

## 百度2024：DGR-基于全局与局部视角的通用GCN框架，提升推荐系统的个性化



SmartMindAI

专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我

已关注

10 人赞同了该文章

### Introduction

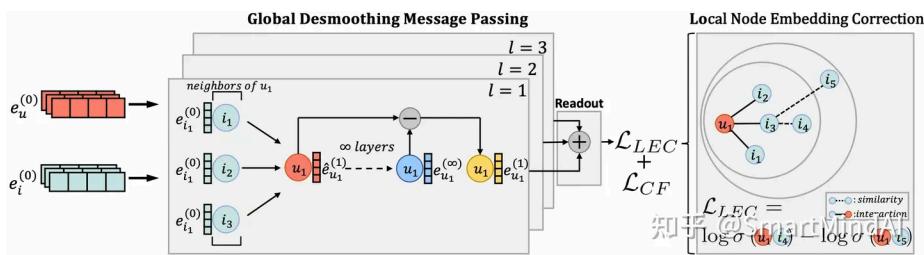
个性化推荐在社交媒体、电子商务和职业推荐等领域非常重要。它通过学习用户和商品的嵌入表示来推荐用户感兴趣的项目。传统的基于GCN的推荐模型使用消息传递机制来聚合局部信息，并将历史交互数据中的信号融入到嵌入向量中，提高了训练效率和性能。然而，这些模型面临过度平滑问题，导致个性化推荐失败，因为所有用户和项目的嵌入值趋向于相似。

此外，推荐系统中的数据往往非常稀疏，这使得模型难以准确捕捉用户和商品之间的细微关系。现有的基于GCN的推荐模型在解决过度平滑问题方面面临挑战。一些模型尝试将每一层的嵌入向量融入到最终的聚合嵌入向量中，以保留节点信息的特性并缓解过度平滑问题。其他方法则调整消息传递过程或采用自我监督学习来保持用户和商品嵌入向量的差异性。然而，这些方法大多是为特定模型设计的，缺乏一个通用的去平滑框架。为了充分利用稀疏的用户-商品交互信息，同时考虑全局和局部拓扑视角是解决这个问题的关键。全局拓扑结构导致现有的GCN模型中的节点嵌入快速收敛，形成过度平滑。因此，修改消息传递层以区分用户和商品嵌入是一个重要的挑战。另一方面，局部协同关系对于推荐性能同样重要，鼓励相似的邻近商品嵌入。在去平滑过程中，提取和重视这些局部协同信号也具有挑战性。

在本文中，我们提出了一种针对基于GCN的推荐系统的统一方法 - DGR (Desmoothing Framework for Graph Convolutional Network Recommendation Systems)。DGR是一个新颖且易于使用的方法，旨在提升基于GCN的推荐系统中的个性化推荐。它包含两个核心模块：GMP (Global Desmoothing Message Passing) 和LEC (Local Node Embedding Correction)。GMP模块通过向每一层引入向量扰动来防止用户和项目嵌入接近过度平滑点的趋势。LEC模块则致力于保持并强化局部协同信号，特别是在数据稀疏的情况下。通过维护用户与其邻近项目的局部协作关系，并融入具有高度相关性的项目，LEC模块加强了局部拓扑信息。与先前专注于个别GCN基础推荐系统改进的方法不同，DGR框架是一种通用的、无模型依赖的解决方案，适用于各种GCN基础模型。我们在五个广泛使用的推荐模型上使用公有基准数据集对DGR进行了验证，证明了其广泛有效性和高适用性。这些实验不仅支持我们框架在提升基于GCN推荐系统个性化推荐方面的效果，还通过多个案例研究进一步证实了其跨模型应用的能力。

### Our Framework

DGR是一种模型无关的方法，通过插入两个组件来增强基于GCN的推荐系统：全局去平滑消息传递 (GMP) 和局部节点嵌入矫正 (LEC)。该方法阻止全局的用户和项目嵌入过度平滑到一个节点，同时保持它们的邻居节点之间的相似性。



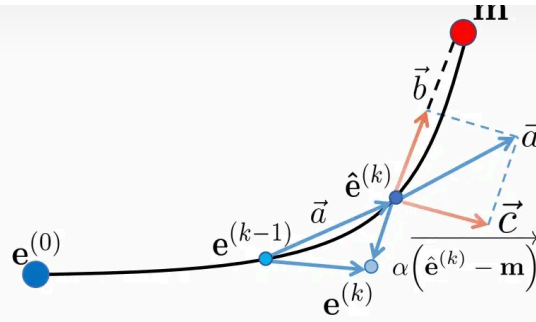


Figure 2: Illustration of the embedding vector updating in Global Desmoothing Message Passing. The solid black line from  $e^{(0)}$  to  $m$  represents the original trajectory of the GCN.

如前所述, 由于全局拓扑结构的数学性质, 每个用户或项目的嵌入  $e^{(k)} \in \mathbb{E}^{(k)}$

为了解决过度平滑问题, 我们提出了一种插件模块, 用于扰动节点嵌入远离过度平滑点。具体来说, 我们在每个消息传递层<sup>+</sup>后插入该模块。图提供了我们方法的直观理解。

节点嵌入在每次消息传递步骤后更新。我们将更新向量表示为  $\vec{a} = \hat{e}^{(k)} - e^{(k-1)}$ 。该向量可以分解为两个正交分量  $\vec{b}$  和  $\vec{c}$ , 其中  $\vec{b}$  与  $e^{(k-1)} - m$  的方向平行,  $\vec{c}$  与之垂直。我们用参数  $\alpha$  来惩罚  $\vec{b}$  方向上的向量更新。

$$e^{(k)} = \hat{e}^{(k)} - \alpha(m - \hat{e}^{(k)}) = (1 + \alpha)\hat{e}^{(k)} - \alpha m,$$

$$\hat{e}^{(k)} = AGG(e_j^{(k-1)}, \{j \in \mathcal{N}\}),$$

其中  $\mathcal{N}$  表示邻近的项目或用户集合, 而  $AGG$  代表定义在公式中的聚合运算。需要注意的是  $\vec{b}$  和  $m - \hat{e}^{(k)}$  具有相同的方位 (不考虑正负号) 但尺度不同。在这里, 我们使用后者的来扰动节点嵌入, 避免在计算过程中使用矩阵正交分解, 从而增强其鲁棒性<sup>+</sup>和实用性。在消息传递过程中去平滑的方法以其简单性和直观性为特点, 在实践中已经展现出了显著的效果。实际上, 我们还可以通过以下定理来验证其有效性。

使用在方程 eq : *gdmpl* 中定义的插件模块, 任何节点嵌入  $e$  与相应的过度平滑点  $m$  之间的距离会增加, 即

$$\|e^{(k)} - m\|_p > \|\hat{e}^{(k)} - m\|_p,$$

其中  $\|\cdot\|_p$  是向量之间的  $p$ -范数距离。

这个不等式的引入是显而易见的。上述不等式表明, 我们的方法通过增加节点嵌入与过度平滑点之间的距离, 防止了用户或项目嵌入向过度平滑点的聚集, 从而保护了嵌入的独立性。

### Local Node Embedding Correction

我们进一步研究了处理过度平滑问题的方法。在局部图中, 一些邻居节点之间的相似性对于合作过滤是有益的。我们提出了一种附加损失, 用于基于GCN的推荐系统的局部用户和项目嵌入学习过程。该方法旨在保持高度相邻且相关的用户和项目节点的聚合嵌入之间的相似性。我们还定义了相似节点和边界节点之间的边界节点。通过对项目节点之间的距离进行排序, 我们可以确定相似节点和边界节点。然后, 我们引入了附加损失项来处理这些节点之间的关系。

其中  $N^+$  表示正对训练数据集  $S(i)$  的相似节点集合  $M(i)$  和边缘节点集合  $\omega_{u, s_{im}}$  和  $\omega_{u, m_{ar}}$  分别表示对用户  $u$  和边缘节点<sup>+</sup>  $m_{ar}$  的归一化系数。在排序项目时, 我们筛选出距离节点  $i$  超过阈值  $\theta$  的项目作为灵感, 仅保留边缘节点。共出现相似度被用作衡量两个项目之间距离的度量, 该度量与用户关联的项目数量成正比。需要注意的是, 共出现相似度与合作过滤的概念一致。因此, 我们可以将损失有效地应用于基于GCN的协同过滤<sup>+</sup>模型中。

通过LEC算法, 最终, 我们为基于GCN的推荐模型推导出以下训练目标:  $\mathcal{L} = \mathcal{L}_{CF} + \lambda \mathcal{L}_{LEC}$

Comparison with Existing Methods

近期，有多个关于GCNs的研究提出了解决过度平滑问题的方法。我们在这里讨论我们的方法与这些方法之间的关系，以提供更全面的理解。

**残差连接<sup>+</sup>**。受到ResNet成功经验的启发，残差连接成为了让GNNs更深更优地发展下去的常识，这在LT-OCF、LightGCN、SGC、DeepGCN和GCNII等作品中都有体现。

以GCNII为例，它将初始节点嵌入整合到每一层的输出中，这一策略可以简洁地归纳为：

$$\mathbf{e}^{(k)} = \hat{\mathbf{e}}^{(k)} + \alpha(\mathbf{e}^{(0)} - \hat{\mathbf{e}}^{(k)}),$$

其中  $\hat{\mathbf{e}}^{(k)}$  和  $\mathbf{e}^{(0)}$  定义方式相同。类似于我们的GMP方法，GCNII也使用另一个向量。

$(\mathbf{e}^{(0)} - \hat{\mathbf{e}}^{(k)})$  去扰动原始节点嵌入  $\hat{\mathbf{e}}^{(k)}$  这个向量的方向旨在远离过度平滑点。

**基于规范化的方法。**

降低过度平滑的方法有规范化节点嵌入和限制嵌入范围。然而，这些方法可能引入噪音并降低性能。DGR通过保持节点独立性和提供方向性来有效防止过度平滑，同时提升性能。

**基于拓扑结构的方法。**

其他方法建议在每层信息传递阶段修改图形的拓扑结构，通过随机删除边或节点来缓解过度平滑问题，例如DropEdge和AdaEdge。然而，这些方法在一定程度上损害了节点周围的局部拓扑结构，这对于建模协作信号非常重要。相比之下，我们的方法不仅惩罚节点嵌入过于相似化的倾向，还保持节点的局部拓扑结构和相邻节点之间的相似性。因此，我们的方法提供了更稳定的性能提升。另一个基于**拓扑结构<sup>+</sup>**的模型IMP-GCN通过对兴趣相似的用户进行分组，并对每个组进行单独的信息传递来缓解过度平滑问题，这与我们提出的LEC的核心理念相似，即鼓励具有类似局部图结构的节点具有相似的嵌入。

Overall Performance

Datasets	Gowalla		MovieLens1M		Douban-Book		Yelp2018		Netflix	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
BPRMF	0.1393	0.1155	0.2486	0.2727	0.1263	0.1032	0.0496	0.0400	0.1034	0.0779
NGCF	0.1602	0.1311	0.2594	0.2676	0.1509	0.1259	0.0593	0.0492	0.1056	0.0814
IMP-GCN	0.1837	0.1510	0.2793	0.3154	0.1599	0.1349	0.0649	0.0529	0.1091	0.0795
LayerGCN	0.1825	0.1502	0.2798	0.3085	0.1619	0.1363	0.0636	0.0515	0.1088	0.0787
LightGCN	0.1832	0.1542	0.2690	0.3014	0.1497	0.1257	0.0650	0.0532	0.0868	0.0615
LightGCN+DGR	0.1880	0.1556	0.2749	0.3065	0.1573	0.1308	0.0669	0.0545	0.0909	0.0652
Improv.	2.61%	0.92%	2.19%	1.69%	5.08%	4.06%	2.92%	2.44%	4.72%	6.02%
SGL	0.1785	0.1503	0.2342	0.2194	0.1511	0.1362	0.0672	0.0553	0.1054	0.0803
SGL+DGR	0.1823	0.1532	0.2414	0.2271	0.1625	0.1453	0.0695	0.0571	0.1105	0.0826
Improv.	2.13%	1.93%	3.07%	3.51%	7.55%	6.68%	3.43%	3.26%	4.84%	2.86%
SimGCL	0.1821	0.1528	0.2560	0.2835	0.1773	0.1562	0.0717	0.0592	0.1213	0.0882
SimGCL+DGR	0.1860	0.1573	0.2752	0.3091	0.1811	0.1601	<b>0.0747</b>	<b>0.0616</b>	0.1271	0.0943
Improv.	2.14%	2.95%	7.50%	9.03%	2.14%	2.50%	4.18%	4.05%	4.78%	6.92%
XSimGCL	0.1681	0.1388	0.2498	0.2669	0.1756	0.1590	0.0684	0.0562	0.1319	0.1014
XSimGCL+DGR	0.1811	0.1533	0.2778	0.3102	0.1828	0.1633	0.0726	0.0599	<b>0.1365</b>	<b>0.1069</b>
Improv.	7.73%	10.49%	11.21%	16.22%	4.10%	2.70%	6.14%	6.58%	3.49%	5.42%
MixGCF	0.1852	0.1579	0.2735	0.3025	0.1771	0.1606	0.0697	0.0571	0.1198	0.0858
MixGCF+DGR	<b>0.1890</b>	<b>0.1604</b>	<b>0.2934</b>	<b>0.3306</b>	<b>0.1922</b>	<b>0.1730</b>	0.0719	0.0592	0.1228	0.0900
Improv.	2.05%	1.56%	7.23%	9.29%	8.53%	7.72%	3.14%	3.63%	2.00%	1.85%

Table 1: Performance comparison of baseline models before and after integrating DGR.

表格总结了我们提出的DGR前后基线模型的性能改善情况以及每个指标的相对提升百分比。主要发现如下：

- 集成DGR的**基线模型<sup>+</sup>**在所有数据集上都表现出性能提升，并取得了最佳结果，证明了DGR的有效性。它解决了基于GCN的推荐系统中的过度平滑问题，为个性化推荐带来了潜力。与传统和最新状态的竞争模型相比，大多数包含DGR的流行模型都有显著的性能提升。
- 在所有数据集上，包含DGR的XSimGCL和SimGCL都获得了巨大的性能改进。例如，在MovieLens1M数据集上，XSimGCL与DGR结合的**召回率<sup>+</sup>@20**增长了11.21%，NDCG@20增长了16.22%。相比之下，XSimGCL和SimGCL采用了向节点嵌入中添加噪声的方法，而DGR则更注重保持全局拓扑引导下的节点独特性，并提升局部用户项目交互的拓扑信息。

稀疏数据集上，如Yelp2018，DGR也能保持稳定的性能提升。

效率方面，我们在Douban-Book数据集上训练了LightGCN、LightGCN+GMP、LightGCN+LEC和LightGCN+DGR模型，结果显示我们的模型在训练过程中相较于基线模型表现更高效。

Further Analysis of DGR

**DGR消融实验<sup>†</sup>**。我们在Douban-Book数据集上对不同模型应用GMP、LEC和DGR进行消融实验。结果显示，GMP和LEC都能改善模型性能。然而，DGR提供的性能改进远超过GMP和LEC的总和，表明这两个组件的协同结合更适合推荐任务中的去平滑。此外，我们观察到在LightGCN、SGL和SimGCL模型上，GMP的性能改进比LEC更显著。但在XSimGCL和MixGCF模型上，这种趋势相反。我们认为这是因为前三个模型已经有效地捕获了协同信息，成为全局去平滑中的性能瓶颈，而后两个模型可能缺乏从用户项目交互中捕获协同信号的能力。

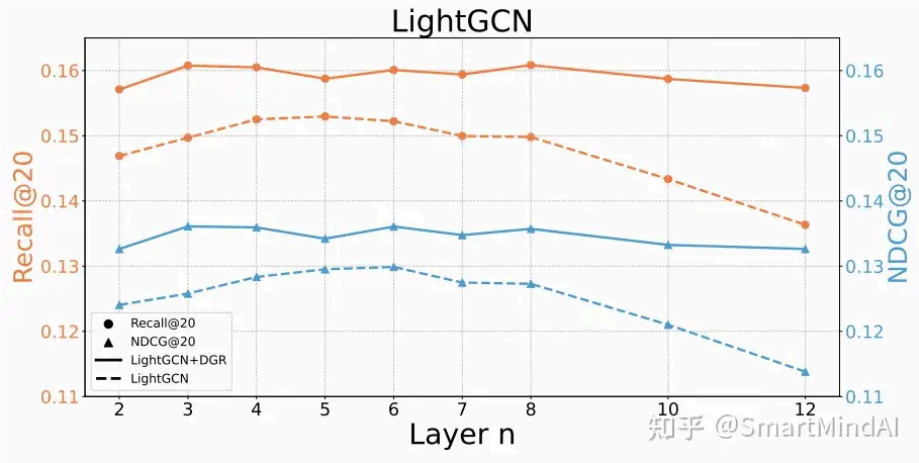
Model	LightGCN		SGL		SimGCL		XSimGCL		MixGCF	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
Baseline Model	0.1497	0.1257	0.1511	0.1362	0.1773	0.1562	0.1756	0.1590	0.1771	0.1606
Model+GMP	0.1533	0.1306	0.1575	0.1418	0.1786	0.1590	0.1777	0.1599	0.1807	0.1623
Improv.	+2.40%	+3.90%	+4.24%	4.11%	+0.73%	+1.79%	+1.20%	+0.57%	+2.03%	+1.06%
Model+LEC	0.1522	0.1279	0.1511	0.1363	0.1790	0.1575	0.1814	0.1631	0.1914	0.1716
Improv.	+1.67%	+1.75%	+0.00%	+0.07%	+0.96%	+0.83%	+3.30%	+2.58%	+8.07%	+6.85%
Model+DGR	0.1573	0.1308	0.1625	0.1453	0.1811	0.1601	0.1828	0.1633	0.1922	0.1730
Improv.	+5.08%	+4.06%	+7.54%	+6.68%	+2.14%	+2.50%	+4.16%	+2.73%	+5.53%	+7.72%

Table 2: Performance comparison of GMP and LEC components in DGR on Douban-Book.

**案例研究。**行差 (Row-diff) 是衡量过度平滑程度的指标，用于评估节点嵌入矩阵的过度平滑程度。我们通过计算节点嵌入矩阵中每对节点之间的欧氏距离<sup>+</sup>来计算行差。为了验证DGR对过度平滑的缓解能力，我们分别训练基线模型和加入DGR的基线模型，并计算它们的行差。结果表明，应用DGR后，基线模型的过度平滑程度显著降低。

Model	LightGCN	SGL	SimGCL	XSimGCL	MixGCF
Baseline	2.68	2.61	2.65	2.32	2.38
Baseline+DGR	2.80	2.85	3.06	2.75	2.74

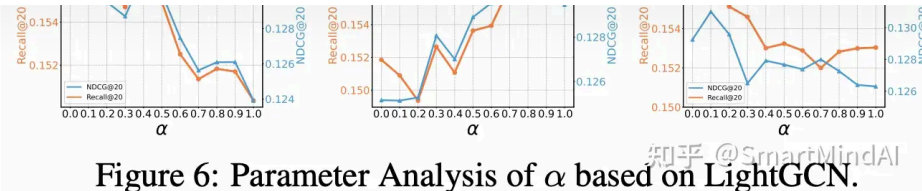
**模型深度的影响。**尽管DGR显著增强了基于浅层GCN的推荐，但还需进一步研究其在深层GCN推荐中的去平滑能力。我们在不同模型深度范围内评估了带有和不带有DGR的五种模型。在Douban-Book数据集上进行实验，观察到带有DGR的模型在性能下降上表现较慢，与不带DGR的模型形成对比。这表明DGR具备强大的去平滑能力，能够在深层模型中有效缓解过度去平滑对推荐性能的负面影响。



**超参数的影响。**超参数 $\alpha$ 对模型性能有重大影响。以轻GCN为例，在Douban-Book数据集上，我们调整了不同的 $\alpha$ 值来融入DGR模型。在每一层中，最优的 $\alpha$ 值有所变化。通过实验，我们发现第一层的最优 $\alpha$ 为0.1，第二层为0.8，第三层为0.1。类似地，其他模型的最优参数也有所不同。例如，SGL模型的最优 $\alpha$ 为1.0, 0.8, 0.8，SimGCL模型为0, 0.6, 0.6，XSimGCL模型为0.2, 0.2, 0.2，MixGCF模型为0, 0.4, 0.2。



知乎



原文《DGR: A General Graph Desmoothing Framework for Recommendation via Global and》

发布于 2024-07-24 10:58 · IP 属地北京

图卷积神经网络 (GCN) 推荐系统 百度

▲ 赞同 10 ▼ ● 添加评论 ↗ 分享 ❤ 喜欢 ★ 收藏 📄 申请转载 ...

理性发言，友善互动

发布

还没有评论，发表第一个评论吧

推荐阅读

**PyG搭建GCN实现链接预测**

Cyril... 发表于GNN

**GCN链路预测项目小结**

枫叶泪

**【干货】GRU神经网络**

zenRR... 发表于深度学习自...

**vn.py发布v2.4.0 - 多进  
算法优化**

用Pyth... 发表于'