Collaborative Similarity Embedding for Recommender Systems

†Chih-Ming Chen, *Chuan-Ju Wang, †Ming-Feng Tsai, *Yi-Hsuan Yang

†National Chengchi University
Taipei, Taiwan

*Academia Sinica
Taipei, Taiwan

CFDA & CLIP Labs

https://clip.csie.org/

MACLab

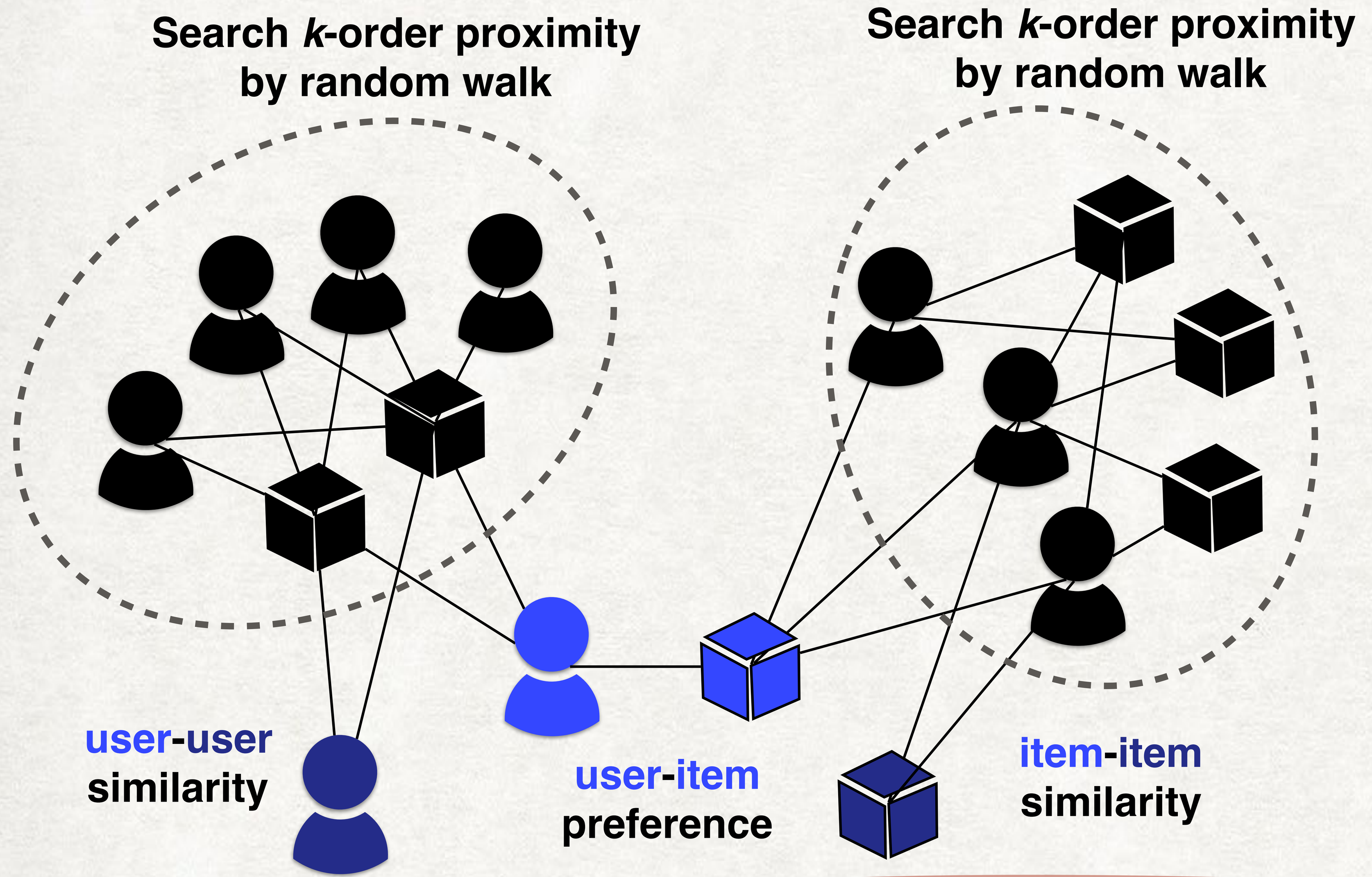
http://mac.citi.sinica.edu.tw/

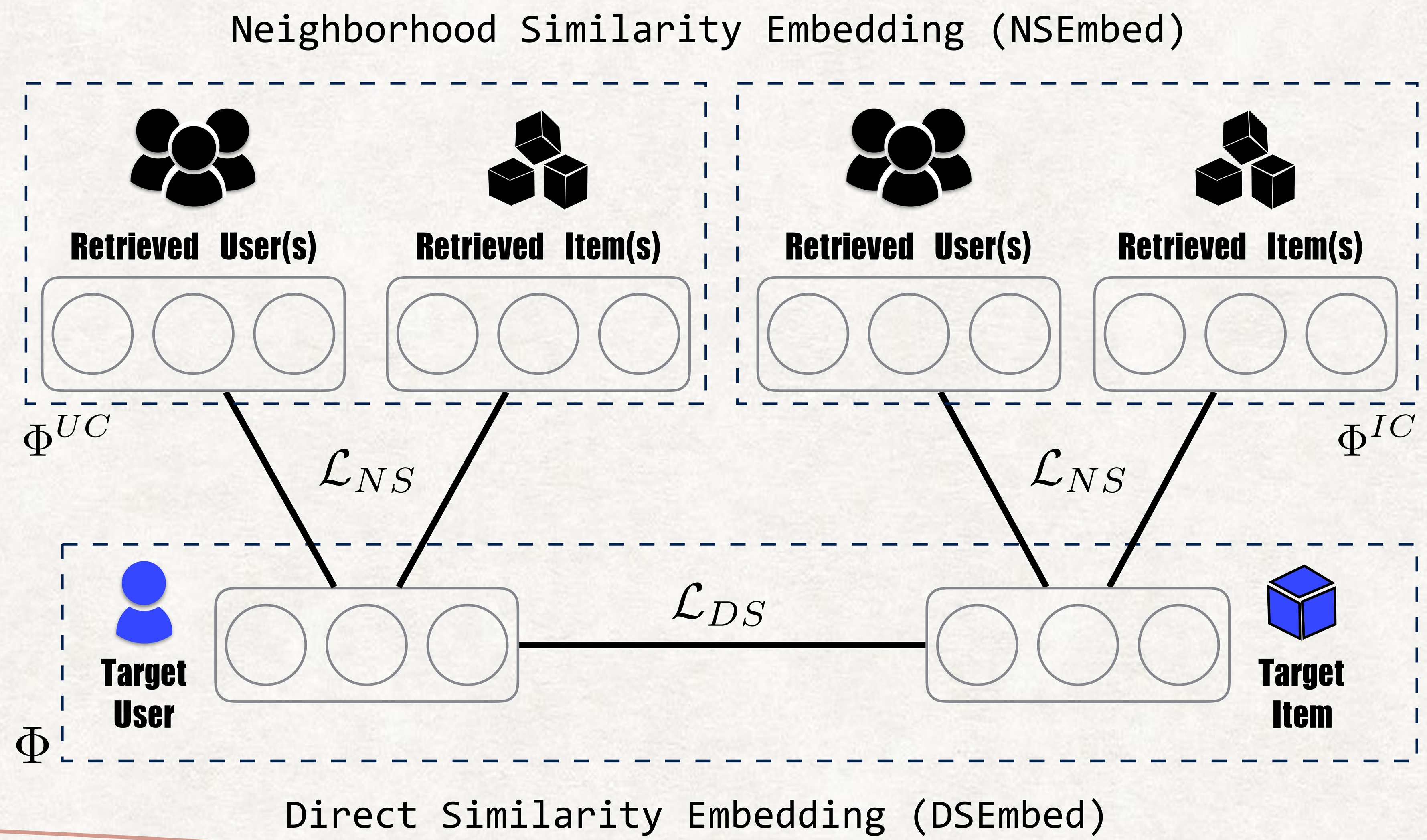
GitHub

https://github.com/cnclabs/proNet-core

Overview

Collaborative similarity embedding (CSE) is a unified framework that exploits comprehensive collaborative relations available in a user-item bipartite graph for representation learning and recommendation. It differentiate two types of proximity relations: **direct proximity** and **k-th order neighborhood proximity**. While learning from the former exploits direct user-item associations observable from the graph, learning from the latter makes use of implicit associations such as user-user similarities and item-item similarities, which can provide valuable information especially when the graph is sparse.





KL-divergence:

$$\begin{aligned}\mathcal{L}_{NS} \quad = \quad & \mathbb{E}_{(v_i,v_j)\sim S_U}\left[-\log p(v_j|v_i;\Phi;\Phi^{UC})\right] + \\ & \sum_M \mathbb{E}_{(v_i,v_j)\sim \bar{E}}\left[\log p(v_j|v_i;\Phi;\Phi^{UC})\right] + \\ & \mathbb{E}_{(v_i,v_j)\sim S_I}\left[-\log p(v_j|v_i;\Phi;\Phi^{IC})\right] + \\ & \sum_M \mathbb{E}_{(v_i,v_j)\sim \bar{E}}\left[\log p(v_j|v_i;\Phi;\Phi^{IC})\right].\end{aligned}$$

RATE-based Loss:

$$\begin{aligned}\mathcal{L}_{DS} \quad = \quad & \mathbb{E}_{(v_i,v_j)\sim E}\left[-\log p(v_i,v_j|\Phi)\right] + \\ & \sum_M \mathbb{E}_{(v_k,v_h)\sim \bar{E}}\left[\log p(v_k,v_h|\Phi)\right],\end{aligned}$$

RANK-based Loss:

$$\mathcal{L}_{DS} = \mathbb{E}_{(v_i,v_k)\sim \bar{E}}\left[\mathbb{E}_{(v_i,v_j)\sim E}\left[-\log p(v_j >_i v_k|\Phi)\right] \mid v_i\right]$$

Optimization mechanism:

$$\Theta \leftarrow \Theta - \alpha\left(\frac{\partial \mathcal{L}_{DS}}{\partial \Theta} + \lambda\left(\frac{\partial \mathcal{L}_{NS}}{\partial \Theta}\right) - \lambda_V \|\Phi\|\right)$$

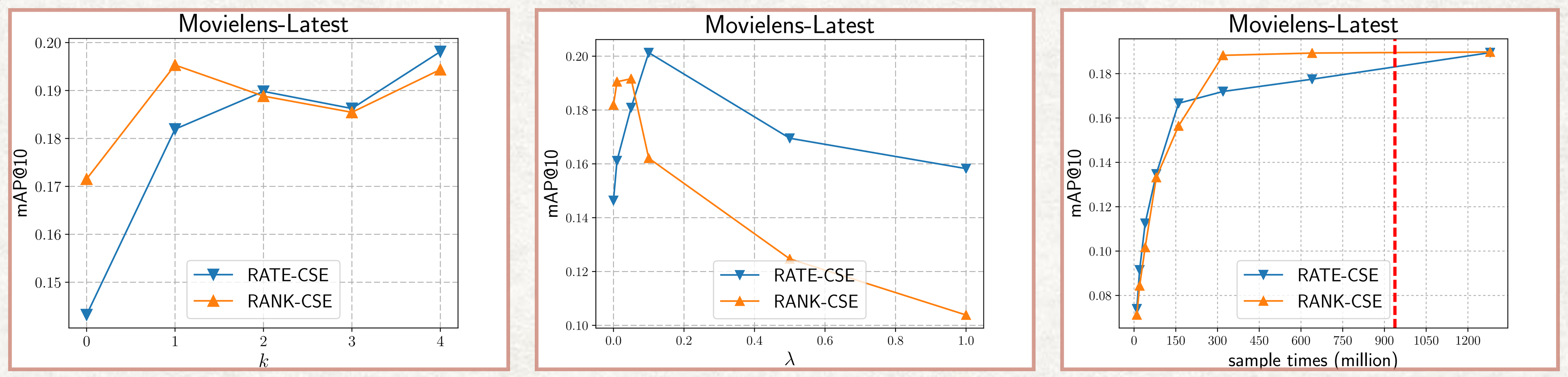
Performance

RATE-CSE and RANK-CSE denote two versions of our method that employ respectively rating-based and ranking-based loss functions for user-item associations. It can be observed that our method achieves the best results in terms of both Recall@10 and mAP@10 for most datasets. Moreover, RANK-CSE generally outperforms RATE-CSE in the experiments, reconfirming that using a ranking-based loss is indeed better for datasets with binary implicit feedbacks. Except for Frappe, RATE-CSE or RANK-CSE achieves significantly much better performance than the best performing baseline methods.

	Frappe		CiteULike		Netflix		MovieLens-Latest	
	Recall@10	mAP@10	Recall@10	mAP@10	Recall@10	mAP@10	Recall@10	mAP@10
Pop	0.1750	0.0708	0.0270	0.0114	0.0861	0.0359	0.0882	0.0289
DeepWalk [16]	0.0430	0.0256	0.0875	0.0458	0.0235	0.0112	0.0207	0.0061
WALS [9]	0.1632	0.1117	0.1851	0.0915	0.1214	0.0471	0.2350	†0.1682
BPR [19]	0.2785	0.1550	0.0861	0.0426	0.1496	0.0757	0.2163	0.1130
WARP [25]	0.3012	0.1796	0.1468	0.0813	†0.1887	†0.1004	†0.2712	0.1651
K-OS [26]	0.3018	0.1914	0.1356	0.0756	0.1783	0.0868	0.2522	0.1641
BiNE [5]	0.2159	0.1201	0.0422	0.0201	-	-	-	-
coFactor [13]	0.2110	0.1309	0.1323	0.0721	-	-	-	-
CML [8]	†0.3311	0.1958	0.1740	0.1008	0.1035	0.0444	0.1109	0.0957
WalkRanker [27]	0.3286	†**0.2099**	†0.2059	†0.1192	0.1090	0.0483	0.1307	0.0351
RATE-CSE	**0.3347**	0.2047	***0.2362**	***0.1452**	*0.2014	*0.1039	***0.3225**	***0.1990**
Improv. (%)	+1.0%	−2.4%	+14.7%	+21.9%	+6.7%	+3.5%	+18.9%	+18.3%
RANK-CSE	0.3155	0.2005	0.1993	*0.1228	***0.2156**	***0.1202**	*0.3094	*0.1902
Improv. (%)	−4.7%	−4.4%	−3.2%	+3.0%	+14.2%	+19.7%	+14.1%	+13.1%
	Last.fm-360K		**Amazon-Book**		**Epinions-Extend**		**Echonest**	
	Recall@10	mAP@10	Recall@10	mAP@10	Recall@10	mAP@10	Recall@10	mAP@10
Pop	0.0309	0.0133	0.0053	0.0015	0.0450	0.0246	0.0257	0.0104
WALS [9]	0.1621	0.0857	†0.0540	†0.0227	0.1479	0.0634	0.1287	†0.0638
BPR [19]	0.1120	0.0545	0.0248	0.0119	0.1126	0.0579	0.0499	0.0210
WARP [25]	0.1556	0.0832	0.0457	0.0199	†0.1509	†0.0775	0.1001	0.0447
K-OS [26]	†0.1641	†0.0888	0.0511	0.0215	0.1493	0.0766	†0.1249	0.0597
CML [8]	0.0496	0.0199	0.0129	0.0052	0.1171	0.0629	0.0357	0.0195
WalkRanker [27]	0.0233	0.0088	0.0080	0.0036	0.0560	0.0289	0.0309	0.0133
RATE-CSE	*0.1687	*0.0909	0.0540	0.0240	*0.1659	0.0788	0.1260	0.0605
Improv. (%)	+2.8%	+2.3%	+0.0%	+5.7%	+9.9%	+1.7%	+0.8%	−0.5%
RANK-CSE	***0.1762**	***0.0970**	***0.0625**	***0.0274**	***0.1767**	***0.0921**	***0.1358**	***0.0679**
Improv. (%)	+8.2%	+14.4%	+15.7%	+20.7%	+17.0%	+20.2%	+8.7%	+6.4%

Table 2: Recommendation performance. The † symbol indicates the best performing method among all the baseline methods; '*' and '%Improv.' denote statistical significance at $p < 0.01$ with a paired t-test and the percentage improvement of the proposed method, respectively, with respect to the best performing baseline.

Sensitivity



The first figure shows that increasing the order k of modeling neighborhood proximity between users or items improves the performance in general. The second Figure shows how the balancing parameter λ affects performance. The last one shows that the required total sample times for convergence is linear with respect to |E|.