Revisiting LightGCN: Unexpected Inflexibility, Inconsistency, and A Remedy Towards Improved Recommendation

Geon Lee, Kyungho Kim, and Kijung Shin (KAIST)

{geonlee0325, kkyungho, kijungs}@kaist.ac.kr

https://github.com/geon0325/LightGCNpp

Summary

- **Analyses** We thoroughly examine the mechanisms of **LightGCN**: (1) scaling embeddings, (2) aggregating neighbors, and (3) pooling embeddings across layers.
- **Observations** We observe that LightGCN suffers from **inflexibility** and **inconsistency** when applied to real-world data.
- **A Remedy** We introduce **LightGCN++**, ***a remarkably simple remedy yet powerful*** that addresses the identified limitations of LightGCN.
  - **Flexible** norm scaling and neighbor weighting
  - **Adjustable** layer-wise embedding pooling
- **Experiments** LightGCN++ mitigates LightGCN's limitations and significantly improves the recommendation performance.
  - **Accurate:** Up to 17.81%↑ NDCG@20 compared to LightGCN
  - **Versatile:** Improves the performance in various domains
  - **Fast & Scalable:** Only 0.08% - 5.29% slower; Same space complexity

Background: LightGCN

- **LightGCN (He, Xiangnan, et al; SIGIR 2020)** is one of the most successful and widely used GNN-based recommendation models.
- **Aggregation** LightGCN adopts simple neighbor aggregation by removing feature transformation and non-linear activations.

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

- **Pooling** The intermediate embeddings from each layer are combined to construct the final user/item embeddings.

$$\mathrm{e}_u = \sum_{k=0}^{K} \omega_k \mathrm{e}_u^{(k)}, \text{ where } \omega_k = \frac{1}{K+1}$$

Analyses of LightGCN

- **Dual Effects of Normalization** The aggregation rule of LightGCN can be written as follows:

$$\mathrm{e}_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_u|}} \mathrm{e}_u^{(k)} = \underbrace{\frac{1}{\sqrt{|\mathcal{N}_i|}}}_{\text{Left term}} \sum_{u \in \mathcal{N}_i} \underbrace{\frac{1}{\sqrt{|\mathcal{N}_u|}}}_{\text{Right term}} \mathrm{e}_u^{(k)}$$

Left term Right term

- The **left term** scales the norm of the aggregated embedding.

$$\left\|\mathrm{e}_i^{(k+1)}\right\| = \frac{1}{\sqrt{|\mathcal{N}_i|}} \left\| \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}} \mathrm{e}_u^{(k)} \right\|$$

- The **right term** *explicitly* determines the "influence" of each neighbor.
  - The **norm of the neighbor embedding** *implicitly* affects the influence.

$$\mathrm{e}_i^{(k+1)} = \frac{1}{\sqrt{|\mathcal{N}_i|}} \sum_{u \in \mathcal{N}_i} \underbrace{\frac{\left\|\mathrm{e}_u^{(k)}\right\|}{\sqrt{|\mathcal{N}_u|}}}_{\text{Effective weight}} \frac{\mathrm{e}_u^{(k)}}{\left\|\mathrm{e}_u^{(k)}\right\|}$$

Effective weight

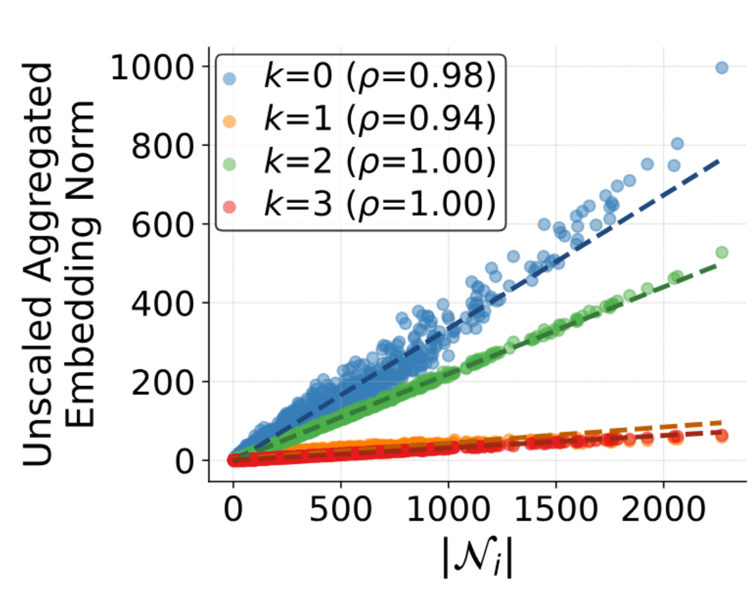
  - Essentially, the **effective weight** of the neighbor accurately describes the actual weight of the neighbor.

Primary Observation

- We observe a **near-linear relationship** between **the norms of the unscaled aggregated neighbor embeddings** and **the number of neighbors**, for all layers $k \geq 0$.

$$\left\| \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}} \mathrm{e}_u^{(k)} \right\| \propto |\mathcal{N}_i|$$

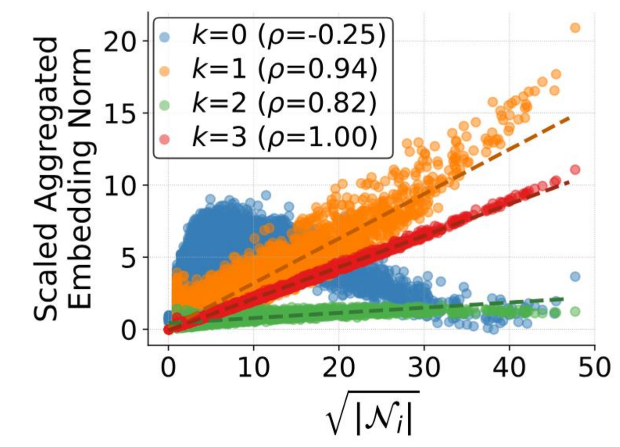
Norm of the aggregated neighbor embeddings — Number of neighbors



Observations on LightGCN

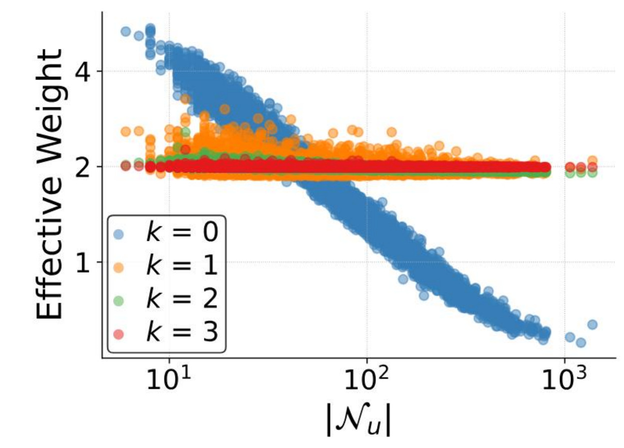
- **Q1.** How does LightGCN **scale embedding norms**?

$$\left\|\mathrm{e}_i^{(k+1)}\right\| = \frac{1}{\sqrt{|\mathcal{N}_i|}} \left\| \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}} \mathrm{e}_u^{(k)} \right\| \propto \frac{1}{\sqrt{|\mathcal{N}_i|}} |\mathcal{N}_i| = \sqrt{|\mathcal{N}_i|}$$



  - **Inflexibility** in norm scaling at $k \geq 1$.
  - **Inconsistency** in norm scaling between embeddings at $k = 0$ & $k \geq 1$.
- **Q2.** How does LightGCN **aggregate neighbors**?
  - Since $\left\|\mathrm{e}_i^{(k)}\right\| \propto \sqrt{|\mathcal{N}_i|}$ for $k \geq 1$, the **effective weight** tends to be **uniform** across neighbors:

$$\frac{\left\|\mathrm{e}_u^{(k)}\right\|}{\sqrt{|\mathcal{N}_u|}} \approx constant \text{ for } k \geq 1$$



  - **Inflexibility** in neighbor weighting at $k \geq 1$.
- **Q3.** How does LightGCN **pool embeddings**?
  - Despite the **inconsistent** norm scaling at $k = 0$ and $k \geq 1$, LightGCN applies **identical weights** to embeddings across layers.

$$\mathrm{e}_u = \frac{1}{K+1}\mathrm{e}_u^{(0)} + \frac{1}{K+1}\mathrm{e}_u^{(1)} + \cdots + \frac{1}{K+1}\mathrm{e}_u^{(K)}$$

Proposed Remedy: LightGCN++

- **LightGCN++** addresses the **inflexibility** and **inconsistency** of LightGCN.
- **Aggregation** LightGCN++ introduces hyperparameters $\boldsymbol{\alpha}$ & $\boldsymbol{\beta}$:

$$\mathrm{e}_i^{(k+1)} = \frac{1}{|\mathcal{N}_i|^{\alpha}} \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_u|^{\beta}} \frac{\mathrm{e}_u^{(k)}}{\left\|\mathrm{e}_u^{(k)}\right\|}$$

  - $1/|\mathcal{N}_i|^{\alpha}$ offers **flexibility** in **norm scaling**.
  - $1/|\mathcal{N}_u|^{\beta}$ (= **effective weight**) offers **flexibility** in **neighbor weighting**.
- **Pooling** LightGCN++ introduces a hyperparameter $\boldsymbol{\gamma}$:

$$\mathrm{e}_u = \boldsymbol{\gamma}\mathrm{e}_u^{(0)} + (\mathbf{1} - \boldsymbol{\gamma}) \frac{1}{K} \sum_{k=1}^{K} \mathrm{e}_u^{(k)}$$

  - γ enables **adaptive** weighting between embeddings with inconsistent norm scaling properties.

Experimental Results

- **Accuracy** **LightGCN++** consistently outperforms LightGCN.
  - Furthermore, **SOTA methods** enhanced with **LightGCN++** outperform their counterparts with LightGCN.

Dataset	LastFM		MovieLens		Gowalla		Yelp		Amazon	
Metric	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN [7]	0.2523	0.2427	0.2392	0.3010	0.1683	0.1426	0.0553	0.0449	0.0367	0.0274
LightGCN++	0.2715**	0.2624**	0.2616**	0.3275**	0.1739**	0.1469**	0.0650**	0.0529**	0.0394**	0.0294**
Improvement	7.60%	8.11%	9.36%	8.80%	3.32%	3.01%	17.54%	17.81%	7.35%	7.29%
NCL [14]	0.2548	0.2453	0.2401	0.3027	0.1704	0.1430	0.0584	0.0475	0.0393	0.0293
NCL++	0.2721**	0.2632**	**0.2621****	**0.3285****	**0.1759****	**0.1478****	**0.0678****	**0.0553****	0.0424**	0.0315**
Improvement	6.78%	7.29%	9.16%	8.52%	1.87%	1.81%	16.09%	16.42%	7.88%	7.50%
SimGCL [31]	0.2602	0.2494	0.2584	0.3217	0.1703	0.1424	0.0650	0.0528	0.0415	0.0314
SimGCL++	0.2723**	0.2617**	0.2615**	0.3276**	0.1704	0.1431	0.0657**	0.0536**	0.0444**	0.0334**
Improvement	4.65%	4.93%	1.19%	1.83%	0.05%	0.49%	1.07%	1.70%	6.98%	6.36%
XSimGCL [28]	0.2614	0.2508	0.2600	0.3245	0.1678	0.1400	0.0651	0.0528	0.0397	0.0298
XSimGCL++	**0.2738****	**0.2638****	0.2613	0.3270*	0.1705**	0.1432**	0.0674**	0.0549**	**0.0454****	**0.0342****
Improvement	4.74%	5.18%	0.50%	0.77%	1.60%	2.28%	3.53%	3.97%	14.35%	14.76%

- **Versatility** **LightGCN++** enhances the accuracy across various domains.

Dataset	Youshu	NetEase	iFashion
CrossCBR [16]	0.1584	0.0359	0.0778
CrossCBR++	**0.1625****	**0.0361**	**0.0847****
Improv.	2.58%	0.55%	8.88%

Bundle Rec.

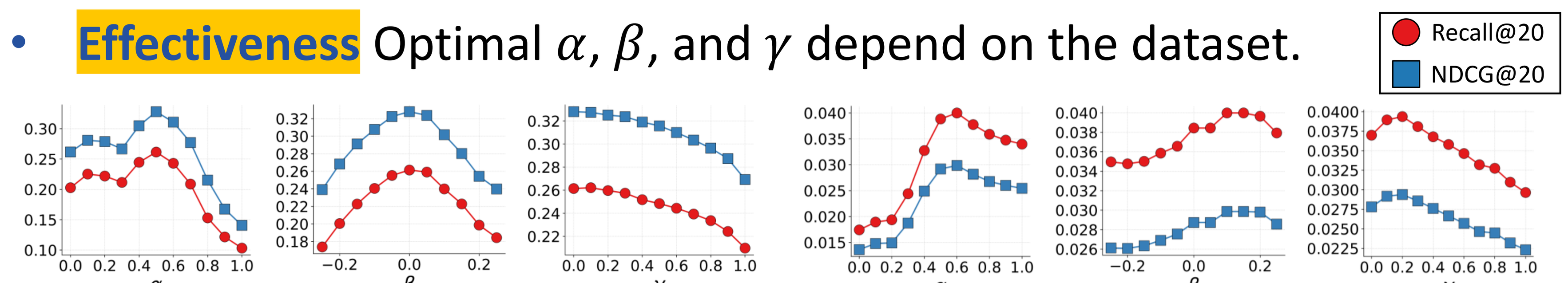
Dataset	Clothing	Sports	Baby
LATTICE [32]	0.0316	0.0428	0.0364
LATTICE++	**0.0322**	**0.0463****	**0.0402****
Improv.	1.90%	8.18%	10.44%

Multimedia Rec.

Dataset	Yelp	Amazon	MIND
KGCL [27]	0.0470	0.0737	0.0519
KGCL++	**0.0475**	**0.0782****	**0.0551****
Improv.	1.06%	6.11%	6.17%

Knowledge Graph Rec.

- **Effectiveness** Optimal α, β, and γ depend on the dataset.



MovieLens Amazon

- **Speed** **LightGCN++** is ***marginally*** slower than LightGCN.

Dataset	LightGCN	LightGCN++	Increase
LastFM	0.9137	0.9345	2.27 %
MovieLens	10.0144	10.0232	0.08 %
Gowalla	13.2507	13.3094	0.44 %
Yelp	21.2809	22.0242	3.49 %
Amazon	46.2871	48.7372	5.29 %