LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation

Xiangnan He
University of Science and Technology of China
xiangnanhe@gmail.com

Kuan Deng
University of Science and Technology of China
dengkuan@mail.ustc.edu.cn

Xiang Wang
National University of Singapore
xiangwang@u.nus.edu

Yan Li
Beijing Kuaishou Technology Co., Ltd.
liyan@kuaishou.com

Yongdong Zhang
University of Science and Technology of China
zhyd73@ustc.edu.cn

Meng Wang*
Hefei University of Technology
eric.mengwang@gmail.com

Shallow embedding

背景

- 问题引入
- 推荐系统的常规方法——矩阵分解
- 扩展用户-项目交互图的高阶连通性——NGCF

问题引入

互联网正处于一个信息爆炸的时代:

10K+ Movies

12M+ Products

70M+ Music

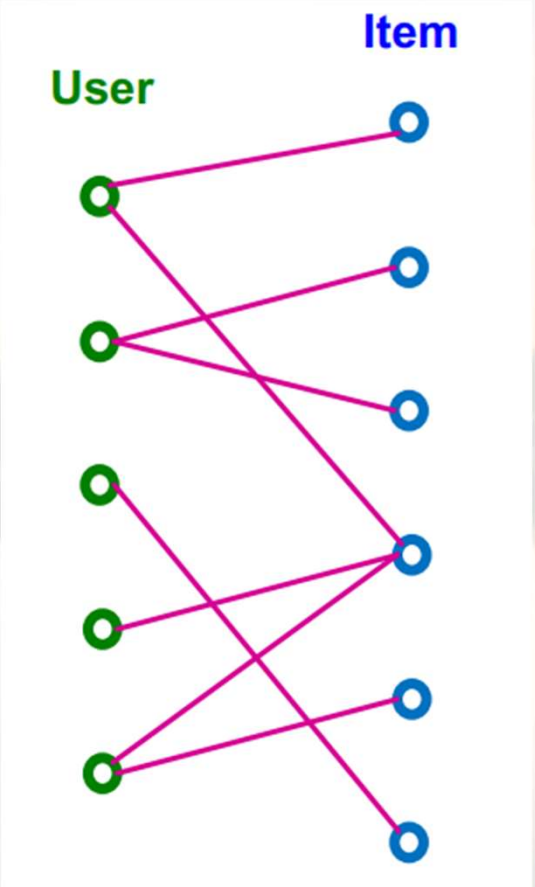
10B+ Vedios

为互联网用户推荐他们感兴趣的内容即个性化推荐是至关重要的!

问题引入

User-Item交互图可以被看作一个二部图:

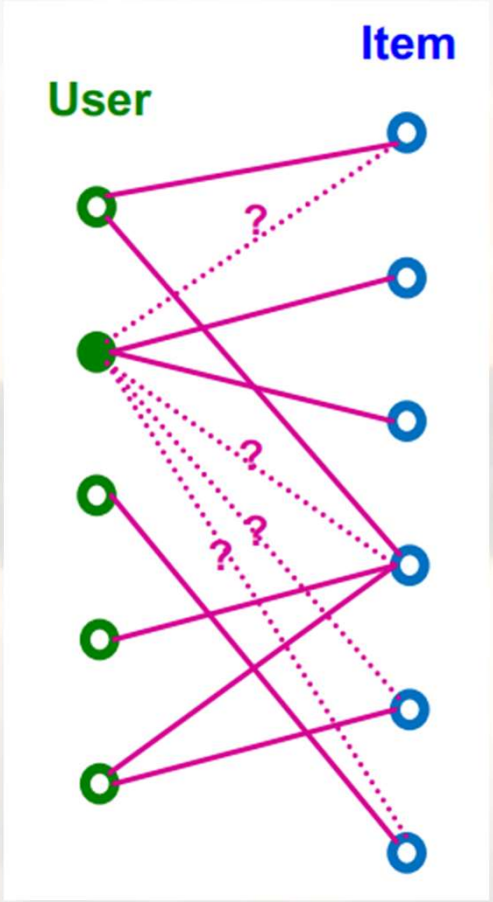
- 图中的结点分为两类：User、Item;
- 图中的边连接着User和Item，这用来表示User和Item之间的交互（例如点击、购买、回复评论等行为）



问题引入

推荐系统的任务：

- **根据**一个历史的User-Item交互图，**预测**未来User会与哪些Item产生交互
- 可以将其视为连接预测问题。即给定过去的边，预测新的用户-物品交互边。

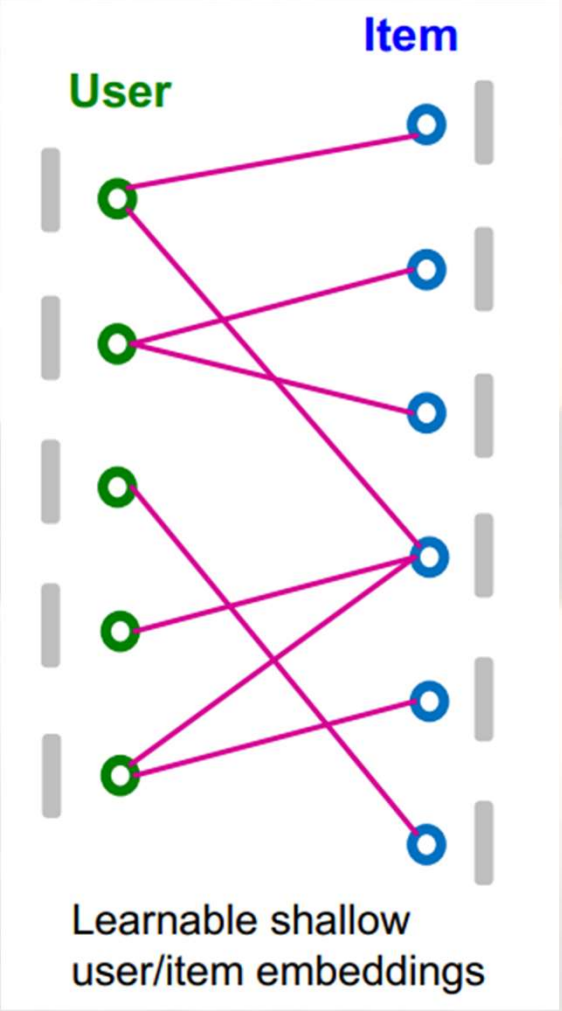


常规方法——Matrix Factorization (MF)

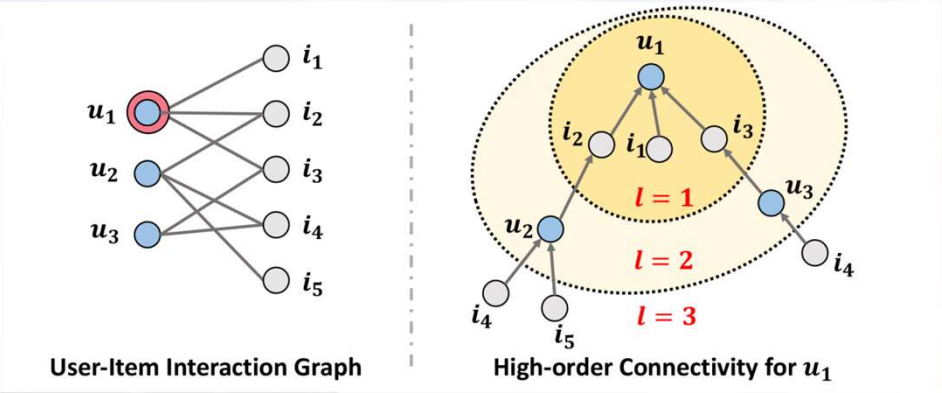
- **把原来的大矩阵，近似分解成两个小矩阵的乘积**
- 使用浅层编码器对用户和物品进行编码。
  - 对于每个用户$u \in U$和物品$v \in V$，可令浅层可学习的嵌入向量u、$v \in \mathbb{R}^D$。
  - 用户u和物品v的评分函数为$f_\theta(u,v) = u^\tau v$

存在问题：

- 该方法并没有很好的利用图结构，只有一阶图结构（边）在训练中被捕捉到，高阶图结构的连通性并没有被显式地捕捉。



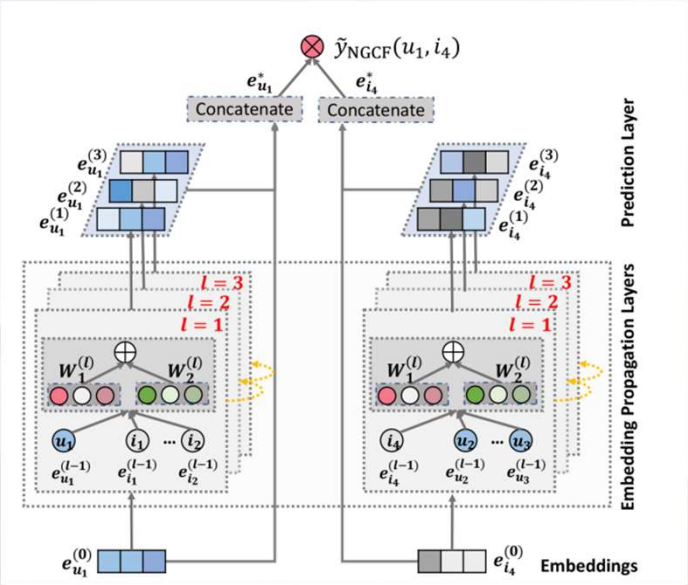
高阶连通性



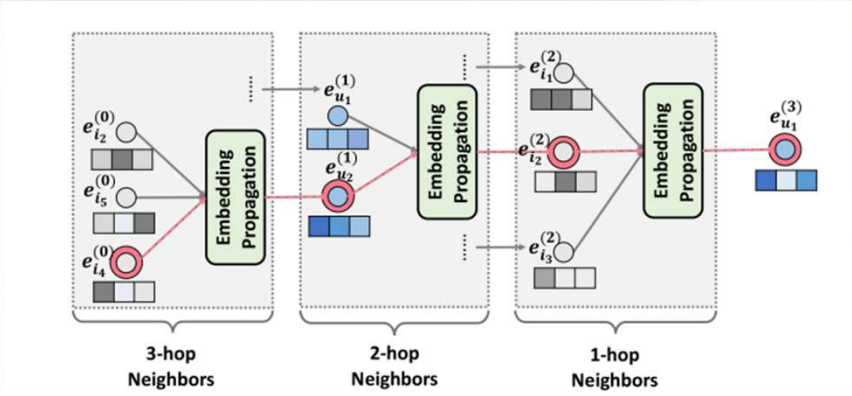
以上图为例说明**高阶连通性**的概念。在用户-物品交互图中，被标记为双圆圈的u1是推荐的目标用户。右图显示了从u1扩展出的树状结构。高阶连接性指的是从路径长度大于1的任何节点到达u1的路径。这种高阶连接性包含了协同信号的丰富语义。例如，路径u1 ← i2 ← u2表示u1和u2之间的行为相似性，因为两个用户都与i2交互过；更长的路径u1 ← i2 ← u2 ← i4表明u1很可能会采纳i4，因为她的相似用户u2在之前与i4有过交互。此外，从路径长度为3的整体视角来看，物品i4对于u1的兴趣可能比物品i5更大，因为连接<i4,u1>的路径有两条，而连接<i5,u1>的路径只有一条。

NGCF——基于GCN

单层的GCN处理图中一阶邻居上的信息，K层GCN处理K阶邻居



NGCF模型体系结构的图示（箭头线表示信息流）。该模型使用多个嵌入传播层来生成useru1（左）和itemi4（右）的表示，最终将其输出连接起来以进行最终预测。



用户u1的三阶嵌入传播示意图

本文主要工作

- 通过**针对NGCF的消融实验**发现常规GCN中的特征变换和非线性激活过程并不适用于协同过滤;
- 受此启发作者在标准的GCN的基础上针对推荐任务进行了简化，提出了轻量级图卷积网络（**LightGCN**）。
- 实验

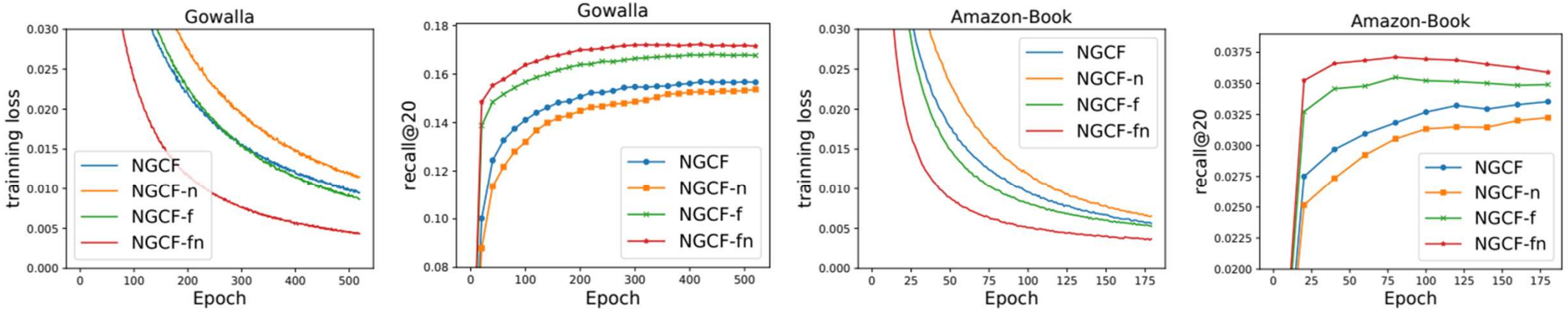
NGCF消融实验

Conclusion：

1） 去除非线性激活函数对于的NGCF性能增益是负的；
2）去除线性特征转换对于NGCF的性能增益是正的；
3）同时去除非线性激活函数和线性特征转换能够带来最大的性能增益。

基于此，作者认为：GCN最初应用于半监督的节点分类任务，该类任务中的节点具有丰富的语义特征信息，然而在协同过滤任务中，user-item交互二部图中的用户和物品节点只有ID编码信息（嵌入）。在这种情况下，执行非线性转换不仅不能提供更好的训练效果，而且会严重阻碍推荐模型想训练。

NGCF的次优表现源于训练的困难，而不是过拟合。



(a) Training loss on Gowalla (b) Testing recall on Gowalla (c) Training loss on Amazon-Book (d) Testing recall on Amazon-Book

Figure 1: Training curves (training loss and testing recall) of NGCF and its three simplified variants.

LightGCN

相比NGCF, LightGCN采用了**简单的加权和聚合器LGC**，移除了特征变换和非线性激活。

LightGCN采用了**层结合**避免了嵌入过度平滑的问题，并能结合不同层的语义信息

LightGCN的模型预测定义为**用户和项目嵌入表示的内积**

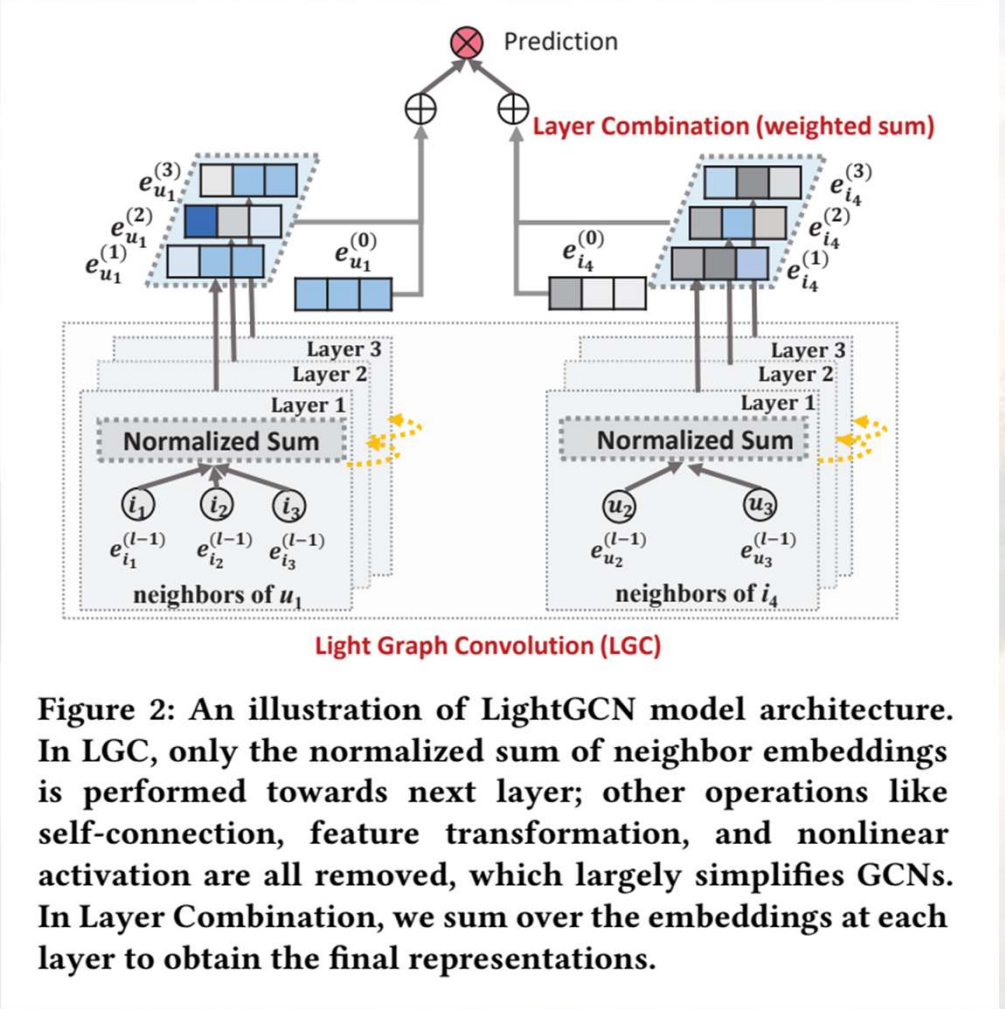


Figure 2: An illustration of LightGCN model architecture. In LGC, only the normalized sum of neighbor embeddings is performed towards next layer; other operations like self-connection, feature transformation, and nonlinear activation are all removed, which largely simplifies GCNs. In Layer Combination, we sum over the embeddings at each layer to obtain the final representations.

传播层 Light Graph Convolution（LGC）

$\mathcal{N}_u$： 和user有交互的item集合
$\mathcal{N}_i$： 和item有交互的user集合

相比NGCF, LightGCN
- 去除非线性激活函数
- 去除线性特征转换
- 与绝大多数图卷积网络设计不同， LGC只包含邻居节点向中心节点的信息传递与聚合过程，而不包括节点的自连接。

NGCF

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

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

Layer Combination

作者并没有使用最后一层嵌入作为最终的用户和物品嵌入，而是选择带权聚合的方式构造最终的用户和物品嵌入：

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)}; \quad \mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{e}_i^{(k)}$$

其中a_k用于调解每一层嵌入对于最终嵌入表示的重要程度。为了减少模型参数，作者将其简单地设为$\frac{1}{K+1}$，即所有层的权重均相同，作者提到，通常来说选择$\frac{1}{K+1}$模型会有更好的表现，同时也能避免复杂的计算。

选择带权聚合的方式构造最终的用户和物品嵌入的原因：

（1）避免过度平滑（over-smoothed）；

（2）不同层的嵌入能捕获不同的语义，所以将它们结合在一起会使嵌入表达更加全面；

（3）将不同层的嵌入与加权和结合起来，具有自连接的图卷积的效果。

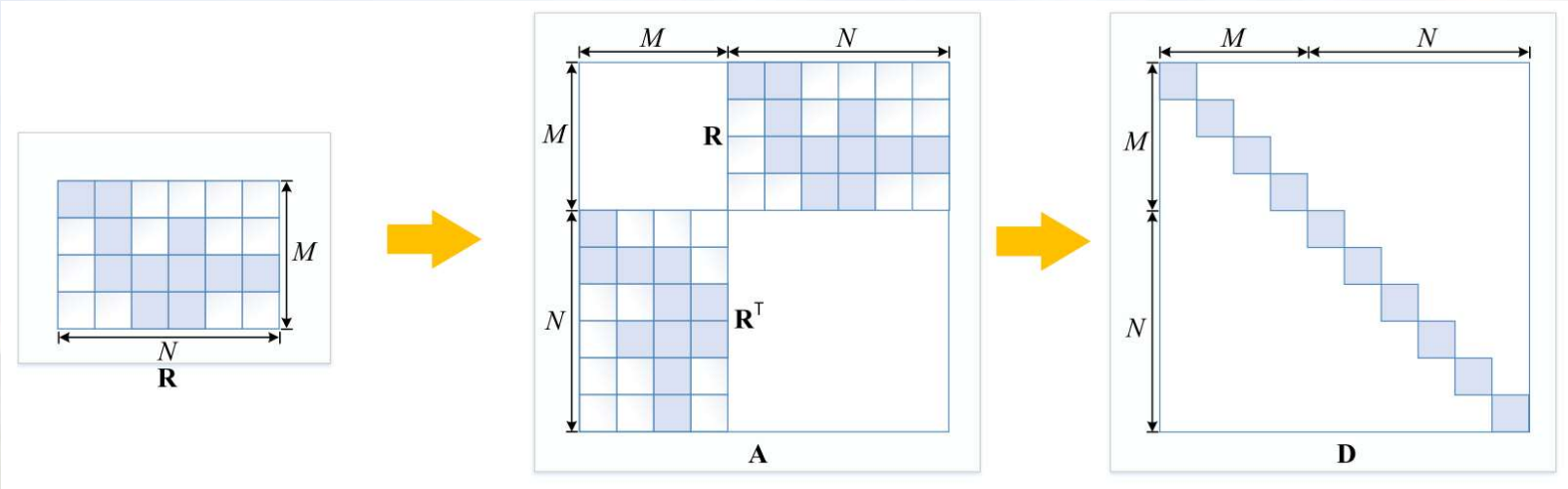
$$\begin{aligned}\mathbf{E}^{(K)} &= (\mathbf{A}+\mathbf{I})\mathbf{E}^{(K-1)} = (\mathbf{A}+\mathbf{I})^K\mathbf{E}^{(0)} \\ &= \binom{K}{0}\mathbf{E}^{(0)} + \binom{K}{1}\mathbf{A}\mathbf{E}^{(0)} + \binom{K}{2}\mathbf{A}^2\mathbf{E}^{(0)} + \ldots + \binom{K}{K}\mathbf{A}^K\mathbf{E}^{(0)}\end{aligned}$$

将自连接插入到A中并在其上传播嵌入，本质上等价于在每个LGC层传播的嵌入的加权和。

Model Prediction

$$\hat{y}_{ui} = \mathbf{e}_u^T \mathbf{e}_i$$

LightGCN——Matrix Form



$$\mathbf{E} = [\underbrace{\mathbf{e}_{u_1}, \cdots, \mathbf{e}_{u_N}}_{\text{users embeddings}}, \underbrace{\mathbf{e}_{i_1}, \cdots, \mathbf{e}_{i_M}}_{\text{item embeddings}}]$$

$$\mathbf{E}^{(0)} \in \mathbb{R}^{(M+N)\times T}$$

$$\mathbf{E}^{(k+1)} = (\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{E}^{(k)}$$

LGC

$$\begin{aligned}\mathbf{E} &= \alpha_0\mathbf{E}^{(0)} + \alpha_1\mathbf{E}^{(1)} + \alpha_2\mathbf{E}^{(2)} + \ldots + \alpha_K\mathbf{E}^{(K)} \\ &= \alpha_0\mathbf{E}^{(0)} + \alpha_1\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \alpha_2\tilde{\mathbf{A}}^2\mathbf{E}^{(0)} + \ldots + \alpha_K\tilde{\mathbf{A}}^K\mathbf{E}^{(0)}\end{aligned}$$

Layer Combination

Model Training

$$L_{BPR} = -\sum_{u=1}^{M} \sum_{i \in \mathcal{N}_u} \sum_{j \notin \mathcal{N}_u} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda ||\mathbf{E}^{(0)}||^2$$

- 使用BPR作为损失函数可以保证模型将用户更喜欢的物品i排在用户更不喜欢的物品u之前，即让正样本和负样本之间的得分之差尽可能的大
- LightGCN的可训练参数仅为第0层的嵌入
- λ用于控制正则化强度

Experiments

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%)	0.530(+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)

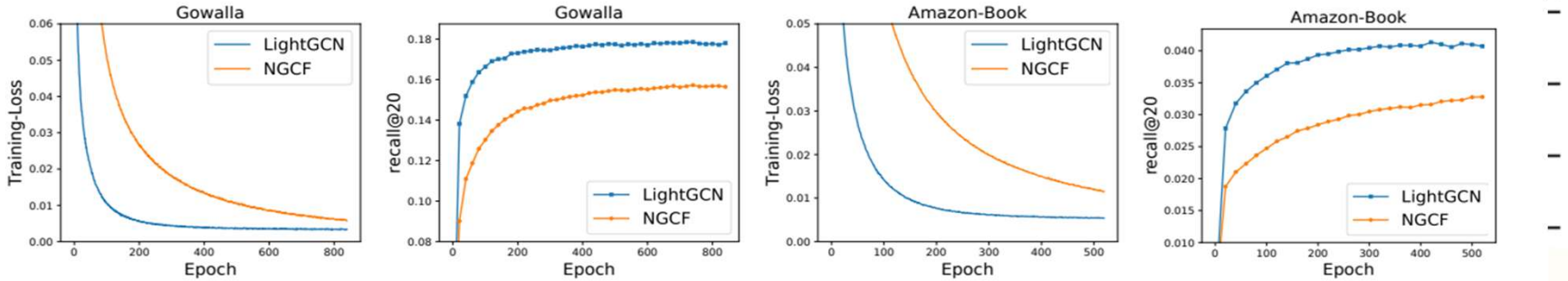


Figure 3: Training curves of LightGCN and NGCF, which are evaluated by training loss and testing recall per 20 epochs on Gowalla and Amazon-Book (results on Yelp2018 show exactly the same trend which are omitted for space).

LightGCN主要与NGCF进行对比，根据实验结果可以看出LightGCN的性能有了明显的提升，这也验证了作者的设想，即特征转换和非线性激活对于推荐任务是冗余的。

Experiments

Table 4: The comparison of overall performance among LightGCN and competing methods.

Dataset	Gowalla		Yelp2018		Amazon-Book	
Method	**recall**	**ndcg**	**recall**	**ndcg**	**recall**	**ndcg**
NGCF	0.1570	0.1327	0.0579	0.0477	0.0344	0.0263
Mult-VAE	0.1641	0.1335	0.0584	0.0450	0.0407	0.0315
GRMF	0.1477	0.1205	0.0571	0.0462	0.0354	0.0270
GRMF-norm	0.1557	0.1261	0.0561	0.0454	0.0352	0.0269
LightGCN	**0.1830**	**0.1554**	**0.0649**	**0.0530**	**0.0411**	**0.0315**

Ablation and Effectiveness Analyses——Layer Combination

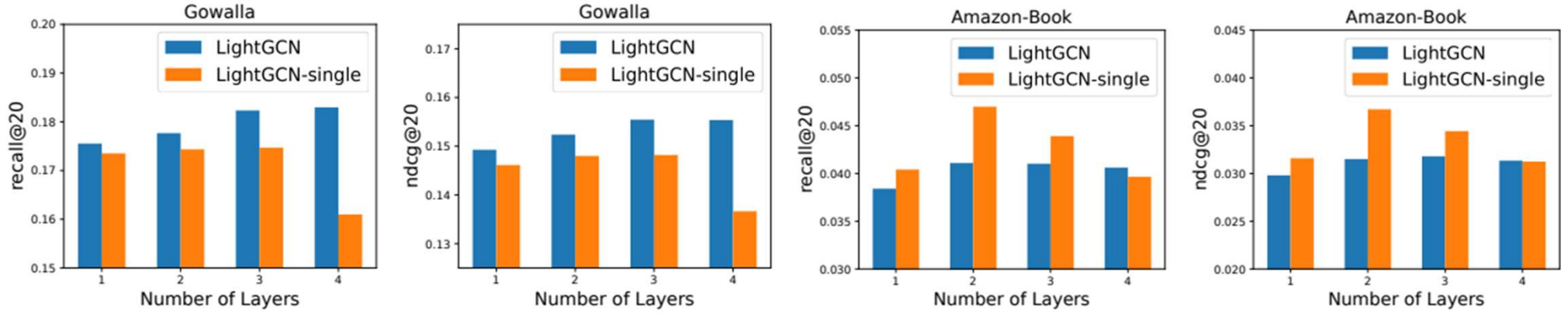


Figure 4: Results of LightGCN and the variant that does not use layer combination (i.e., LightGCN-single) at different layers on Gowalla and Amazon-Book (results on Yelp2018 shows the same trend with Amazon-Book which are omitted for space).

LightGCN-single：该模型只将最后一个图卷积层的嵌入表示作为用户和物品的最终嵌入表示

- 并非所有情况下LightGCN的性能都优于LightGCN-single，这一现象在Amazon-Book和Yelp2018数据集上尤为明显，这表明**带权求和的层结合机制并不是最优方案；**
- 用一阶和二阶邻居对节点的嵌入进行传播对CF非常有用，但当使用高阶邻居时，会出现过度平滑问题；
- LightGCN的性能随着层数的增加而逐渐提高。即使使用4层，LightGCN的性能也不会降低。这证明了层结合用于解决过度平滑是有效的。

Ablation and Effectiveness Analyses——Symmetric Sqrt Normalization

Table 5: Performance of the 3-layer LightGCN with different choices of normalization schemes in graph convolution.

Dataset	Gowalla		Yelp2018		Amazon-Book	
Method	recall	ndcg	recall	ndcg	recall	ndcg
LightGCN-L_1-L	0.1724	0.1414	0.0630	0.0511	**0.0419**	**0.0320**
LightGCN-L_1-R	0.1578	0.1348	0.0587	0.0477	0.0334	0.0259
LightGCN-L_1	0.159	0.1319	0.0573	0.0465	0.0361	0.0275
LightGCN-L	0.1589	0.1317	0.0619	0.0509	0.0383	0.0299
LightGCN-R	0.1420	0.1156	0.0521	0.0401	0.0252	0.0196
LightGCN	**0.1830**	**0.1554**	**0.0649**	**0.0530**	0.0411	0.0315

Method notation: -L means only the left-side norm is used, -R means only the right-side norm is used, and -L_1 means the L_1 norm is used.

- 如果移除归一化操作，训练过程会变得数值不稳定，并可能出现非数值（NAN）的问题，因此作者不建议使用此设置。
- 最佳设置一般是在两侧都使用平方根归一化（即当前的LightGCN设计）。如果移除其中一侧，性能都会大幅下降。
- 第二佳设置是仅在左侧使用L1归一化（即，LightGCN-L1-L）。这相当于通过入度将邻接矩阵归一化为随机矩阵。
- 对两侧进行平方根归一化对LightGCN是有效的，但对两侧进行L1归一化会降低性能。

THANKS!