Exploiting Preferences in Loss Functions for Sequential Recommendation via Weak Transitivity

Hyunsoo Chung
Omnious AI
Seoul, South Korea
hyunsoo.chung@omnious.com

Jungtaek Kim
University of Pittsburgh
Pittsburgh, Pennsylvania, United States
jungtaek.kim@pitt.edu

Hyungeun Jo
Omnious AI
Seoul, South Korea
hyungeun.jo@omnious.com

Hyungwon Choi
Omnious AI
Seoul, South Korea
hyungwon.choi@omnious.com

Abstract

A choice of optimization objective is immensely pivotal in the design of a recommender system as it affects the general modeling process of a user's intent from previous interactions. Existing approaches mainly adhere to three categories of loss functions: pairwise, pointwise, and setwise loss functions. Despite their effectiveness, a critical and common drawback of such objectives is viewing the next observed item as a unique positive while considering all remaining items equally negative. Such a binary label assignment is generally limited to assuring a higher recommendation score of the positive item, neglecting potential structures induced by varying preferences between other unobserved items. To alleviate this issue, we propose a novel method that extends original objectives to explicitly leverage the different levels of preferences as relative orders between their scores. Finally, we demonstrate the superior performance of our method compared to baseline objectives.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Sequential recommendation, Loss functions, Transitivity

ACM Reference Format:
Hyunsoo Chung, Jungtaek Kim, Hyungeun Jo, and Hyungwon Choi. 2024. Exploiting Preferences in Loss Functions for Sequential Recommendation via Weak Transitivity. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679920

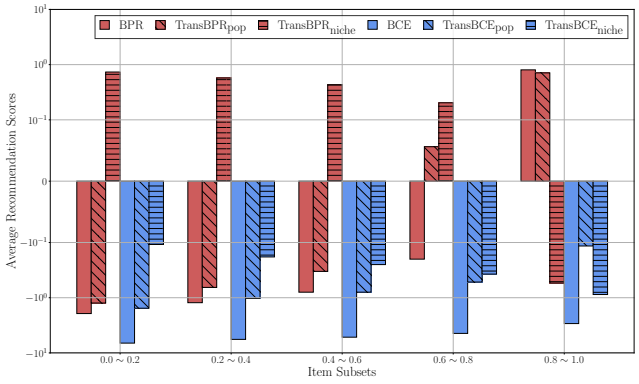


Figure 1: Average recommendation scores of item subsets on the Amazon Beauty dataset trained with BPR (red solid), BCE (blue solid), and our transitive extensions (hatch). Each item subset in x-axis equally consists of 20% of the total items divided based on their popularity. Bars with diagonal hatch assume that a user generally prefer popular items whereas bars with horizontal hatch regard niche items more preferred.

1 Introduction

Distinguishing items by relevance to an each user's previous interaction history is the essence of most applied sequential recommender systems. Conventional approaches [12, 23, 24, 30] frame this task as learning the feature representations of a chronologically-ordered list of user's interacted items and a candidate item. By computing the inner product between two vectors, we then obtain a recommendation score that quantitatively indicates the relevance. On that basis, a prevalent training process reformulates the problem into a supervised learning framework with binary labels, where the next interacted item of each user becomes a unique positive to its sequence of previous items. Such a procedure often accompanies negative sampling [3–5, 7, 15, 21] among the rest of items since the score computation of all items during each training step inevitably results in severe inefficiency [14, 33].

In this line of research, numerous approaches have adopted advanced neural networks as feature encoders [2, 8–10, 13, 16, 27, 28, 35, 36] to capture more complex correlations. Apart from vast architectural improvements, the majority of models yet utilize one of three types of loss functions as an optimization objective: pairwise, e.g., BPR [22], pointwise, e.g., BCE [1, 34], and setwise, e.g., SSM [29] functions. While the formulation of each function

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679920

varies, they all share the desired result of increasing the score of the unique positive than the scores of the other items. Notably, most recent studies on loss functions and negative sampling strategies [4, 7, 11, 25, 26, 31, 32] aim for improved training efficiency and robustness to false negatives within three core objectives. Despite their effectiveness, the model trained with such objectives consequently learns to regard unobserved items as equally negative with their labels simply set to zero. In real-world recommendation, however, a subset of negative items is occasionally more favorable than the other due to various side factors, e.g., item popularity. Unfortunately, the current scheme of binary label assignments hinders from fully taking advantage of diverse levels of preferences among unobserved items.

To address the aforementioned issue, we first derive an inductive relation, dubbed weak transitivity, which represents preference-driven orders of item scores. We then propose novel extensions of original loss functions that directly leverage this weak transitivity in their forms. Consequently, the recommendation scores of unobserved items are aligned with respect to their preferences. Figure 1 highlights such a property of resulting recommendation policy trained with our proposed family of objectives. Opposed to the results of BPR [22] and BCE [13], item scores from our transitive extensions $\text{TransBPR}_{\text{pop}}$ and $\text{TransBCE}_{\text{pop}}$ (horizontal hatch) are generally proportional to their popularity. Meanwhile, scores from $\text{TransBPR}_{\text{niche}}$ and $\text{TransBCE}_{\text{niche}}$ (diagonal hatch) are inversely proportional to their popularity, favoring more niche (i.e., unique) items. To cap it all, we serve the predefined preferences of items as an additional supervisory factor for their recommendation scores.

It is noteworthy that in this work we do not propose any new distributions for negative sampling but instead introduce the modification of original objectives. Hence, we solely utilize a combination of item popularity and uniform distributions for negative sampling, easily accessible in implicit settings. We validate the effectiveness of our proposed extensions compared to the original loss functions and their renowned variants on four sequential recommendation benchmarks. In all settings, our approaches substantially improve the recommendation performance compared to baseline methods.

2 Background

In this section, we describe a sequential recommendation task and the details of representative loss functions for recommendation.

2.1 Problem Statement

Let $\mathcal{U} = \{u_1, \dots, u_M\}$ and $\mathcal{I} = \{i_1, \dots, i_N\}$ denote a set of users and items where M and N are maximum numbers of users and items, respectively. Given a chronologically ordered history $h_u = \{i_1^u, \dots, i_t^u\}$ of observed items for a user u, a goal of sequential recommendation is to recommend the most relevant next item i_{t+1}^u. We first embed the history h_u and a candidate item i onto vectors h'_u and i', respectively. Accordingly, a recommendation score $\hat{s}_{ui}$ is calculated via an inner product between two embeddings.

2.2 Training Objectives

Pairwise Objective. Bayesian personalized ranking (BPR) [22] models personalized ranking of items. It forces the score of the positive (i.e., next interacted) item to be higher than the scores of the rest unobserved items. With a dataset $\mathcal{D}_s$ composed of a triplet (u, i, j) of which item i as a positive and item j as a sampled negative to a user u, the corresponding loss is formulated as follows:

$$\mathcal{L}_{\text{BPR}} = -\sum_{(u,i,j)\in\mathcal{D}_s} \log \sigma(\hat{s}_{ui} - \hat{s}_{uj}), \tag{1}$$

where $\sigma(\cdot)$ is a sigmoid function. A resulting term $\sigma(\hat{s}_{ui} - \hat{s}_{uj})$ is the probability of a user u preferring an item i more than an item j.

Pointwise Objective. Similar to the pairwise loss form, binary cross-entropy (BCE) [13] casts the recommendation problem into binary classification with a single sampled negative. Hence with a dataset $\mathcal{D}_s$ of a triplet (u, i, j), the loss is defined as below:

$$\mathcal{L}_{\text{BCE}} = -\sum_{(u,i,j)\in\mathcal{D}_s} \log(\sigma(\hat{s}_{ui})) + \log(1 - \sigma(\hat{s}_{uj})). \tag{2}$$

Setwise Objective. The sampled softmax loss (SSM) [29] turns the problem into a multi-class classification with a finite number of sampled negatives. With a dataset $\mathcal{D}_m$ of a triplet $(u, i, \mathcal{N}_j)$ consisting of the positive i and a set of multiple negative samples $\mathcal{N}_j$ to the user u, the loss optimizes the probability of the positive as:

$$\mathcal{L}_{\text{SSM}} = -\sum_{(u,i,\mathcal{N}_j)\in\mathcal{D}_m} \log \frac{\exp(\hat{s}_{ui})}{\exp(\hat{s}_{ui}) + \sum_{j\in\mathcal{N}_j} \exp(\hat{s}_{uj})}. \tag{3}$$

Table 1: Statistics of four preprocessed datasets.

Dataset	#Interactions	#Users	#Items	Density
Beauty	198,502	22,363	12,101	0.00073
Toys	167,597	19,412	11,924	0.00072
Sports	296,337	35,598	18,357	0.00045
Yelp	317,182	30,499	20,068	0.00052

3 Our Method

Here, we introduce weak transitivity between unobserved items, and then describe how to integrate it to original objectives.

3.1 Weak Transitivity

Inducing orders on recommendation scores of unobserved items requires a model to sample two or more negatives differing in preferences. In a basic form, let p_1 and p_2 represent two different sampling distributions for negatives. Given a user u and its positive i, we sample a negative j from p_1 and k from p_2 where $i \neq j \neq k$. Since the positive i is the most preferred item to the user u, a transitive relation $\hat{s}_{ui} > \hat{s}_{uj} > \hat{s}_{uk}$ holds true when j is more preferred than k. Strict transitivity then corresponds to a scheme where any j from p_1 is guaranteed to be more preferred than any k from p_2. However, when the sampled negative j is actually less preferable than k, the relation $\hat{s}_{ui} > \hat{s}_{uj} > \hat{s}_{uk}$ is violated. We refer this scheme as weak transitivity which allows such occasional violations.

The concept of transitivity with multiple negatives is straightforward as well. Instead of sampling a single negative from each distribution, we sample a set of negatives $\mathcal{N}_j = \{j_1, \dots, j_n\}$ from p_1 and $\mathcal{N}_k = \{k_1, \dots, k_n\}$ from p_2. Consequently, with $\hat{s}_{\mathcal{N}_j} = \{\hat{s}_{uj_1}, \dots, \hat{s}_{uj_n}\}$ and $\hat{s}_{\mathcal{N}_k} = \{\hat{s}_{uk_1}, \dots, \hat{s}_{uk_n}\}$, a transitive relation

Table 2: Quantitative results of different methods on public datasets in terms of HR and NDCG. The number of recommended items is fixed to 10. Reported metrics in bold are best performing methods whereas underlined numbers are second to the best.

Type	Loss	Beauty		Toys		Sports		Yelp	
		HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
Pairwise	BPR	0.0470	0.0213	0.0518	0.0244	0.0263	0.0122	0.0555	0.0328
	GBPR	0.0529	0.0246	0.0618	0.0281	0.0277	0.0121	**0.0607**	0.0346
	$\text{TransBPR}_{\text{pop}}$	**0.0674**	**0.0293**	0.0747	0.0338	**0.0397**	**0.0179**	0.0576	**0.0362**
	$\text{TransBPR}_{\text{niche}}$	0.0651	0.0289	**0.0777**	**0.0358**	0.0372	0.0171	0.0566	0.0355
Pointwise	BCE	0.0511	0.0229	0.0590	0.0274	0.0290	0.0130	0.0529	0.0306
	gBCE_1	0.0545	0.0245	0.0656	0.0297	0.0294	0.0135	0.0551	0.0319
	$\text{TransBCE}_{\text{pop}}$	**0.0730**	**0.0324**	**0.0805**	**0.0372**	**0.0413**	**0.0186**	0.0561	**0.0360**
	$\text{TransBCE}_{\text{niche}}$	0.0720	0.0322	0.0800	0.0369	0.0399	0.0179	**0.0569**	0.0358
Setwise	SSM	0.0656	0.0318	0.0673	0.0351	0.0381	0.0185	0.0616	0.0350
	InfoNCE	0.0632	0.0318	0.0742	0.0395	0.0363	0.0188	0.0483	0.0255
	BPR-DNS	0.0776	0.0356	0.0839	0.0386	0.0406	0.0187	0.0523	0.0349
	$\text{gBCE}_{\mathcal{N}}$	0.0725	0.0362	0.0783	0.0407	0.0415	0.0205	0.0632	0.0359
	$\text{TransSSM}_{\text{pop}}$	**0.0843**	**0.0395**	**0.0922**	**0.0439**	**0.0499**	**0.0229**	**0.0693**	**0.0418**
	$\text{TransSSM}_{\text{niche}}$	0.0774	0.0370	0.0874	0.0417	0.0399	0.0187	0.0642	0.0385

is modified as $\hat{s}_{ui} > \max(\hat{s}_{\mathcal{N}_j}) > \min(\hat{s}_{\mathcal{N}_j}) > \max(\hat{s}_{\mathcal{N}_k})$ with the same criteria for weak and strict cases.

3.2 Extensions with Weak Transitivity

We propose novel extensions of original training objectives that resolves the limitation of binary label assignments by incorporating the derived transitive relation to the loss formulation. We first introduce two sampling schemes by utilizing an item popularity distribution p_{pop} and a uniform distribution p_{unif}:

$$\mathcal{D}'_s(\text{pop}) = \{(u, i, j, k) \mid j \sim p_{\text{pop}}, k \sim p_{\text{unif}}\}, \tag{4}$$

$$\mathcal{D}'_s(\text{niche}) = \{(u, i, j, k) \mid j \sim p_{\text{unif}}, k \sim p_{\text{pop}}\}, \tag{5}$$

where j is the more preferred negative and k is the less preferred one. The mini-batch $\mathcal{D}'_s(\text{pop})$ assumes that a user generally prefers popular items whereas $\mathcal{D}'_s(\text{niche})$ regards niche items more favored. Our extension of the BPR objective with transitivity is then given by the following:

$$\mathcal{L}_{\text{TransBPR}} = -\sum_{(u,i,j,k)\in\mathcal{D}'_s} \underbrace{\log\sigma(\hat{s}_{ui} - \hat{s}_{uj})}_{\text{original}} + \gamma \underbrace{\log\sigma(\hat{s}_{uj} - \hat{s}_{uk})}_{\text{preference}}, \tag{6}$$

where γ is a balancing coefficient for two terms. In essence, the preference term encourages the score of j to be higher than that of k while the original term assures it to be smaller than that of i. As a consequence, our objective explicitly imposes the transitive relation $\hat{s}_{ui} > \hat{s}_{uj} > \hat{s}_{uk}$ to recommendation scores. Combining the proposed formulation with the previously introduced sampling schemes, we obtain two distinct training objectives as follows:

$$\text{TransBPR}_{\text{pop}} = \mathcal{L}_{\text{TransBPR}}(\mathcal{D}'_s(\text{pop}); \theta), \tag{7}$$

$$\text{TransBPR}_{\text{niche}} = \mathcal{L}_{\text{TransBPR}}(\mathcal{D}'_s(\text{niche}); \theta), \tag{8}$$

where θ is learnable parameters of the model. Similarly, the extension of the BCE function is formulated as below:

$$\mathcal{L}_{\text{TransBCE}} = -\sum_{(u,i,j,k)\in\mathcal{D}'_s} \Big[\log(\sigma(\hat{s}_{ui})) + \log(1-\sigma(\hat{s}_{uj})) + \gamma\Big(\log(\sigma(\hat{s}_{uj})) + \log(1-\sigma(\hat{s}_{uk}))\Big)\Big]. \tag{9}$$

Naturally, $\text{TransBCE}_{\text{pop}}$ and $\text{TransBCE}_{\text{niche}}$ are two consequent training schemes with $\mathcal{D}'_s(\text{pop})$ and $\mathcal{D}'_s(\text{niche})$, respectively. For the setwise loss function, we sample a set of items $\mathcal{N}_j$ and $\mathcal{N}_k$ from each distribution instead of sampling a single item j or k:

$$\mathcal{D}'_m(\text{pop}) = \{(u, i, \mathcal{N}_j, \mathcal{N}_k) \mid \mathcal{N}_j \sim p_{\text{pop}}, \mathcal{N}_k \sim p_{\text{unif}}\}, \tag{10}$$

$$\mathcal{D}'_m(\text{niche}) = \{(u, i, \mathcal{N}_j, \mathcal{N}_k) \mid \mathcal{N}_j \sim p_{\text{unif}}, \mathcal{N}_k \sim p_{\text{pop}}\}. \tag{11}$$

A corresponding extension for the SSM loss function is given by:

$$\mathcal{L}_{\text{TransSSM}} = -\sum_{(u,i,\mathcal{N}_j,\mathcal{N}_k)\in\mathcal{D}'_m} \left[\log\frac{\exp(\hat{s}_{ui})}{\exp(\hat{s}_{ui}) + \sum_{j\in\mathcal{N}_j}\exp(\hat{s}_{uj})} + \gamma\frac{1}{|\mathcal{N}_j|}\sum_{j\in\mathcal{N}_j}\log\frac{\exp(\hat{s}_{uj})}{\exp(\hat{s}_{uj}) + \sum_{k\in\mathcal{N}_k}\exp(\hat{s}_{uk})}\right]. \tag{12}$$

As similar to previous extensions, $\text{TransSSM}_{\text{pop}}$ and $\text{TransSSM}_{\text{niche}}$ employs either $\mathcal{D}'_m(\text{pop})$ or $\mathcal{D}'_m(\text{niche})$ for negative sampling.

One thing to note is that our introduced sampling strategies $\mathcal{D}'_s$ and $\mathcal{D}'_m$ are both weak transitive. Nonetheless, modifying them to a strict setting can be readily accomplished if we can quantify the preference of each item. For instance, we can reformulate the mini-batch construction for pairwise and pointwise objectives as:

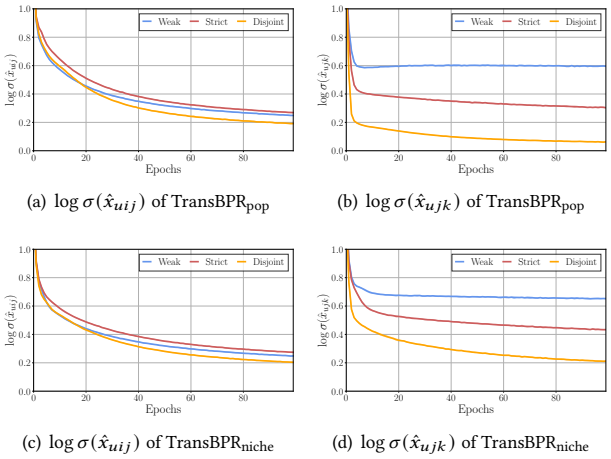
$$\mathcal{D}^{\dagger}_s(\text{pop}) = \{(u, i, j, k) \mid f(j) > f(k), j \sim p_{\text{pop}}, k \sim p_{\text{unif}}\}, \tag{13}$$

$$\mathcal{D}^{\dagger}_s(\text{niche}) = \{(u, i, j, k) \mid f(j) < f(k), j \sim p_{\text{unif}}, k \sim p_{\text{pop}}\}, \tag{14}$$

where $f(\cdot)$ denotes a function that measures the popularity of an item. However, we argue that such strict transitivity rather hurts the quality of the resulting recommendation policy.

Table 3: Comparison of our methods with strict transitivity and weak transitivity in the Amazon Beauty dataset.

Method		HR	NDCG
$\text{TransBPR}_{\text{pop}}$	Strict	0.0508	0.0229
	Weak	0.0674	0.0293
$\text{TransBPR}_{\text{niche}}$	Strict	0.0571	0.0252
	Weak	0.0651	0.0289
$\text{TransBCE}_{\text{pop}}$	Strict	0.0697	0.0311
	Weak	0.0730	0.0324
$\text{TransBCE}_{\text{niche}}$	Strict	0.0635	0.0282
	Weak	0.0720	0.0322
$\text{TransSSM}_{\text{pop}}$	Strict	0.0631	0.0309
	Weak	0.0843	0.0395
$\text{TransSSM}_{\text{niche}}$	Strict	0.0700	0.0346
	Weak	0.0774	0.0370



(a) $\log \sigma(\hat{x}_{uij})$ of $\text{TransBPR}_{\text{pop}}$ (b) $\log \sigma(\hat{x}_{ujk})$ of $\text{TransBPR}_{\text{pop}}$

(c) $\log \sigma(\hat{x}_{uij})$ of $\text{TransBPR}_{\text{niche}}$ (d) $\log \sigma(\hat{x}_{ujk})$ of $\text{TransBPR}_{\text{niche}}$

Figure 2: Average values of each loss term in TransBPR with Weak (blue), Strict (red), and Disjoint (yellow) transitivity.

4 Experiments

In this section, we conduct experiments to compare our proposed extensions to original objectives and their variants.

4.1 Experimental Setup

Datasets. We employ public sequential recommendation tasks from different domains: *Beauty*, *Toys*, and *Sports*, which are product review datasets introduced by Amazon.com [17], and *Yelp*, which is a widely tested business recommendation dataset. Detailed statistics of preprocessed datasets are reported in Table 1.

Evaluation Settings and Metrics. For dataset partitioning, we adopt the conventional *leave-one-out* strategy [13, 23] to assure the quality of the trained recommender system. Then, we recommend K items with the highest recommendation scores from the entire item pool. For evaluation, we adopt two common top-K metrics, HR@K and NDCG@K with K fixed to 10.

Baselines. We fix a model architecture to *SASRec* [13] and switch only a training objective. For baselines, we compare our method against *BPR* [22], *GBPR* [19], *BCE* [13], *SSM* [29], *InfoNCE* [18], *BPR-DNS* [31], and *gBCE* [20]. Here, gBCE_1 and $\text{gBCE}_{\mathcal{N}}$ denote gBCE with a single negative and multiple negatives, respectively.

Hyperparameters. For all objectives, We train with a fixed batch size of 256, a learning rate of 0.0003, and a maximum sequence length of 50. The SASRec model is with 2 layers and 1 attention head with an embedding dimension of 256. For setwise objectives, we sample 100 negatives in total. Our proposed TransSSM sets cardinality of $\mathcal{N}_j$ and $\mathcal{N}_k$ to 50 such that they sum to 100. A balancing coefficient γ is selected from $\{0.5, 1.0, 1.5\}$.

4.2 Performance Comparison

Table 2 summarizes the performance of models trained with different optimization objectives. In general, our proposed objectives outperform three original objectives and their notable variants in both metrics within all benchmarks. Particularly, we recognize $\text{TransSSM}_{\text{pop}}$ obtain the highest performance among all baselines in all datasets. Such results demonstrate that inducing preference order through weak transitivity consistently improves the performance regardless of base loss functions. Typically, we observe our extensions with popularity preference (e.g., $\text{TransSSM}_{\text{pop}}$) achieve commonly improved metrics compared to niche preference (e.g., $\text{TransSSM}_{\text{niche}}$). Hence, exploiting popularity as the preference indicator is particularly effective regardless of datasets. Though, we find recommendations with niche preference performs on par with best-performing baselines or often even better.

4.3 Transitivity Analysis

As seen in Table 3, we observe comparatively higher performance with weak transitivity. We hypothesize that the number of uninformative (i.e., easy) negatives [21, 26] increases more within the strict transitivity as training proceeds. To validate the claim, we report how average value of each term in the BPR extension, $\log \sigma(\hat{x}_{uij})$ and $\log \sigma(\hat{x}_{ujk})$, changes throughout the training with different schemes. Here, we compare three classes of transitivity: *Weak*, i.e., (4) and (5), *Strict*, i.e., (13) and (14), and *Disjoint*, i.e., a specific case of Strict that sets 50% of total items with high preference as a support for p_{pop} and the rest for p_{niche}. Results in Figure 2 illustrate more steep decline of the preference term, $\log \sigma(\hat{x}_{ujk})$, as the transitivity becomes more strict. Thus, weak transitivity provides the most informative gradients, increasing the training effectiveness.

5 Conclusion

In this work, we identified the common and crucial disadvantage of binary label assignments within conventional recommendation objectives. To overcome this limitation, we have proposed novel extensions that directly exploit the preference differences of unobserved items. Through extensive experiments, we demonstrated the effectiveness of our method with the thorough analysis. As future work, the uniformity of normalized embeddings [6] can be applied to our framework to improve recommendation performance more.

References

[1] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Virtual, 378–387.

[2] Mengru Chen, Chao Huang, Lianghao Xia, Wei Wei, Yong Xu, and Ronghua Luo. 2023. Heterogeneous graph contrastive learning for recommendation. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, Singapore, Singapore, 544–552.

[3] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, Halifax, Nova Scotia, Canada, 767–776.

[4] Xiao Chen, Wenqi Fan, Jingfan Chen, Haochen Liu, Zitao Liu, Zhaoxiang Zhang, and Qing Li. 2023. Fairly adaptive negative sampling for recommendations. In *Proceedings of the Web Conference (WWW)*. ACM, Autin, TX, USA, 3723–3733.

[5] Yongjun Chen, Jia Li, Zhiwei Liu, Nitish Shirish Keskar, Huan Wang, Julian McAuley, and Caiming Xiong. 2022. Generating Negative Samples for Sequential Recommendation. *arXiv preprint arXiv:2208.03645* August (2022), 1–11.

[6] Hyunsoo Chung and Jungtaek Kim. 2023. Leveraging Uniformity of Normalized Embeddings for Sequential Recommendation. In *Neural Information Processing Systems Workshop on Self-Supervised Learning - Theory and Practice (SSL-TP)*. NeurIPS, New Orleans, LA, USA, 1–9.

[7] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. Curran Associates, Virtual, 1094–1105.

[8] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential recommendation via stochastic self-attention. In *Proceedings of the Web Conference (WWW)*. ACM, Lyon, France, 2036–2047.

[9] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, Virtual, 433–442.

[10] 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 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Virtual, 639–648.

[11] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic Matrix Factorization with Non-Random Missing Data. In *Proceedings of the International Conference on Machine Learning (ICML)*. JMLR, Beijing, China, 1512–1520.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, San Juan, Puerto Rico, 1–10.

[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. IEEE, Singapore, Singapore, 197–206.

[14] Anton Klenitskiy and Alexey Vasilev. 2023. Turning Dross Into Gold Loss: is BERT4Rec really better than SASRec?. In *Proceedings of the ACM International Conference on Recommender Systems (RecSys)*. ACM, Singapore, Singapore, 1120–1125.

[15] Defu Lian, Qi Liu, and Enhong Chen. 2020. Personalized ranking with importance sampling. In *Proceedings of the Web Conference (WWW)*. ACM, Taipei, Taiwan, 1093–1103.

[16] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing GCN for recommendation. In *Proceedings of the Web Conference (WWW)*. ACM, Ljubljana, Slovenia, 1296–1305.

[17] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Shanghai, China, 43–52.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* July (2018), 1–13.

[19] Weike Pan and Li Chen. 2013. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI, Beijing, China, 2691–2697.

[20] Aleksandr Vladimirovich Petrov and Craig Macdonald. 2023. gsasrec: Reducing overconfidence in sequential recommendation trained with negative sampling. In *Proceedings of the ACM International Conference on Recommender Systems (RecSys)*. ACM, Singapore, Singapore, 116–128.

[21] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, New York, NY, USA, 273–282.

[22] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*. AUAI Press, Montreal, Quebec, Canada, 452–461.

[23] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, Beijing, China, 1441–1450.

[24] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, Los Angeles, CA, USA, 565–573.

[25] 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 *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Tokyo, Japan, 515–524.

[26] Peifeng Wang, Shuangyin Li, and Rong Pan. 2018. Incorporating gan for negative sampling in knowledge representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, New Orleans, LA, USA, 2005–2012.

[27] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Paris, France, 165–174.

[28] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Virtual, 726–735.

[29] Jiancan Wu, Xiang Wang, Xingyu Gao, Jiawei Chen, Hongcheng Fu, and Tianyu Qiu. 2024. On the effectiveness of sampled softmax loss for item recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 1–26.

[30] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Proceedings of the ACM International Conference on Recommender Systems (RecSys)*. ACM, Virtual, 328–337.

[31] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Dublin, Ireland, 785–788.

[32] Yongqi Zhang, Quanming Yao, Yingxia Shao, and Lei Chen. 2019. NSCaching: simple and efficient negative sampling for knowledge graph embedding. In *International Conference on Data Engineering (ICDE)*. IEEE, Chicago, IL, USA, 614–625.

[33] Chang Zhou, Jianxin Ma, Jianwei Zhang, Jingren Zhou, and Hongxia Yang. 2021. Contrastive learning for debiased candidate generation in large-scale recommender systems. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, Virtual, 3985–3995.

[34] Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, and Xiaoling Wang. 2021. Temporal augmented graph neural networks for session-based recommendations. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, Virtual, 1798–1802.

[35] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, Virtual, 1893—-1902.

[36] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the Web Conference (WWW)*. ACM, Lyon, France, 2388–2399.