COMP 4332 / RMBI 4310
Big Data Mining and Management/Advanced Data Mining for Risk Management and Business Intelligence

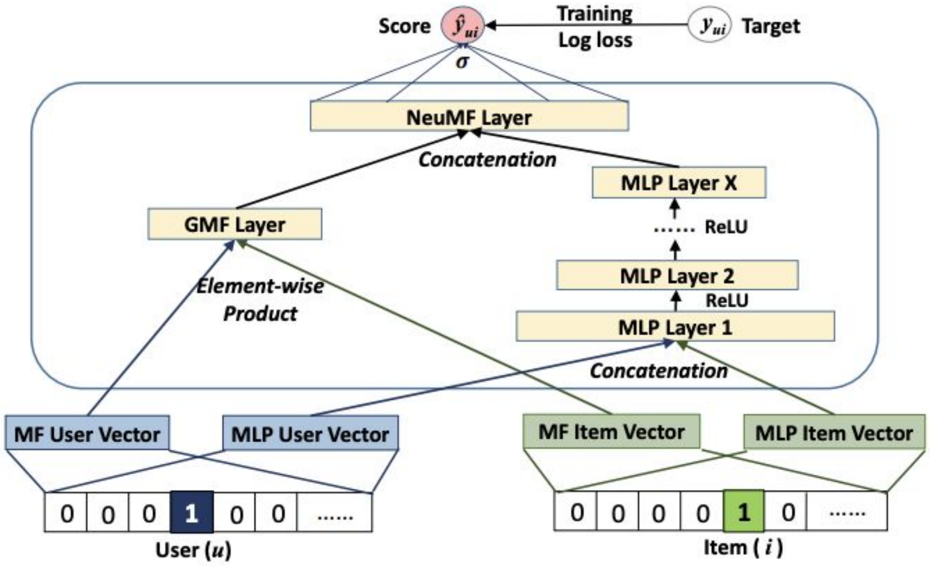
Tutorial 7. LightGCN for RecSys

TA: ZHENG Tianshi (tzhengad@connect.ust.hk)

1. Neural Collaborative Filtering

Advantage: Better expressiveness than traditional matrix factorization due to non-linearity of MLPs.

Limitation: Cannot explicitly capture higher-order graph structures. (only intakes 1 user + 1 item at one time)



https://arxiv.org/pdf/1708.05031

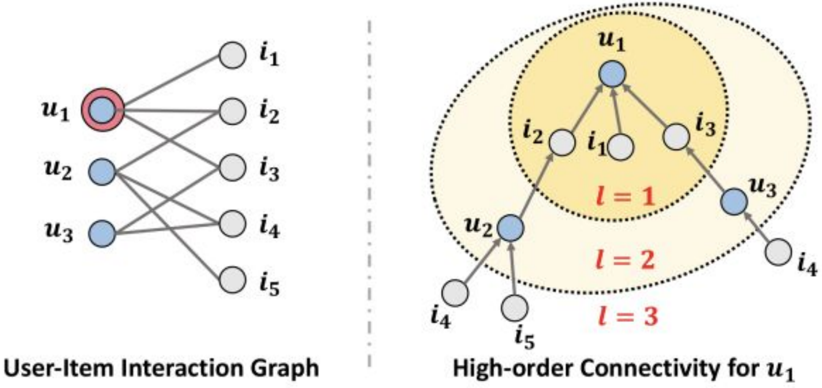
2. Neural Graph Collaborative Filtering (NGCF)

Use **GNN** to process a user-item interaction graph. Both user/item embeddings and GNN parameters are **learnable**.

Score = dot product of user/item embeddings.

Objective: minimize the gap between predicted score and real score.

BPR: maximizing (score_like - score_dislike)

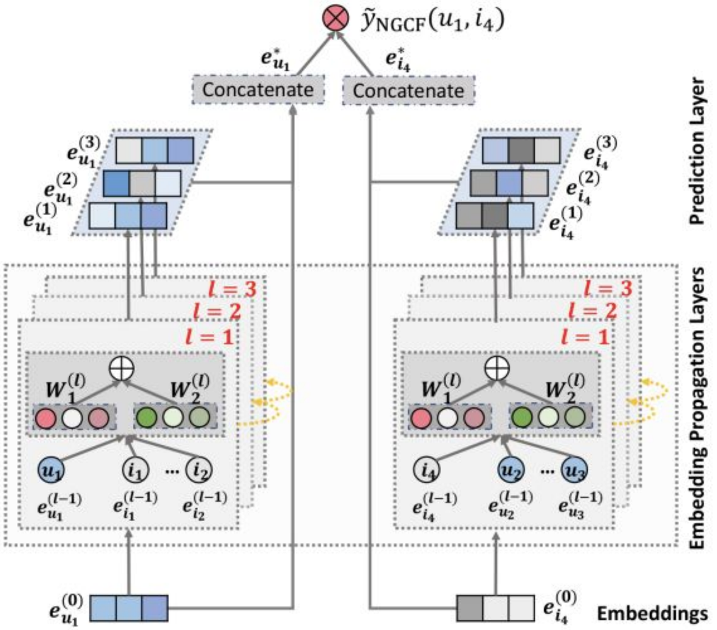


https://arxiv.org/pdf/1905.08108

2. Neural Graph Collaborative Filtering (NGCF) (Cont'd)

Advantage: High-order graph structure is captured through iterative neighbor aggregation.

Limitations: User/Item embeddings are **already quite expressive**. The learnable GNN parameters make the architecture **complex**, take more **computational resource** and prone to **overfitting**.



3. From NGCF to LightGCN

1. Keeping the neighbour aggregation to preserve higher-order structure awareness.
2. Remove:
   a. Feature transformation layers W1/W2 (GNN params)
   b. Non-linear activations (ReLUs)
   c. Self-connections between layers

NGCF

$$\mathbf{e}_u^{(k+1)} = \sigma\big(\mathbf{W}_1\mathbf{e}_u^{(k)} + \textstyle\sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}(\mathbf{W}_1\mathbf{e}_i^{(k)} + \mathbf{W}_2(\mathbf{e}_i^{(k)} \cdot \mathbf{e}_u^{(k)}))\big)$$
$$\mathbf{e}_i^{(k+1)} = \sigma\big(\mathbf{W}_1\mathbf{e}_i^{(k)} + \textstyle\sum_{u\in\mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}}(\mathbf{W}_1\mathbf{e}_u^{(k)} + \mathbf{W}_2(\mathbf{e}_u^{(k)} \cdot \mathbf{e}_i^{(k)}))\big)$$

LightGCN

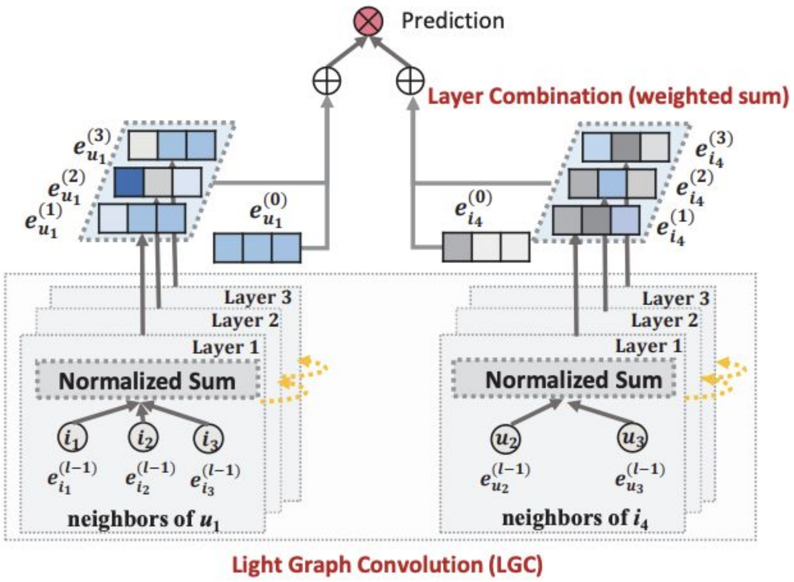
$$\mathbf{e}_u^{(k+1)} = \textstyle\sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}\mathbf{e}_i^{(k)}$$
$$\mathbf{e}_i^{(k+1)} = \textstyle\sum_{u\in\mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}}\mathbf{e}_u^{(k)}$$

4. LightGCN - Architecture

The output of a LightGCN layer is:

Just the **normalized sum** of the neighbors' embeddings from the previous layer.

(Keeping the core neighborhood aggregation mechanism from GCN)



https://arxiv.org/pdf/2002.02126

5. LightGCN - Performance

Table 3: Performance comparison between NGCF and LightGCN at different layers.

Dataset		Gowalla		Yelp2018		Amazon-Book	
Layer #	**Method**	**recall**	**ndcg**	**recall**	**ndcg**	**recall**	**ndcg**
1 Layer	NGCF	0.1556	0.1315	0.0543	0.0442	0.0313	0.0241
	LightGCN	0.1755(+12.79%)	0.1492(+13.46%)	0.0631(+16.20%)	0.0515(+16.51%)	0.0384(+22.68%)	0.0298(+23.65%)
2 Layers	NGCF	0.1547	0.1307	0.0566	0.0465	0.0330	0.0254
	LightGCN	0.1777(+14.84%)	0.1524(+16.60%)	0.0622(+9.89%)	0.0504(+8.38%)	0.0411(+24.54%)	0.0315(+24.02%)
3 Layers	NGCF	0.1569	0.1327	0.0579	0.0477	0.0337	0.0261
	LightGCN	0.1823(+16.19%)	0.1555(+17.18%)	0.0639(+10.38%)	0.0525(+10.06%)	0.0410(+21.66%)	0.0318(+21.84%)
4 Layers	NGCF	0.1570	0.1327	0.0566	0.0461	0.0344	0.0263
	LightGCN	0.1830(+16.56%)	0.1550(+16.80%)	0.0649(+14.58%)	0.0530(+15.02%)	0.0406(+17.92%)	0.0313(+18.92%)

*The scores of NGCF on Gowalla and Amazon-Book are directly copied from Table 3 of the NGCF paper (https://arxiv.org/abs/1905.08108)

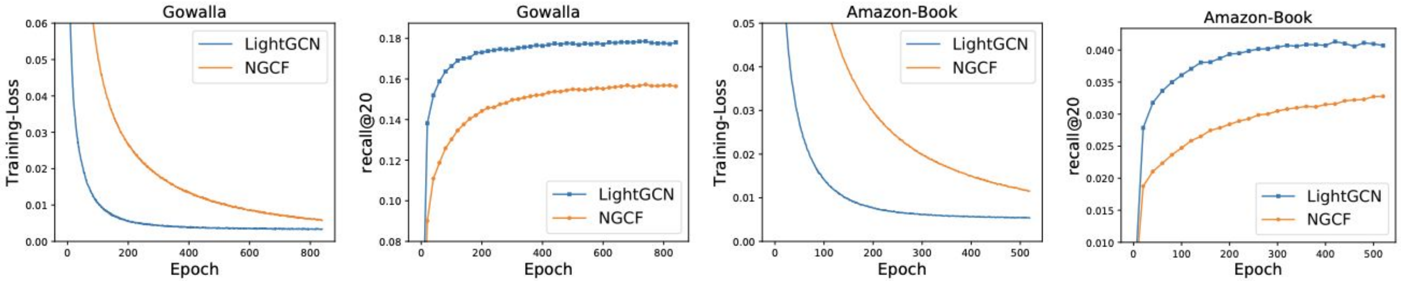


Figure 3: Training curves of LightGCN and NGCF, which are evaluated by training loss and testing recall per 20 epochs on Gowalla and Amazon-Book (results on Yelp2018 show exactly the same trend which are omitted for space).

6. Pytorch Implementation of LightGCN on Amazon-Book

See the python notebook file.