximum value of $OPAUC(\beta)$ is $\frac{-i^2 + (N_+ + K)i}{N_+ N_-}$. Meanwhile, since i can only be integers, we derive that:

$$\frac{1}{N_+} \left[\frac{N_+ + K - \sqrt{(N_+ + K)^2 - 4N_+ N_- \times OPAUC(\beta)}}{2} \right] \leq Recall@K.$$

Similarly, we can find out the case which has the minimum value of $OPAUC(\beta)$:

$$\underbrace{-\dots-}_{K-i} \underbrace{+\dots+}_{i} \mid \underbrace{-\dots-}_{i} \underbrace{\dots}_{N_+ + N_- - K - i}$$

Hence, the minimum value of $OPAUC(\beta)$ is $\frac{i^2}{N_+ N_-}$. Since i can only be integers, we can also derive that:

$$Recall@K \leq \frac{1}{N_+} \left[\sqrt{N_+ N_- \times OPAUC(\beta)} \right].$$

These complete the proof of Eq. (19). Noticing that for a given permutation, $Precision@K = \frac{N_+}{K} \cdot Recall@K$, where $\frac{N_+}{K}$ is a constant. Hence, we can easily derive the Eq. (20). □

C METRICS

Suppose we sort the left items in descending order according to scores r_{cj} for each context c . The positive item sets are denoted as $I_{c, test}^+$. The detailed definitions of the widely-used metrics are summarized as follow:

- Precision@ K : metrics the fraction of positive items among the top K predicted items:

$$Precision@K = \frac{|\{j \in I_{c, test}^+ \mid Rank_j < K\}|}{K}.$$

- Recall@ K : metrics the fraction of all positive items that were recovered in the top K :

$$Recall@K = \frac{|\{j \in I_{c, test}^+ \mid Rank_j < K\}|}{|I_{c, test}^+|}.$$

- NDCG@ K metrics the quality of recommendation through discounted importance based on position:

$$NDCG@K = \frac{1}{\sum_{i=1}^{\min(|I_{c, test}^+|, K)} \frac{1}{\log_2(i+1)}} \sum_{j \in I_{c, test}^+} \frac{\mathbb{I}(Rank_j < K)}{\log_2(Rank_j + 1)}.$$

D PERFORMANCE ACROSS A WIDE RANGE OF K UNDER DIFFERENT SAMPLING DISTRIBUTION

In this subsection, we investigate how Top- K metrics across a wide range of K will change under different sampling distributions. For simplicity representation, we only investigate hyperparameters N in $DNS(M, N)$. As shown in Figure 11, we observe that:

- The lower the K in Top- K metrics is, the larger the N in $DNS(M, N)$ when the curve achieves its maximum performance.
- With a larger difference of K , there is a larger gap when the curve achieves its maximum performance.

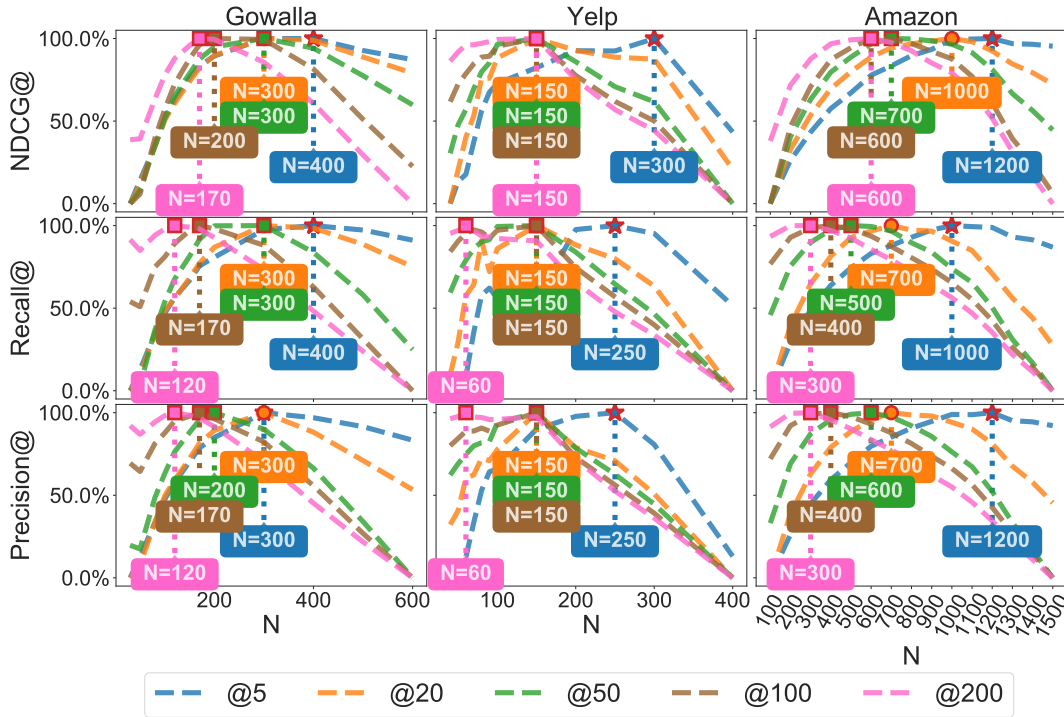


Figure 11: The effect of N in $\text{DNS}(M, N)$, where K is set to 5 for all three datasets.

Table 3: Performance comparison on three datasets using LightGCN. The best results are in bold and the second best are underlined. “**” denote the improvement is significant with t-test with $p < 0.05$.

Method	Gowalla		Yelp		Amazon	
	NDCG@50	Recall@50	NDCG@50	Recall@50	NDCG@50	Recall@50
BPR	0.1469	0.2470	0.0742	0.1507	0.0566	0.1307
PRIS(U)	0.1604	0.2677	0.0831	0.1670	0.0527	0.1221
PRIS(P)	0.1665	0.2753	0.0870	0.1741	0.0602	0.1369
AdaSIR(U)	0.1804	0.2979	0.0918	0.1808	0.0804	0.1719
AdaSIR(P)	0.1806	0.2974	0.0914	0.1796	0.0804	0.1712
DNS(*)	<u>0.1954</u> **	<u>0.3176</u> **	<u>0.0984</u> **	<u>0.1926</u> **	<u>0.1060</u> **	<u>0.2106</u> **
Softmax-v	0.1991 *	0.3209 **	0.1012 **	0.1974 **	0.1100 **	0.2134 **

E ADDITIONAL EXPERIMENTS WITH LIGHTGCN

As shown in Table 3, we conduct additional experiments on the LightGCN model, getting similar results. Our methods $\text{DNS}(M, N)$, $\text{Softmax-v}(\rho, N)$ significantly outperform BPR and HNS baselines, which is consistent with our analysis in Subsection 6.3.

F DISCUSSION

Our analysis for BPR loss can be generalized to other loss functions, like BCE loss, Triplet loss, Softmax loss, and InfoNCE loss, which

are widely applied in recommendation or other areas. Generally speaking, these loss functions have a high correlation with the AUC metric, and our conclusions also work for them. Theoretically, we have the following discussions. BCE optimizes a modified version of AUC [25]; BPR loss is a soft version of Triplet loss. Adjusting margin term in Triplet loss is equal to adjusting M in $\text{DNS}(M, N)$; $\text{Softmax-v}(\rho, N)$ is the upper bound of Softmax loss and InfoNCE loss. Adjusting the temperature in Softmax loss and InfoNCE loss is equal to adjusting ρ in $\text{Softmax-v}(\rho, N)$.