Rethinking Contrastive Learning for Graph Collaborative Filtering: Limitations and a Simple Remedy

KAIST

Geon Lee, Sunwoo Kim, Kyungho Kim, and Kijung Shin

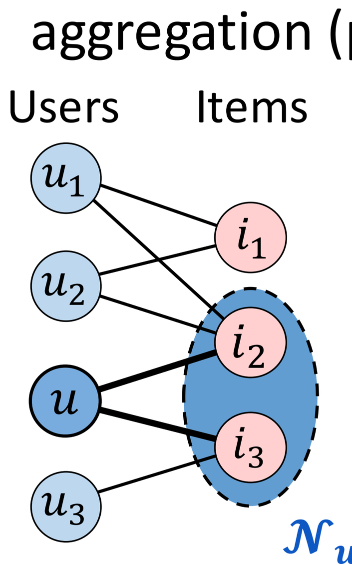
https://github.com/geon0325/NT-SSM

Summary

- **Summary** We revisit **contrastive learning** for **graph collaborative filtering** (**GCF**), showing that recommendation performance critically depends on **which multi-hop neighbor pairs are upweighted** during optimization & **how these updates vary across neighbor pair types**.
- **Analysis** We analyze GCF from the perspective of its prediction mechanism and identify **desirable learning dynamics in GCF**:
  - **Selective Upweighting**: Upweighting only a small subset of structurally similar neighbor pairs improves performance;
  - **Type-Aware Dynamics**: Optimal upweighting strategy differs across neighbor pair types (UU, II, UI, and IU).
- **Method** We present **NT-SSM**, a **simple** and **principled** contrastive learning objective for GCF that induces neighbor type-aware selective upweighting during optimization.
- **Experiments** **NT-SSM** demonstrates strong performance with:
  - **Accurate**: Improves NDCG@20 by up to 21.45% over SSM;
  - **General**: Consistently benefits LightGCN, SimGCL, and NCL;
  - **Effective**: Validates the importance of each component.

Graph Collaborative Filtering (GCF)

- **GCF Formulation** The embedding propagation and layer-wise aggregation (pooling) rules of GCF are defined as:



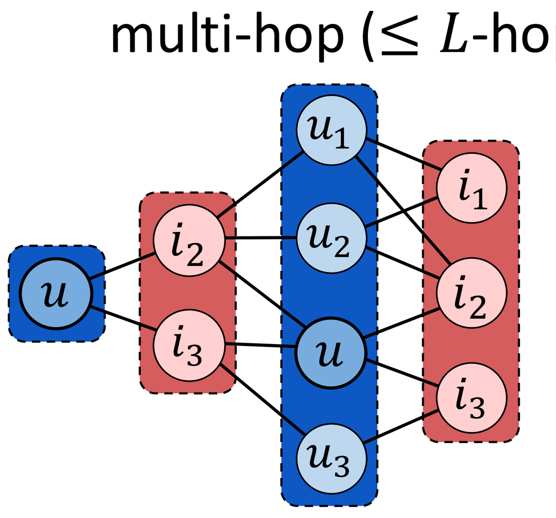
Propagation:

$$\mathrm{e}_u^{(\ell+1)} = \sum_{i \in \mathcal{N}_u} \frac{\mathrm{e}_i^{(\ell)}}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}$$

Pooling:

$$\mathrm{e}_u = \frac{1}{L+1} \sum_{\ell=0}^{L} \mathrm{e}_i^{(\ell)}$$

- **Multi-hop Neighbors** Let $\widetilde{\mathcal{N}}_u = \{v \in \mathcal{U} \cup \mathcal{I} \mid \tilde{\mathbf{S}} \neq 0\}$ be the set of multi-hop ($\leq L$-hop) neighbors of node u.



The multi-hop neighbor set can be **decomposed** into:

- $\widetilde{\mathcal{N}}_u^{(U)}$: **user-type** multi-hop neighbors
- $\widetilde{\mathcal{N}}_u^{(I)}$: **item-type** multi-hop neighbors

where $\widetilde{\mathcal{N}}_u^{(U)} \cup \widetilde{\mathcal{N}}_u^{(I)} = \widetilde{\mathcal{N}}_u$ and $\widetilde{\mathcal{N}}_u^{(U)} \cap \widetilde{\mathcal{N}}_u^{(I)} = \emptyset$.

The final embedding can be **decomposed** into contributions from **user-type** and **item-type** multi-hop neighbors:

$$\mathrm{e}_u = \sum_{v \in \widetilde{\mathcal{N}}_u} \tilde{\mathbf{S}}_{uv} \mathrm{e}_v^{(0)} = \sum_{v \in \widetilde{\mathcal{N}}_u^{(U)}} \tilde{\mathbf{S}}_{uv} \mathrm{e}_v^{(0)} + \sum_{v' \in \widetilde{\mathcal{N}}_u^{(I)}} \tilde{\mathbf{S}}_{uv'} \mathrm{e}_{v'}^{(0)}$$

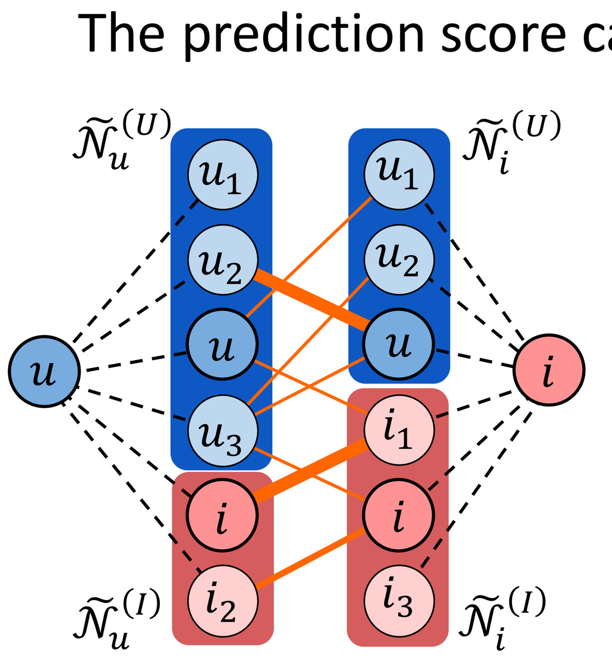
where $\tilde{\mathbf{S}} = \frac{1}{L+1} \sum_{\ell=0}^{L} \tilde{\mathbf{A}}^{\ell}$ is the multi-hop structural similarity matrix and $\tilde{\mathbf{A}} = \mathbf{D}^{-0.5} \mathbf{A} \mathbf{D}^{-0.5}$ is the normalized adjacency matrix.

- **Prediction** The score $\hat{r}_{ui} = \mathrm{e}_u^{\top} \mathrm{e}_i$ between user u and item i is:

$$\hat{r}_{ui} = \left(\sum_{v \in \widetilde{\mathcal{N}}_u} \tilde{\mathbf{S}}_{uv} \mathrm{e}_v^{(0)} \right)^{\top} \left(\sum_{v' \in \widetilde{\mathcal{N}}_i} \tilde{\mathbf{S}}_{iv'} \mathrm{e}_{v'}^{(0)} \right) = \sum_{v \in \widetilde{\mathcal{N}}_u} \sum_{v' \in \widetilde{\mathcal{N}}_i} \tilde{\mathbf{S}}_{uv} \tilde{\mathbf{S}}_{iv'} \underbrace{\left(\mathrm{e}_v^{(0)\top} \mathrm{e}_{v'}^{(0)} \right)}_{\text{Learnable Weight}}$$

The prediction score is determined by aggregating learnable weights over all pairs of multi-hop neighbors of users and items.

The prediction score can be decomposed into four sub-scores:



$$\hat{r}_{ui} = \hat{r}_{ui}^{(U,U)} + \hat{r}_{ui}^{(I,I)} + \hat{r}_{ui}^{(U,I)} + \hat{r}_{ui}^{(I,U)}$$

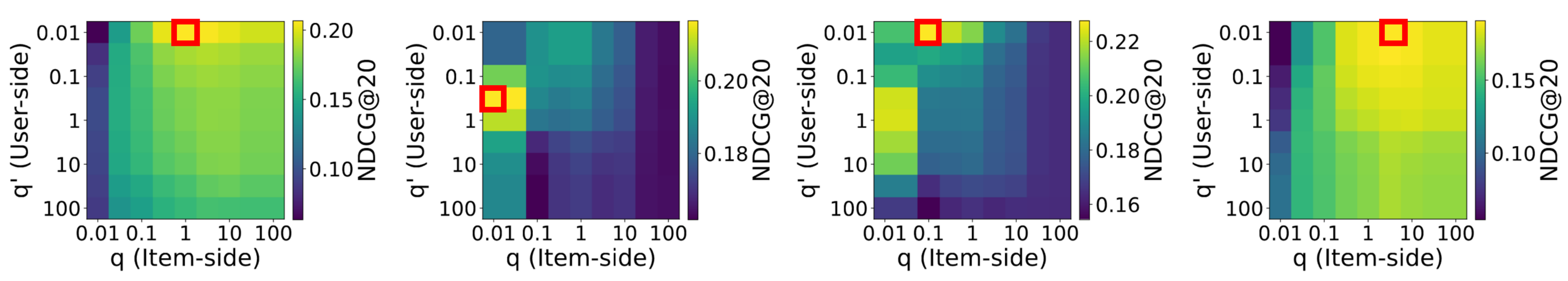
where $\hat{r}_{ui}^{(t,t')} = \sum_{v \in \widetilde{\mathcal{N}}_u^{(t)}} \sum_{v' \in \widetilde{\mathcal{N}}_i^{(t')}} \tilde{\mathbf{S}}_{uv} \tilde{\mathbf{S}}_{iv'} \left(\mathrm{e}_v^{(0)\top} \mathrm{e}_{v'}^{(0)} \right)$

GCF optimization:
Which neighbor pairs should be upweighted and which should be downweighted?

Learning Dynamics of GCF

- We selectively retain only the **top-q% multi-hop neighbors** with the highest structural similarity and measure the prediction performance:

$$\widetilde{\mathcal{N}}_u^{\langle q \rangle} = \{ v \in \widetilde{\mathcal{N}}_u \mid \tilde{\mathbf{S}} \geq \mathrm{top}_{q\%}(\tilde{\mathbf{S}}_{\cdot,v}) \}$$



Observation 1. *Using only a small subset of neighbor pairs with high structural similarity leads to improved performance.*

Observation 2. *The effect of selectively upweighting neighbor pairs varies across different neighbor pair types.*

Learning Dynamics of Contrastive Learning in GCF

- The **Sampled Softmax (SSM)** loss is defined as:

$$\mathcal{L}(i;u) = -\log \frac{\exp(s(u,i)/\tau)}{\exp(s(u,i)/\tau) + \sum_{j \in \mathcal{B}_u} \exp(s(u,j)/\tau)}$$

where $\mathcal{B}_u$ is the set of negative items.

- The **gradient** of the loss w.r.t. the learnable weight term $\mathrm{e}_v^{(0)\top} \mathrm{e}_{v'}^{(0)}$ associated with a neighbor pair (v, v') is:

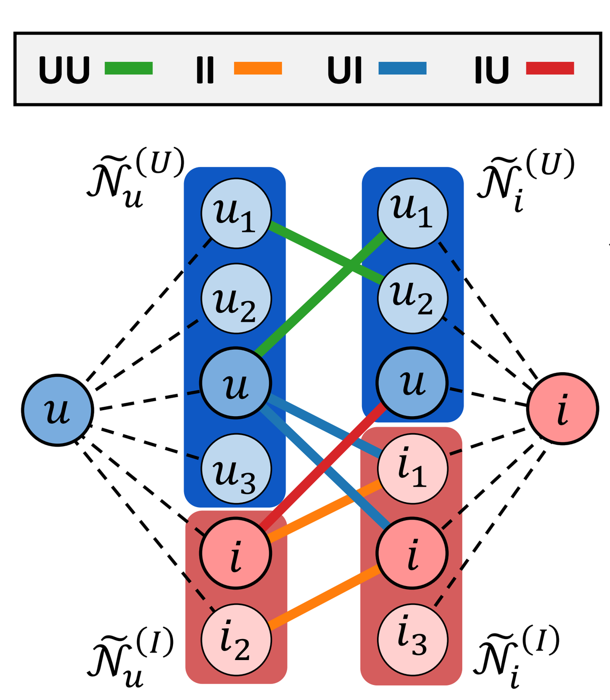
$$\frac{\partial \mathcal{L}(i;u)}{\partial \left(\mathrm{e}_v^{(0)\top} \mathrm{e}_{v'}^{(0)} \right)} = \frac{\tilde{\mathbf{S}}_{uv}}{\tau} \left(\mathbb{E}_{x \sim \pi_u}[\tilde{\mathbf{S}}_{xv'}] - \tilde{\mathbf{S}}_{iv'} \right) \Rightarrow \begin{cases} \text{if } \tilde{\mathbf{S}}_{iv'} \geq \mathbb{E}_{x \sim \pi_u}[\tilde{\mathbf{S}}_{xv'}] \rightarrow \text{Upweight} \\ \text{if } \tilde{\mathbf{S}}_{iv'} < \mathbb{E}_{x \sim \pi_u}[\tilde{\mathbf{S}}_{xv'}] \rightarrow \text{Downweight} \end{cases}$$

where π_u is the model-induced distribution over the items.

Limitation 1. *The learning dynamics of neighbor pair weights are centered primarily on items' neighbors.*

Limitation 2. *The learning dynamics apply the same upweighting and downweighting criterion across different types of neighbor pairs.*

Proposed CL Objective: NT-SSM



- To address the limitations, we introduce **NT-SSM**, a neighbor-type-aware CL loss:

$$\tilde{\mathcal{L}}(i;u) = -\log \frac{\exp(s(u,i)/\tau)}{\exp(s(u,i)/\tau) + \sum_{j \in \mathcal{B}_u} \exp(\tilde{\boldsymbol{s}}(\boldsymbol{u},\boldsymbol{j})/\tau)}$$

where the *negative similarity* is defined as:

$$\tilde{s}(u,j) = \alpha_I^{(U)} \hat{r}_{uj}^{(\cdot,U)} + \alpha_I^{(I)} \hat{r}_{uj}^{(\cdot,I)}$$

- The full loss is defined as *bidirectional*:

$$\mathcal{L}(u,i) = \tilde{\mathcal{L}}(i;u) + \tilde{\mathcal{L}}(u;i)$$

Property 1. *NT-SSM induces neighbor pair weight update dynamics accounting for both user- and item-side neighbors.*

Property 2. *NT-SSM induces type-dependent weight update dynamics across different neighbor pair types.*

Experimental Results

- GCF models perform better when trained with **NT-SSM** than with SSM.

NDCG@20		LastFM	MovieLens	Yelp	Amazon
LightGCN	SSM	0.2404	0.2648	0.0530	0.0414
	NT-SSM	**0.2709**	**0.3216**	**0.0565**	**0.0421**
NCL	SSM	0.2573	0.2614	0.0536	0.0322
	NT-SSM	**0.2708**	**0.3132**	**0.0545**	**0.0414**

- Different neighbor pair types (UU, II, UI, and II) benefit from different degrees of upweighting.