Soft Contrastive Learning for Implicit Feedback Recommendations

Zhen-Hua Zhuang[1,2] and Lijun Zhang[1,2](✉)

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{zhuangzh,zhanglj}@lamda.nju.edu.cn

[2] School of Artificial Intelligence, Nanjing University, Nanjing, China

Abstract. Collaborative filtering (CF) plays a crucial role in the development of recommendations. Most CF research focuses on implicit feedback due to its accessibility, but deriving user preferences from such feedback is challenging given the inherent noise in interactions. Existing works primarily employ unobserved interactions as negative samples, leading to a critical noisy-label problem. In this study, we propose SCLRec (Soft Contrastive Learning for Recommendations), a novel method to alleviate the noise issue in implicit recommendations. To this end, we first construct a similarity matrix based on user and item embeddings along with item popularity information. Subsequently, to leverage information from nearby samples, we employ entropy optimal transport to obtain the matching matrix from the similarity matrix. The matching matrix provides additional supervisory signals that uncover matching relationships of unobserved user-item interactions, thereby mitigating the noise issue. Finally, we treat the matching matrix as soft targets, and use them to train the model via contrastive learning loss. Thus, we term it soft contrastive learning, which combines the denoising capability of soft targets with the representational strength of contrastive learning to enhance implicit recommendations. Extensive experiments on three public datasets demonstrate that SCLRec achieves consistent performance improvements compared to state-of-the-art CF methods.

Keywords: Contrastive Learning · Implicit Recommendations · Collaborative Filtering

1 Introduction

In the era of information explosion, recommender system has become a crucial tool for enhancing user engagement and satisfaction by providing personalized suggestions for products [16], videos [6], among others. Collaborative filtering (CF) has been widely adopted in personalized recommendation systems [3,20,22], with the key idea that similar users tend to share similar preferences. Typically, CF models mainly rely on historical interactions to predict user interests for candidate items [26]. Most CF research [13,20] focuses on implicit

D.-N. Yang et al. (Eds.): PAKDD 2024, LNAI 14649, pp. 219–230, 2024.
https://doi.org/10.1007/978-981-97-2262-4_18

feedback which only contains user-item interactions (e.g., clicks, browsing history) because it encompasses a large volume of data and captures abundant collaborative information in a simple manner [27,28].

A persistent challenge for implicit recommendations is how to formulate the loss function based on implicit feedback [3]. In general, there are three popular types of loss functions in recommendation systems: pointwise loss [18], pairwise loss [20], and listwise loss [3]. Specifically, the contrastive learning loss [17], as a novel type of listwise loss, has been introduced to the implicit feedback recommendations due to its excellent representational capabilities [26,31]. Contrastive learning (CL) aims to learn feature representations by minimizing the distance between similar (matched) sample pairs and maximizing the distance between dissimilar (unmatched) pairs. Existing methods [26,31] assume that user-item pairs within observed interactions are matched, while user-item pairs within unobserved interactions are considered unmatched in implicit feedback. However, such hard labeling mechanism, which strictly classifies user-item pairs as either matched or unmatched, fails to account for the inherent ambiguity present in missing feedback within implicit datasets. Specifically, unobserved interactions might not indicate disinterest, but simply that the items have not been exposed to the user. Thus directly fitting implicit feedback without addressing the noise issue cannot yield optimal user representations, leading to performance degradation.

Inspired by the recent advancements in CL for noisy-label problems [4], we introduce a novel contrastive learning loss to mitigate the noise issue in implicit feedback recommendations. Our approach comprises three phases: First, we estimate a similarity matrix that captures the likeness between users and items in a batch based on their embeddings and item popularity information. Next, to utilize information from nearby samples, we leverage entropy-regularized optimal transport to obtain the matching matrix which reflects the matching degree for user-item interactions from the similarity matrix. In deviation from previous methods [26,31], our approach assigns non-zero matching values, i.e., soft targets, to unobserved user-item interactions. The soft targets provide additional supervisory signals to guide the learning of the recommendation system, uncovering unobserved user-item matching relationships and effectively alleviating the noise issue. Finally, we optimize the model using these soft targets via the contrastive learning loss, which we thus term as soft contrastive learning loss. We conduct extensive experiments on three real-world datasets and observe consistent performance improvements when optimizing a matrix factorization model using our proposed soft contrastive loss.

The main contributions of this work can be summarized as follows:

- To the best of our knowledge, SCLRec is the first work that introduces soft contrastive learning into the recommendation domain.
- SCLRec assigns soft targets to unmatched user-item pairs, providing additional supervisory signals to identify latent user-item correspondences in unobserved interactions and effectively alleviate the noise problem.
- Experiments on three public datasets show that SCLRec achieves better performance than the state-of-the-art methods.

2 Related Work

Collaborative Filtering. CF plays a crucial role in recommender systems [22]. Most CF research focuses on implicit feedback, with a prominent approach being Bayesian Personalized Ranking (BPR) [20]. BPR uniformly samples unobserved interactions as negative samples, leading to a critical noisy-label problem. Existing denoising techniques [24,27,28,30] can be divided into two categories: sample selection methods and sample re-weighting methods. Sample selection methods choose clean and information-rich samples to train the model and enhance its performance. IR [28] represents a typical sample selection approach that iteratively creates pseudo-labels based on the disparity between labels and predictions to exclude noisy samples. However, sample selection methods, while effective in gathering cleaner data, rely on the sampling distribution, potentially resulting in biased gradient estimation and degrading recommendation performance. Conversely, sample re-weighting methods differentiate between clean and noisy interactions based on the model's learning process (e.g., loss values and predictions). T-CE [27] adopts a sample re-weighting strategy, dynamically assigning lower weights to samples exhibiting high loss values under the premise that noisy samples suffer larger losses. Yet, although these methods achieve promising results, they run the risk of neglecting hard clean samples and lack of adaptivity and universality [10].

Contrastive Learning. CL is a representative self-supervised learning (SSL) method, which measures the dependency of input variables by calculating their mutual information [1]. A prominent methodology in contrastive learning is the InfoNCE loss [17], which has been extensively applied in the fields of computer vision [2,4,19] and natural language processing [9]. The InfoNCE loss aims to minimize the distance between positive sample pairs while maximizing the distance between negative pairs, thereby facilitating effective representation learning. With the growing popularity of SSL, there have been efforts [14,26,31] to incorporate contrastive loss into recommendation systems. CLRec [31] employs the InfoNCE loss to address the exposure bias in CF and enhance deep candidate generation (DCG) in terms of fairness within large-scale recommendation scenarios. DirectAU [26] explores the desired alignment and uniformity properties of CF from the perspectives of contrastive representation learning. It works to push positive pairs closer to each other and make random pairs scatter across the unit hypersphere. However, these attempts have neglected the inherent noise issue in implicit recommendations. In contrast, we assign soft targets to unmatched user-item pairs, providing additional supervisory signals to alleviate the noise issue.

3 Methodology

In this section, we first introduce some notations related to collaborative filtering and InfoNCE loss [17]. Then we demonstrate the architecture and optimization process of our proposed SCLRec model.

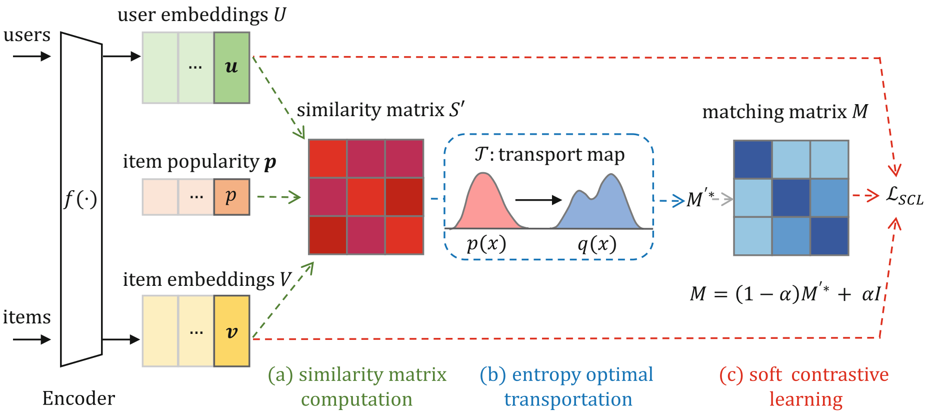


Fig. 1. Overview of the proposed SCLRec, which can mainly be divided into three parts: (a) similarity matrix computation, (b) entropy optimal transportation, and (c) soft contrastive learning.

3.1 Notations

Collaborative Filtering. Let $\mathcal{X}$ represent the set of users and $\mathcal{Y}$ denote the set of items. Given the observed user-item interactions $\mathcal{R} = \{(x, y) \mid x$ has interacted with $y\}$, the goal of CF methods is to estimate a score $s(x, y) \in \mathbb{R}$ for each unobserved interaction. The score indicates the likelihood that user x will interact with item y, and items with the highest scores for each user will be recommended. In general, most CF methods [12, 20] employ an encoder network $f(\cdot)$ that maps each user and item into a low-dimensional embedding. The embeddings are further l_2-normalized to the unit hypersphere, represented as $\widetilde{f(x)}, \widetilde{f(y)} \in \mathbb{R}^d$, where d is the dimension of the latent space. We denote $\widetilde{f(x)}$ as $\mathbf{u}$ and $\widetilde{f(y)}$ as $\mathbf{v}$, then the user embeddings within the batch are denoted as U and the item embeddings are represented by V. Finally, the predicted score is defined as the similarity between the user and item representation (e.g., dot product, $s(x, y) = \mathbf{u}^T\mathbf{v}$).

InfoNCE Loss. CLRec [31] trains the user and item encoders with contrastive learning to pull the matched user-item pairs closer and push the unmatched user-item pairs farther. This is achieved by minimizing the InfoNCE loss [17], which is defined as

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} I_{ij} \log \frac{\exp((\mathbf{u}_i^\top \mathbf{v}_j)/\tau)}{\sum_{k=1}^{N}\exp((\mathbf{u}_i^\top \mathbf{v}_k)/\tau)}, \tag{1}$$

where $(\mathbf{u}_i^\top \mathbf{v}_j)$ is the cosine similarity between two ℓ_2-normalized embedding vectors. τ represents a temperature parameter, while N denotes the batch size, i.e., the number of user-item pairs. I_{ij} is the element of an identity matrix I with $I_{ii} = 1, \forall i$ and $I_{ij} = 0, \forall i \neq j$. Note that $\mathbf{u}_i$ and $\mathbf{v}_j$ are on the unit hypersphere.

3.2 The SCLRec Framework

As mentioned above, there have been efforts [26,31] to incorporate contrastive learning loss into recommendations. However, these efforts have overlooked the inherent noise issue in implicit recommendations. Inspired by recent breakthroughs in CL for the noisy-label issue [4], we propose SCLRec, a novel method to alleviate the noise problem in implicit recommendations.

An overview of SCLRec is illustrated in Fig. 1, specifically, we first use the encoder to generate user and item embeddings. Then we construct a similarity matrix for unmatched users and items based on those embeddings and item popularity information. Subsequently, to utilize information from nearby samples, we employ entropy optimal transport to obtain the matching matrix for unmatched user-item pairs. After incorporating information from the matched pairs, we obtain the final matching matrix which serves as soft targets to provide additional signals for enhancing the recommendations. Finally, we adopt a modified contrastive learning loss as optimization objective, transforming the one-hot hard target I from Eq. (1) into a soft target M. Hence, it is denoted as the soft contrastive learning loss. In summary, our method primarily consists of three parts: similarity matrix computation, entropy optimal transportation and soft contrastive learning, detailed in the following sections.

Similarity Matrix Computation. Naturally, we consider harnessing embedding information to calculate similarity. To obtain reliable embeddings, we incorporate the Exponential Moving Average (EMA) method [23] to stabilize the encoder. This involves constructing a teacher encoder with the same model structure as the original encoder but with parameter updates following the EMA principle, i.e., $\tilde{\theta} \leftarrow m\tilde{\theta} + (1-m)\theta$, where θ and $\tilde{\theta}$ represent the weights of the original encoder and the teacher encoder, respectively, and m is momentum parameter set to 0.9. The user and item embeddings are generated by this teacher encoder. Then we utilize these embeddings and item popularity information to compute the similarity matrix as

$$S' = \gamma_u U^T U + \gamma_v V^T V + \gamma_p \mathbf{1}^T \mathrm{softmax}(\boldsymbol{p}) + U^T V - \eta I. \tag{2}$$

For users, the term $U^T U$ computes cosine similarity between user embeddings. Intuitively, it assumes that similar users might favor items liked by their counterparts. For items, $V^T V$ can be used to measure item similarity. Note that user and item embeddings could be regarded as representations of latent factor vectors across the dimensions of user preferences and item attributes, thus their similarities are inherently additive. Also, previous research [3] suggests that the popularity-based negative sampler usually exceeds the random negative sampler in implicit recommendations. Inspired by this, we utilize item popularity for computing the similarity. We obtain the item popularity distribution with $\boldsymbol{p} = [p_1, p_2, \cdots, p_N]$, where p_i denotes the frequency of item i for all users. Generally, more popular items are viewed as positive samples since they're more likely to be recommended. We then arrive at the formula

$\mathbf{1}^T\text{softmax}(\boldsymbol{p})$, using softmax$(\cdot)$ for normalizing the distribution. Specifically, $[\text{softmax}(\boldsymbol{p})]_i = \exp(p_i)/\sum_j \exp(p_j)$, converting the vector elements into a probability distribution. For user-item relationships, U^TV captures the affinity between user and item embeddings [21]. Additionally, the term $-\eta I$ with $\eta \rightarrow \infty$ ensures diagonal elements of S' are infinitely small. We calculate the similarity matrix in Eq. (2) by linearly weighting all the aforementioned terms.

Entropy Optimal Transportation. Next, we focus on generating the matching matrix from the similarity matrix S'. A naïve approach might be to directly use the similarity values as matching degrees. However, this is an oversimplified perspective because the similarity matrix is heuristically designed and might deviate from the actual scenario. Drawing inspiration from the application of optimal transport in noisy label scenarios [4,8], we utilize information from nearby samples to estimate accurate matching values. Specifically, the matching matrix is obtained by solving the following problem:

$$M'^* = \arg\max_{M'} \langle M', S' \rangle_F + \lambda H(M'). \tag{3}$$

Here, $\langle M', S' \rangle_F$ represents the Frobenius inner product of the matching matrix M' and the similarity matrix S', and it aims to establish a direct relationship between the similarity degree and the matching strength in user-item pairs. Specifically, a higher similarity degree yields a greater matching score. $H(M') = -\sum_{ij} M'_{ij} \log M'_{ij}$ serves as an entropy regularization term to enhance robustness. Moreover, the solution to Eq. (3) has a closed-form formulation:

$$M'^* = \text{Diag}(\mathbf{r}) \exp(S'/\lambda)\text{Diag}(\mathbf{c}), \tag{4}$$

where vectors $\mathbf{r}$ and $\mathbf{c}$ can be computed using the iterative Sinkhorn-Knopp algorithm [7]. Additionally, the algorithm ensures that the sum of each row and column in M'^* equals 1. As depicted in Eq. (4), the term $\exp(S'/\lambda)$ highlights the significance of entropy regularization. Similar to the temperature parameter in CL, increasing λ will make the distribution of M'^* more dispersed, while decreasing it will yield the opposite effect.

Soft Contrastive Learning. After generating the matching matrix M'^* for unmatched user-item pairs, we can then obtain the overall matching matrix M, which serves as the soft target. To incorporate the matching information of matched user-item pairs, we define M as a linear combination of the identity matrix I and M'^*:

$$M = \alpha I + (1 - \alpha)M'^*. \tag{5}$$

We define it for two reasons: First, $\alpha \in [0, 1]$ can represent the prior matching probability (degree) between user $\mathbf{u}_i$ and item $\mathbf{v}_i$. Second, the formula can ensure that the sum of each user's matching probability (degree) with all items within a batch remains 1, and likewise for each item with all users. Consequently, in the

InfoNCE loss, the one-hot hard target I can be substituted with the soft target M, resulting in a soft contrastive learning loss:

$$\mathcal{L}_{\mathrm{SCL}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\alpha I_{ij} + (1-\alpha)\, {M'}^{*}_{ij}\right)\log\frac{\exp((\mathbf{u}_i^{\top}\mathbf{v}_j)/\tau)}{\sum_{k=1}^{N}\exp((\mathbf{u}_i^{\top}\mathbf{v}_k)/\tau)}. \quad (6)$$

The formulation assigns soft targets to unmatched user-item pairs, providing additional supervision signals to better guide the learning process of recommendation systems.

4 Experiments

In this section, we evaluate our proposed model through comprehensive experiments and compare its results with current leading models on three public datasets. Our experiments are guided by the following research questions (RQs):

- **RQ1:** Does SCLRec outperform existing baselines in recommendation performance?
- **RQ2:** How do hyperparameters and components within SCLRec affect its performance?
- **RQ3:** Can SCLRec achieve denoising objectives effectively?

4.1 Experimental Settings

Datasets. We use three real-world datasets as outlined in Table 1: **MovieLens-10M** [11]: A well-known dataset containing movie ratings. **Gowalla** [5]: A dataset recording user check-in data from the Gowalla platform. **TmallBuy** [25]: An e-commerce dataset containing user purchase records on the Tmall platform. To construct implicit feedback, each entry is marked as 0/1 indicating whether the user rates the item. During preprocessing, we further ensure every user and item has at least 5 associated interactions.

Baselines. We compare the performance of SCLRec with various state-of-the-art CF methods:

General Methods: (1) **BPRMF** [20]: a typical approach optimizing matrix factorization via pairwise ranking loss; (2) **BUIR** [15]: a CF method that learns user and item embeddings only from positive interactions; (3) **LightGCN** [12]: a simplified graph convolution technique for CF.

CL-based Methods: (1) **CLRec** [31]: a method that uses the InfoNCE loss to mitigate the issue of exposure bias in CF; (2) **DirectAU** [26]: an approach that optimizes the properties of alignment and uniformity, inspired by contrastive representation learning.

Table 1. Statistics of datasets.

Dataset	#user ($\|\mathcal{X}\|$)	#item ($\|\mathcal{Y}\|$)	#inter. ($\|\mathcal{R}\|$)	avg. inter. per user	density
MovieLens	69.9k	10.2k	9998.9k	143.1	1.42%
Gowalla	29.9k	41.0k	1027.4k	34.4	0.08%
TmallBuy	413.1k	221.9k	4985.6k	12.1	0.02%

Denoising Methods: (1) **IR** [28]: a sample selection method iteratively relabels ambiguous samples to address noisy interactions; (2) **T-CE** [27]: a sample re-weighting method that employs the Truncated BCE loss to assign zero weights to examples with high losses beyond a dynamic threshold.

Evaluation Protocols. We partition the datasets into training, validation, and testing sets with an 8:1:1 ratio. The evaluation metrics are Recall@N and Normalized Discounted Cumulative Gain (NDCG)@N for N = 10, 20, 50.

Implementation Details. We use Adam as the default optimizer and early stop is adopted if NDCG@20 on the validation dataset continues to drop for 10 epochs. We set the embedding size to 64 and the learning rate to 10^{-3} for all the methods. The training batch size is set to 1024 and the weight decay is tuned among [0, 10^{-8}, 10^{-6}]. The default encoder in SCLRec is a simple embedding table that maps user/item IDs to embeddings.

4.2 Overall Performance (RQ1)

The overall recommendation performance of SCLRec and various baselines are presented in Table 2. From the results, we can draw several key findings: First, we observe consistent performance improvements when comparing the proposed SCLRec with recent baselines on the MovieLens, Gowalla, and TmallBuy datasets. Furthermore, our results show that the T-CE method outperforms most CL-based and general methods on the MovieLens and TmallBuy datasets, and the IR method excels on the Gowalla dataset in a similar manner. These findings demonstrate that denoising methods perform better than most other methods, particularly in sparse datasets that are susceptible to noise. This can be attributed to the inherent noise in implicit feedback, making effective denoising techniques especially beneficial. In addition, our results indicate that the best baseline is significantly influenced by the specific characteristics of individual datasets. For example, T-CE performs notably on the MovieLens and Tmallbuy datasets, while IR is more suited for user check-in data like Gowalla. However, only SCLRec, which combines contrastive learning representations with soft target denoising ability, consistently delivers superior results across all datasets.

Table 2. Top-K recommendation performance on three datasets. The best results are in boldface, and the best baselines are underlined.

Setting		Baseline Methods							Ours
Dataset	Metric	BPRMF	BUIR	LightGCN	CLRec	DirectAU	IR	T-CE	SCLRec
MovieLens-10M	Recall@10	0.1734	0.1885	0.1946	0.2071	0.2023	<u>0.2112</u>	0.2108	**0.2160***
	Recall@20	0.2606	0.2725	0.2856	0.2901	0.2937	0.2985	<u>0.3026</u>	**0.3148***
	Recall@50	0.4081	0.4073	0.4352	0.4370	0.4379	0.4401	<u>0.4413</u>	**0.4669***
	NDCG@10	0.2061	0.2322	<u>0.2427</u>	0.2402	0.2392	0.2408	0.2417	**0.2453***
	NDCG@20	0.2256	0.2467	0.2590	0.2595	0.2585	0.2587	<u>0.2598</u>	**0.2616***
	NDCG@50	0.2685	0.2831	0.3003	0.3019	0.2982	0.3016	<u>0.3021</u>	**0.3071***
Gowalla	Recall@10	0.0866	0.0798	0.1289	0.1215	<u>0.1394</u>	0.1388	0.1382	**0.1420***
	Recall@20	0.1263	0.1164	0.1871	0.1755	<u>0.2014</u>	0.2008	0.2011	**0.2078***
	Recall@50	0.2040	0.1917	0.2934	0.2813	0.3127	<u>0.3140</u>	0.3134	**0.3230***
	NDCG@10	0.0622	0.0570	0.0930	0.0868	<u>0.0991</u>	0.0980	0.0989	**0.1008***
	NDCG@20	0.0736	0.0676	0.1097	0.1022	0.1170	<u>0.1196</u>	0.1184	**0.1215***
	NDCG@50	0.0926	0.0858	0.1356	0.1281	0.1442	<u>0.1448</u>	0.1437	**0.1489***
TmallBuy	Recall@10	0.0366	0.0385	0.0455	0.0695	0.0696	<u>0.0709</u>	0.0701	**0.0730***
	Recall@20	0.0470	0.0571	0.0620	0.0958	0.0952	0.0953	<u>0.0970</u>	**0.1011***
	Recall@50	0.0668	0.0917	0.0937	<u>0.1390</u>	0.1368	0.1372	0.1388	**0.1476***
	NDCG@10	0.0268	0.0220	0.0299	0.0416	0.0422	0.0418	<u>0.0428</u>	**0.0440***
	NDCG@20	0.0296	0.0269	0.0342	0.0486	0.0490	0.0485	<u>0.0497</u>	**0.0514***
	NDCG@50	0.0337	0.0341	0.0409	0.0577	0.0577	0.0562	<u>0.0579</u>	**0.0612***

4.3 Ablation Study (RQ2)

As shown in Table 3, we analyze the impact of various hyperparameters and components on the performance of SCLRec. Evaluations are conducted using the Recall@50 and NDCG@50 metrics on the MovieLens test dataset.

Confidence in the Implicit Datasets: As outlined in Sect. 3.2, we define α as the matching probability between the corresponding user and item in observed interactions. This indicates the confidence or the noise level of matched pairs. Through testing values of 0.80, 0.90, and 0.99 for α, we find that both low confidence (0.80) and over-confidence (0.99) compromise performance.

Coefficients of the Similarity Matrix: The computation of the similarity matrix involves user similarity, item similarity, and item popularity. In order to verify their role, we set their coefficients $\gamma_u, \gamma_v, \gamma_p$ to zero respectively. Our results show that all components contribute positively. Notably, item popularity appears to be more crucial than the others, as omitting it results in a more pronounced decline in performance.

Implications of Optimal Transport: A key aspect of SCLRec is its use of entropy optimal transport to obtain the matching matrix based on the similarity matrix. The question arises: is entropy optimal transport truly necessary? We verify the requirement by setting the number of Sinkhorn iterations to 0 and 6. The experiments indicate that using 0 iterations results in lower performance, thus highlighting the effectiveness of entropy optimal transport in mitigating

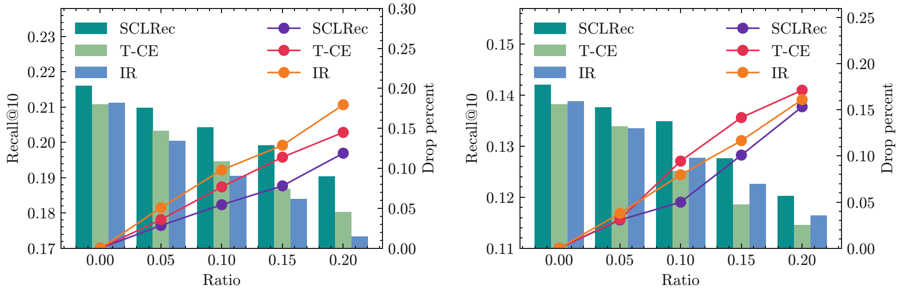
Table 3. Ablation study. SCLRec evaluated on MovieLens test set.

	α	γ_u	γ_v	γ_p	EMA	λ	#iter	Recall@50	NDCG@50
SCLRec	0.9	1.0	1.0	0.1	✓	0.1	6	**0.4669**	**0.3071**
α	0.80	1.0	1.0	0.1	✓	0.1	6	0.4283 (↓ 8.3%)	0.2848 (↓ 7.3%)
	0.99	1.0	1.0	0.1	✓	0.1	6	0.4368 (↓ 6.4%)	0.3014 (↓ 1.9%)
similarity matrix	0.9	0.0	1.0	0.1	✓	0.1	6	0.4496 (↓ 3.7%)	0.2945 (↓ 4.1%)
	0.9	1.0	0.0	0.1	✓	0.1	6	0.4487 (↓ 3.9%)	0.2927 (↓ 4.7%)
	0.9	1.0	1.0	0.0	✓	0.1	6	0.4380 (↓ 5.2%)	0.2921 (↓ 4.9%)
Sinkhorn	0.9	1.0	1.0	0.1	✓	0.05	6	0.4586 (↓ 1.8%)	0.2985 (↓ 2.8%)
	0.9	1.0	1.0	0.1	✓	0.2	6	0.4424 (↓ 5.2%)	0.2923 (↓ 4.8%)
	0.9	1.0	1.0	0.1	✓	0.1	0	0.4440 (↓ 4.9%)	0.2908 (↓ 5.3%)

data noise. We also investigate the impact of entropy regularization, dictated by λ in Eq. (4), and find that overly "hard" (0.05) or "soft" (0.2) target distribution both harm performance.

4.4 Robustness to Interaction Noises (RQ3)

To evaluate SCLRec's robustness to interaction noise, following the experimental settings of recent work [29], we incorporate specified ratios of unobserved interactions (i.e., 5%, 10%, 15%, and 20%) into the training set and test on an untouched test set. The results on MovieLens and Gowalla datasets are shown in Fig. 2. From the results, we can draw several key findings: Obviously, as the amount of added noise increases, the performance of all models declines. This is because CF models rely on user-item interactions for enhanced representations. Besides, on both datasets, SCLRec exhibits less performance degradation compared to other



(a) MovieLens (b) Gowalla

Fig. 2. Model performance w.r.t. noise ratio. The bar represents Recall@10, and the line shows the percentage of performance degradation.

models. Interestingly, the performance gap increases as noise levels rise on the MovieLens dataset. These observations indicate that our method, leveraging entropy optimal transport to determine soft user-item matches in contrastive loss, is effective in mitigating interaction noise. Moreover, it is worth noting that SCLRec shows stronger robustness on Movielens, which is consistent with MovieLens having denser interactions than Gowalla according to Table 1.

5 Conclusion

In this study, we introduce a novel soft contrastive learning method that improves implicit feedback recommendations. Specifically, we utilize entropy optimal transport to find soft user-item matches as labels for contrastive learning. Our proposed method provides additional supervisory signals to better guide the learning process of the recommendations. Furthermore, extensive experiments on three public evaluation datasets demonstrate that SCLRec achieves better performance compared to state-of-the-art CF methods.

Acknowledgements. This work was partially supported by NSFC (62122037) and the Collaborative Innovation Center of Novel Software Technology and Industrialization. We thank Zi-Hao Qiu for the helpful discussions.

References

1. Cao, J., Cong, X., Sheng, J., Liu, T., Wang, B.: Contrastive cross-domain sequential recommendation. In: CIKM, pp. 138–147 (2022)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML, pp. 1597–1607 (2020)
3. Cheng, M., et al.: Learning recommender systems with implicit feedback via soft target enhancement. In: SIGIR, pp. 575–584 (2021)
4. Cheng, R., Wu, B., Zhang, P., Vajda, P., Gonzalez, J.E.: Data-efficient language-supervised zero-shot learning with self-distillation. In: CVPR, pp. 3119–3124 (2021)
5. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: KDD, pp. 1082–1090 (2011)
6. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: RecSys, pp. 191–198 (2016)
7. Cuturi, M.: Sinkhorn distances: lightspeed computation of optimal transport. In: NIPS, pp. 2292–2300 (2013)
8. Damodaran, B.B., Flamary, R., Seguy, V., Courty, N.: An entropic optimal transport loss for learning deep neural networks under label noise in remote sensing images. Comput. Vis. Image Understand. **191**(C), 1–12 (2020)
9. Gao, T., Yao, X., Chen, D.: SimCSE: Simple contrastive learning of sentence embeddings. In: EMNLP, pp. 6894–6910 (2021)
10. Gao, Y., et al.: Self-guided learning to denoise for robust recommendation. In: SIGIR, pp. 1412–1422 (2022)
11. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**(4), 1–19 (2015)

12. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: simplifying and powering graph convolution network for recommendation. In: SIGIR, pp. 639–648 (2020)
13. He, X., Zhang, H., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: SIGIR, pp. 549–558 (2016)
14. Jiang, Y., Huang, C., Huang, L.: Adaptive graph contrastive learning for recommendation. In: KDD, pp. 4252–4261 (2023)
15. Lee, D., Kang, S., Ju, H., Park, C., Yu, H.: Bootstrapping user and item representations for one-class collaborative filtering. In: SIGIR, pp. 317–326 (2021)
16. McAuley, J., Targett, C., Shi, Q., van den Hengel, A.: Image-based recommendations on styles and substitutes. In: SIGIR, pp. 43–52 (2015)
17. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
18. Pan, R., et al.: One-class collaborative filtering. In: ICDM, pp. 502–511 (2008)
19. Qiu, Z.H., Hu, Q., Yuan, Z., Zhou, D., Zhang, L., Yang, T.: Not all semantics are created equal: contrastive self-supervised learning with automatic temperature individualization. In: ICML, pp. 28389–28421 (2023)
20. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)
21. Sankar, A., Wang, J., Krishnan, A., Sundaram, H.: Beyond localized graph neural networks: an attributed motif regularization framework. In: ICDM, pp. 472–481 (2020)
22. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. In: Advances in Artificial Intelligence, pp. 1–19 (2009)
23. Tarvainen, A., Valpola, H.: Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results. In: NIPS, pp. 1195–1204 (2017)
24. Tian, C., Xie, Y., Li, Y., Yang, N., Zhao, W.X.: Learning to denoise unreliable interactions for graph collaborative filtering. In: SIGIR, pp. 122–132 (2022)
25. Tianchi: IJCAI-16 brick-and-mortar store recommendation dataset (2018). https://tianchi.aliyun.com/dataset/dataDetail?dataId=53
26. Wang, C., et al.: Towards representation alignment and uniformity in collaborative filtering. In: KDD, pp. 1816–1825 (2022)
27. Wang, W., Feng, F., He, X., Nie, L., Chua, T.S.: Denoising implicit feedback for recommendation. In: WSDM, pp. 373–381 (2021)
28. Wang, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q.: Implicit feedbacks are not always favorable: iterative relabeled one-class collaborative filtering against noisy interactions. In: MM, pp. 3070–3078 (2021)
29. Wu, J., et al.: Self-supervised graph learning for recommendation. In: SIGIR, pp. 726–735 (2021)
30. Yu, W., Qin, Z.: Sampler design for implicit feedback data by noisy-label robust learning. In: SIGIR, pp. 861–870 (2020)
31. Zhou, C., Ma, J., Zhang, J., Zhou, J., Yang, H.: Contrastive learning for debiased candidate generation in large-scale recommender systems. In: KDD, pp. 3985–3995 (2021)