Implicit Feedbacks are Not Always Favorable: Iterative Relabeled One-Class Collaborative Filtering against Noisy Interactions

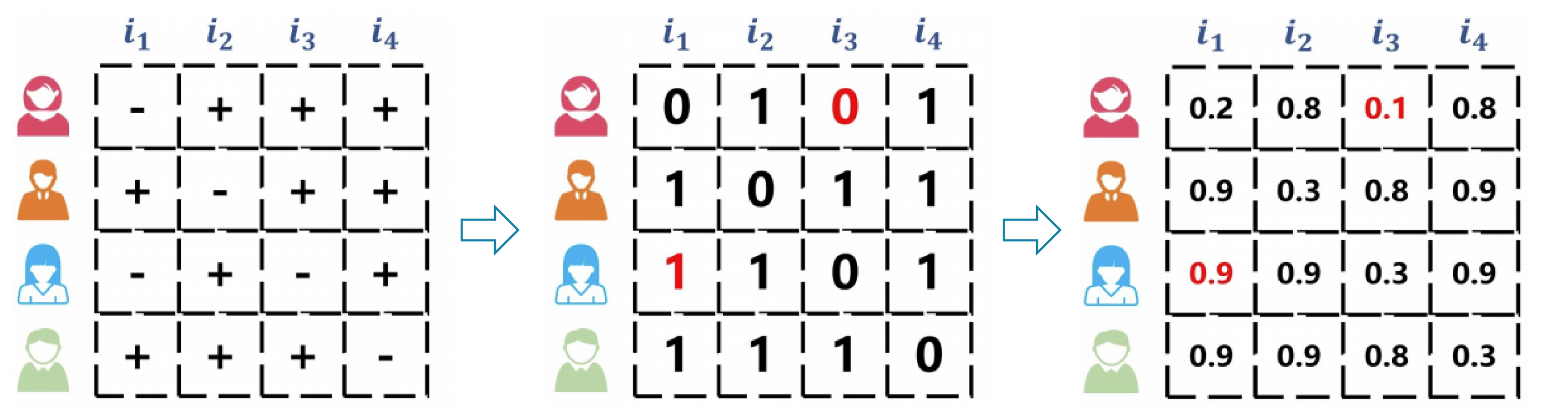
Zitai Wang[1,2], Qianqian Xu[3]*, Zhiyong Yang[1,2], Xiaochun Cao[1,4], Qingming Huang[3,5,6,7]*
[1]Institute of Information Engineering, CAS [2]SCS, UCAS [3]Institute of Computing Technology, CAS
[4]SCST, Sun Yat-sen University [5]SCST, UCAS [6]BDKM, CAS [7]Peng Cheng Laboratory
wangzitai@iie.ac.cn, xuqianqian@ict.ac.cn, {yangzhiyong, caoxiaochun}@iie.ac.cn, qmhuang@ucas.ac.cn

ACM multimedia
Chengdu, China OCT 20-24 2021

Motivation

Implicit feedback is inherently inconsistent with user preference due to its collection process:

- Unobserved interactions are a mixture of negative feedback and missing values.
- Observed interactions only represent the historical user behaviors, from which we cannot directly infer user preference.



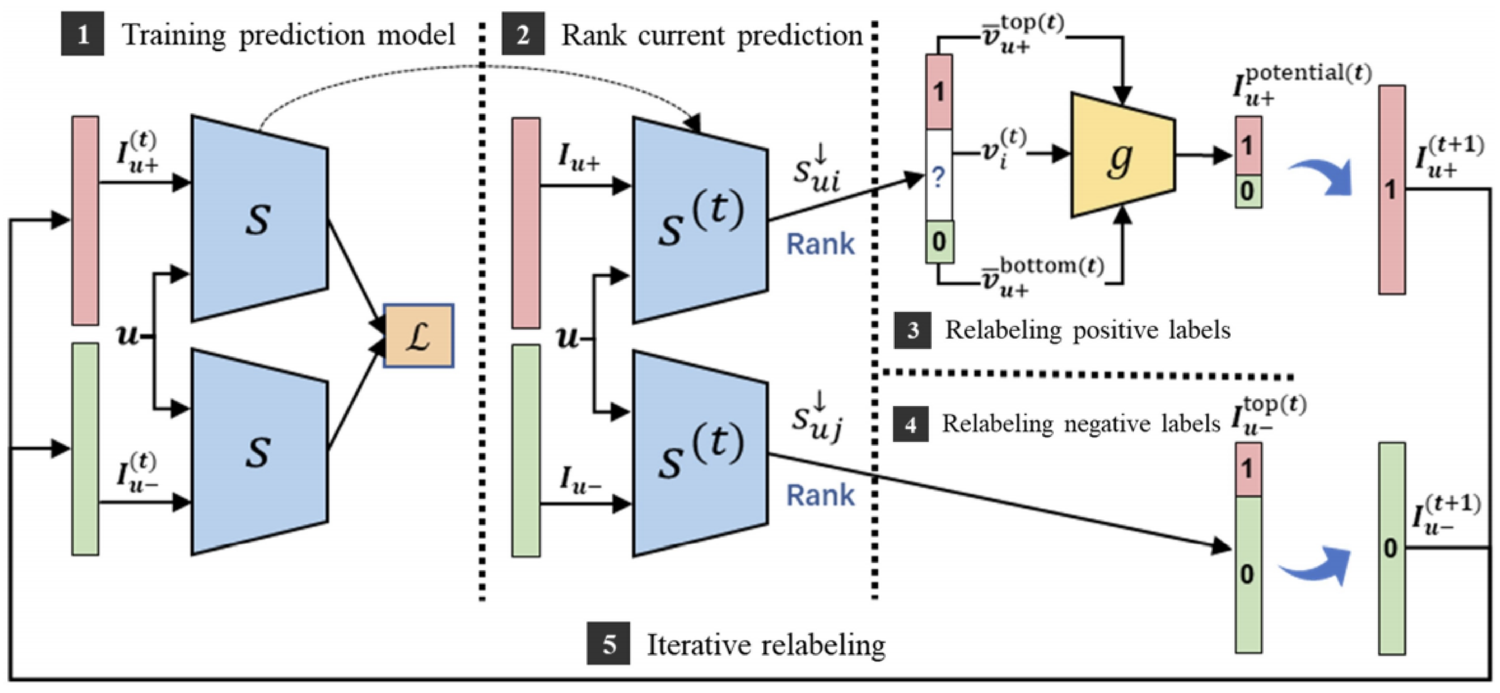
Ground-truth preference — Interaction matrix, where red means noise — Model predictions, Where red means error

Challenges are two-fold when handling both types of noise:

- No valid supervision information is available.
- No auxiliary information could help recognizes noise.

Solution: exploit the self-training principle to iteratively generate labels more consistent with user preference.

Method



Train prediction model

We train a prediction model based on the current label to evaluate the label confidence:

$$\min_{\theta} - \sum_{(u,i,j)\in\mathcal{T}} \log\sigma\Big(\underbrace{\boldsymbol{v}_u^T\boldsymbol{v}_i}_{\text{Observed}} - \underbrace{\boldsymbol{v}_u^T\boldsymbol{v}_j}_{\text{Unobserved}}\Big).$$

Relabel observed interactions

We first rank the observed items for each user and view the top/bottom items as confident positive and negative feedback:

$$s^{\downarrow}_{u,1} > s^{\downarrow}_{u,2} > \cdots > s^{\downarrow}_{u,|I_{u+}|} \quad I^{\text{top}}_{u+} = \{i : s_{u,i} \geq s^{\downarrow}_{u,n_+}\} \quad I^{\text{bottom}}_{u+} = \{i : s_{u,i} \leq s^{\downarrow}_{u,n_-}\}$$

Then, we could relabel the rest interactions based on the confident feedback. One simple strategy is prototype-based method classifier:

$$\hat{y}_{u,i} = g(\boldsymbol{v}_i^*) = \begin{cases} 1, & \alpha\|\boldsymbol{v}_i^* - \underbrace{\bar{\boldsymbol{v}}^{\text{top}}_{u+}}_{\text{Prototype of confident pos.}}\|_2 < \|\boldsymbol{v}_i^* - \underbrace{\bar{\boldsymbol{v}}^{\text{bottom}}_{u+}}_{\text{Prototype of confident neg.}}\|_2; \\ 0, & \text{otherwise}, \end{cases}$$

Relabel unobserved interactions

We mitigate the noise in unobserved interactions with similar method:

$$s^{\downarrow}_{u,1} > s^{\downarrow}_{u,2} > \cdots > s^{\downarrow}_{u,|I_{u-}|} \quad I^{\text{top}}_{u-} = \{i : s_{u,i} \geq s^{\downarrow}_{u,n_+^{un}}\} \quad I^{\text{bottom}}_{u-} = \{i : s_{u,i} \leq s^{\downarrow}_{u,n_-^{un}}\}$$

Iterative relabeling

As the labels become less noisy, the model predictions will be more consistent with user preference. Thus, we repeat the above steps to further eliminate the noise, and the labels generated in each epoch are defined as:

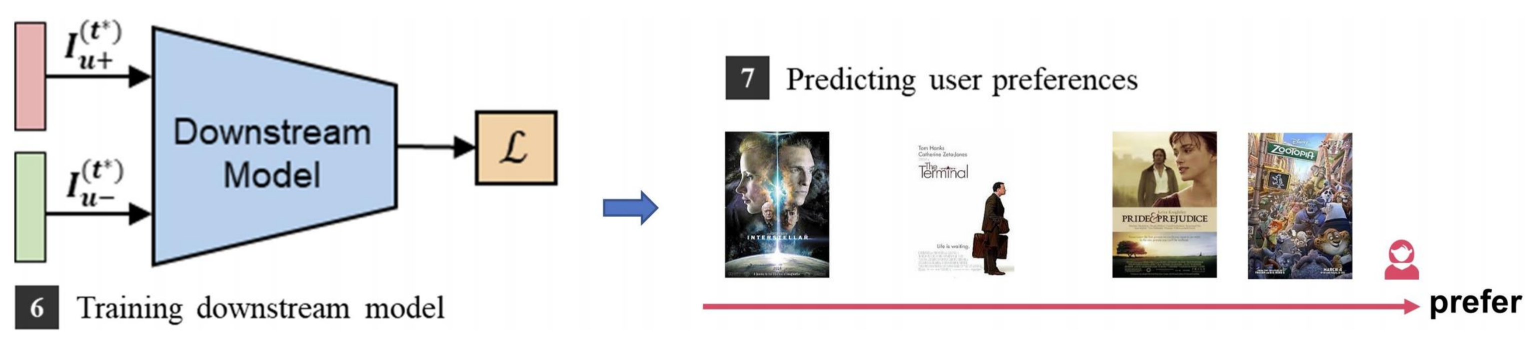
$$y^{(t)}_{u,i} = g_{\mu}(u,i) = \begin{cases} 1, & i \in I^{\text{top}(t)}_{u+} \cup I^{\text{potential}(t)}_{u+} \cup I^{\text{top}(t)}_{u-}; \\ 0, & otherwise, \end{cases}$$

where

$$I^{\text{potential}(t)}_{u+} = \{i : g(\boldsymbol{v}_i^{(t)}) = 1, i \in I_{u+}\setminus(I^{\text{top}(t)}_{u+} \cup I^{\text{bottom}(t)}_{u+})\}$$

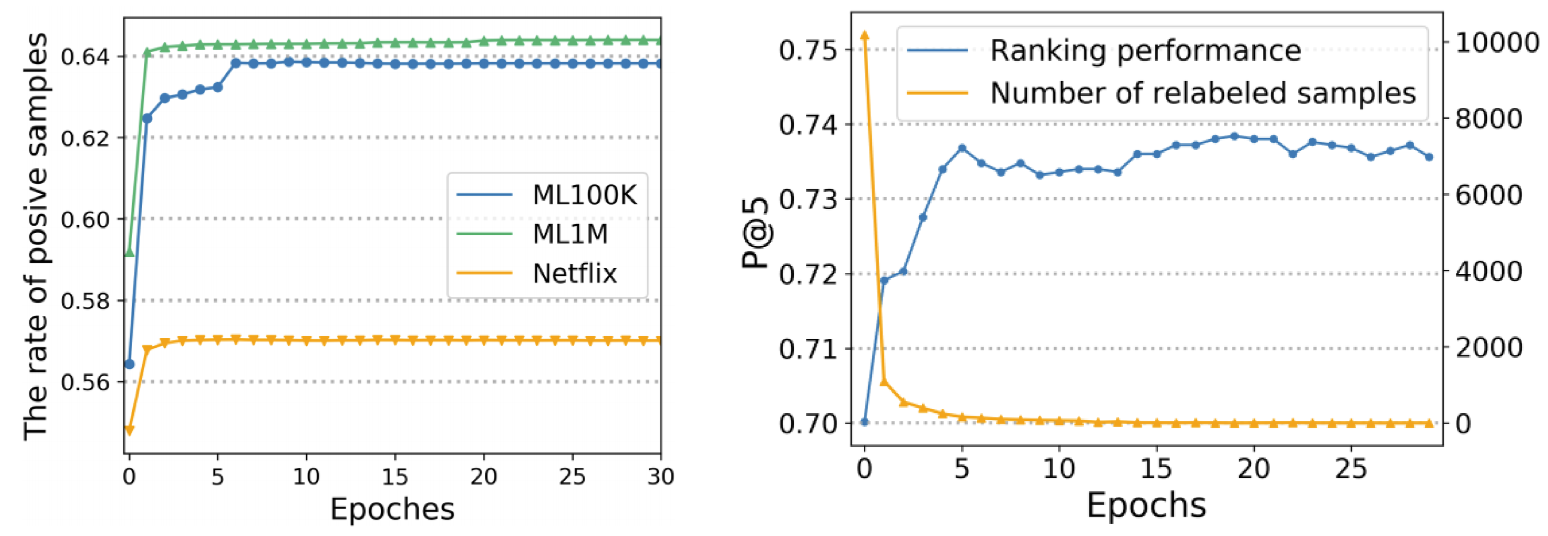
Train downstream model

Based on these refreshed labels, the downstream model could make better recommendation:



Experiment

The generated labels are more consistent with user preference



(L) The rate of positive samples increases gradually as the iterative relabeling process goes on.
(R) The preference improvement is consistent with the number of relabeled samples.

The generated labels boost the downstream models

Dataset	Relabeled	P@5	R@5	NDCG@5
ML100K	✓	0.7481	0.1214	0.7635
	×	0.7179	0.1178	0.7264
ML1M	✓	0.7861	0.0979	0.7969
	×	0.7762	0.0966	0.7791
Netflix	✓	0.7607	0.1368	0.7754
	×	0.7343	0.1315	0.7425

Dataset	Relabeled	P@5	R@5	NDCG@5
ML100K	✓	0.7513	0.1230	0.7727
	×	0.7167	0.1195	0.7347
ML1M	✓	0.7885	0.0977	0.7974
	×	0.7782	0.0970	0.7870
Netflix	✓	0.7301	0.1315	0.7448
	×	0.7180	0.1295	0.7306

The performance of downstream models: (L) BPR (R) MLP.

The proposed method outperforms the prior arts:

	P@1↑	P@5↑	P@10↑	R@5↑	R@10↑	NDCG@5↑	NDCG@10↑
DNS [44]	0.7606	0.7171	0.6825	0.1175	0.2201	0.7247	0.6980
NA [20]	0.7686	0.7111	0.6870	0.1190	0.2244	0.7225	0.7088
WRMF [25]	0.8048	0.7284	0.6855	0.1191	0.2187	0.7455	0.7099
iCD [2]	0.7706	0.7171	0.6982	0.1196	0.2245	0.7274	0.7104
SRRMF [4]	0.8129	0.7066	0.6730	0.1171	0.2195	0.7281	0.6967
T_CE [39]	0.7968	0.7272	0.6996	0.1212	0.2271	0.7413	0.7169
IR_{BPR}(Ours)	0.8229	0.7481	0.7040	0.1214	0.2272	0.7635	0.7269

Experimental results on MovieLens100K. The results on MovieLens1M and Netflix are similar.

P2B-08