论文《Cluster-based Graph Collaborative Filtering》的复现报告

摘要

本文复现了 ClusterGCF 的方法，该方法旨在通过集成图卷积网络（GCN）与聚类技术，改进传统的图卷积推荐算法。ClusterGCF 方法通过将用户和物品嵌入到一个低维向量空间，并采用基于聚类的图结构来捕捉用户和物品之间的潜在关系。具体而言，模型首先通过聚类算法对用户或物品进行分组，以减少计算复杂度和提高推荐的准确性。然后，利用图卷积网络在聚类后的图上进行信息传播，从而实现更为精确的推荐。实验表明，ClusterGCF 方法在多个标准数据集上取得了较好的性能，相较于传统方法具有更优的推荐性能。

关键词：图卷积；聚类；ClusterGCF

1 引言

随着信息技术的迅速发展和在线服务的普及，人们能够快速获取大量信息，从而告别了“信息匮乏”的时代，进入了“信息过载”时代。在这一背景下，推荐系统已成为各种在线平台不可或缺的工具。它不仅帮助用户高效地筛选信息，还为服务提供商提供了一个强大的手段，用以提升客户留存率和增加收入。

在推荐系统领域，基于协同过滤（CF）的方法通过有效利用历史用户-物品交互数据，在学习用户和物品的表示方面取得了显著进展 [1,6]。例如，矩阵分解（MF）方法将用户-物品交互矩阵分解为低维潜在向量，并通过内积来预测用户偏好。近年来，基于图卷积网络（GCN）的模型 [3] 利用交互数据所构建的拓扑结构及其高阶连通性，为推荐系统提供了全新的前沿方法，进一步提高了推荐精度和鲁棒性。

尽管图卷积网络（GCN）在推荐系统中取得了显著的成功，但在相邻节点信息聚合过程中仍面临两个关键挑战。首先，来自不可靠相邻节点的噪声信息可能会损害目标节点的表示 [1,10]。大多数现有的基于 GCN 的推荐模型都构建在用户-物品图上，其中用户和物品通过交互关系相连。这些模型通常假设通过嵌入向量学习能够充分利用高阶邻居的协同信息。因此，它们会将来自相邻节点的信息聚合到目标用户或物品的表示中，而往往没有区分高阶邻居的可靠性。结果，来自不可靠的用户或物品的噪声信息也会参与目标节点表示的学习，导致性能下降。例如，假设用户 w_1 对神秘小说有强烈偏好。然而，一个与 w_1 连接的高阶相邻用户 w_2 更喜欢园艺书籍。在基于 GCN 的模型中，关于园艺书籍的噪声信息可能会被错误地传递到 w_1 的表示中，从而影响推荐的准确性。然而，基于 GCN 的推荐方法还具有过平滑问

题 [2]。大多数基于 GCN 的推荐模型通过堆叠少数几层（通常是 2 或 3 层）来达到最佳性能，但随着层数的增加，性能会显著下降。这是因为图卷积操作会平滑图拉普拉斯算子，导致节点表示在经过多次卷积后变得难以区分 [10]。

2 相关工作

2.1 协同过滤

从推荐系统诞生以来，协同过滤（CF）技术 [2,6] 已广泛应用于推荐系统，并在向用户推荐物品方面发挥着至关重要的作用。其核心思想是通过利用用户与物品之间的交互行为，学习用户和物品的表示。具体而言，矩阵分解（MF）[9] 将每个用户和物品的 ID 映射到嵌入向量空间，并通过内积操作来预测它们之间的交互。为了增强嵌入表示的表达能力，研究者们将物品内容、社交关系、物品关系、用户评论和外部知识图谱等辅助信息引入到模型中 [13]。然而，尽管内积操作能够将观察到的交互中的用户和物品的表示拉近，但它的线性结构不足以揭示用户和物品之间复杂的非线性关系 [6,7]。因此，近年来的研究 [4] 开始关注如何通过深度学习技术来增强交互功能，更好地捕捉用户和物品之间的非线性特征交互。例如，神经协同过滤模型（如 NeuMF）[6] 使用非线性神经网络作为交互函数。

尽管这些方法取得了显著成功，它们仍未能实现推荐嵌入的最佳效果。主要原因在于，协同过滤技术仅隐式地捕捉了用户和物品之间的行为相似性，缺乏对这种相似性的全面建模 [14]。而常见的解决方案是明确利用图结构来辅助用户和物品表示的学习，从而更有效地捕获用户和物品之间复杂的关系。

2.2 基于 GCN 的推荐模型

图卷积网络（GCN）通过在图上执行图卷积操作，迭代地聚合邻居节点的特征，从而更新目标节点的表示，能够有效地捕捉用户和物品之间的高阶关系。随着 GCN 在推荐系统领域的广泛应用，许多方法通过在用户-物品交互图上传播嵌入，充分利用高阶邻接节点的信息。例如，NGCF [14] 就是通过这种方式来捕获高阶邻居的信息。受到简化 GCN 研究启发，学者们批判性地审视了基于 GCN 的推荐模型的复杂性。He 等人 [5] 发现，常用的设计元素如特征变换和非线性激活并未对最终的推荐性能做出显著贡献。因此，他们提出了 LightGCN，该模型省略了这两个组件，简化了模型结构，并显著提高了推荐性能。为了进一步提高推荐效果，UltraGCN [12] 在此基础上进一步简化了图卷积网络，省略了嵌入传播层，并通过结合用户-用户图中的高阶关系，相比 LightGCN，展现了更好的性能和更短的运行时间。

尽管这些方法取得了显著进展，现有的基于 GCN 的推荐方法仍未能充分挖掘高阶邻居中的有价值信息，特别是在没有考虑用户的多重兴趣的情况下。为了解决这一问题，原文提出了一种无监督且可优化的软节点聚类方法，能够对用户和物品进行聚类，从而构建特定于集群的图。通过在这些特定于集群的图上执行高阶图卷积，可以将更有价值的信息传递到目标节点，从而有效提升推荐效果。

3 原文方法

3.1 方法概述

原文提出了基于聚类的图协同过滤（ClusterGCF）模型及其推荐方法。如图 1 所示，ClusterGCF 首先在原始用户-物品交互图和基于聚类的图上进行高阶图卷积操作，然后执行一阶图卷积，最终生成用户和物品的嵌入表示。为了构建基于聚类的图，原文提出了一种软聚类方法，将用户和物品划分为多个聚类。基于聚类结果，原文为每个聚类分配一个概率分数，该分数表示每个节点在原始用户-物品交互图中的重要性。根据这个方法能够生成每个聚类特有的图，并展示该聚类中用户和物品之间的连接关系。

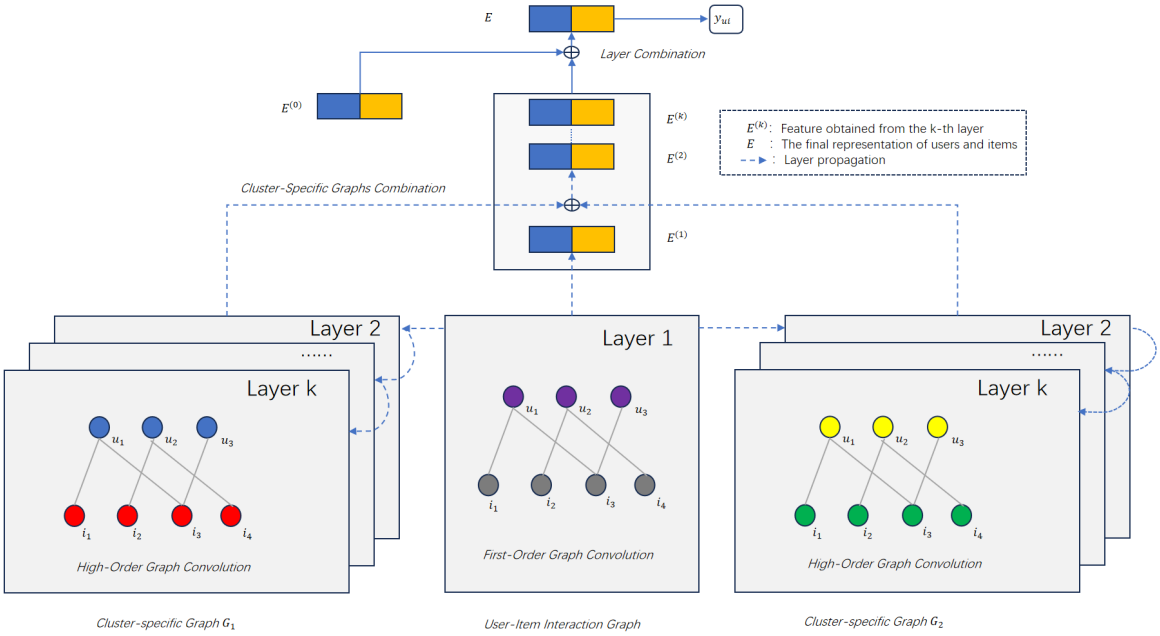

图 1. ClusterGCF 模型示意图

3.2 一阶图卷积

由于用户和物品之间的直接交互通常提供了最重要且最可靠的信息 [10]，因此图卷积网络在处理时会考虑到所有一阶邻居的卷积操作。在 ClusterGCF 模型中，一阶图卷积为：

$$\mathbf{e}_u^{(1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \cdot \sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(0)} \tag{1}$$

$$\mathbf{e}_i^{(1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|} \cdot \sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(0)} \tag{2}$$

对于目标用户 u 和物品 i，$\mathbf{e}_u^{(0)}$ 和 $\mathbf{e}_i^{(0)}$ 分别是它们的 ID 嵌入表示，而它们的第一层嵌入表示为 $\mathbf{e}_u^{(1)}$ 和 $\mathbf{e}_i^{(1)}$。此外，$\frac{1}{\sqrt{|\mathcal{N}_u|} \cdot \sqrt{|\mathcal{N}_i|}}$ 是用于对称归一化的项，用于调整节点的邻接关系权重。

3.3 高阶图卷积

为了减少噪声信息，并且有助于从高阶相邻节点聚合更有价值的信息，原文在多个特定于集群的图上传播嵌入向量。对于集群 $c \in \mathcal{N}_c$，其中 $\mathcal{N}_c$ 是集群集，与集群 c 关联的特定于

集群的图 G_c 上的二阶图卷积定义为:

$$\mathbf{e}_u^{(c,k)} = \sum_{i \in \mathcal{N}_u} \frac{p_i^c}{\sqrt{|\mathcal{N}_u|} \cdot \sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(1)} \tag{3}$$

$$\mathbf{e}_i^{(c,k)} = \sum_{u \in \mathcal{N}_i} \frac{p_u^c}{\sqrt{|\mathcal{N}_i|} \cdot \sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(1)} \tag{4}$$

其中，$\mathbf{e}_u^{(c,k)}$ 和 $\mathbf{e}_i^{(c,k)}$ 表示 k 层图卷积后用户 u 和物品 i 的嵌入，此时 $k=2$。p_u^c 和 p_i^c 表示用户 u 和物品 i 在集群 c 的概率。当 $k>2$ 时，原文将集群特定图 G_c 上的第 k 阶卷积定义为:

$$\mathbf{e}_u^{(c,k)} = \sum_{i \in \mathcal{N}_u} \frac{p_i^c}{\sqrt{|\mathcal{N}_u|} \cdot \sqrt{|\mathcal{N}_i|}} \mathbf{e}_i^{(c,k-1)} \tag{5}$$

$$\mathbf{e}_i^{(c,k)} = \sum_{u \in \mathcal{N}_i} \frac{p_u^c}{\sqrt{|\mathcal{N}_i|} \cdot \sqrt{|\mathcal{N}_u|}} \mathbf{e}_u^{(c,k-1)} \tag{6}$$

接下来，将每个集群特定图组合。通过上述图卷积操作，可以获得集群特定图每一层的节点表示。换句话说，节点信息基于构建的集群特定图从 k 层邻居节点中传递。将节点信息进一步组合为用户 u 和物品 i 第 k 层表示 $(k>2)$:

$$\mathbf{e}_u^{(k)} = \sum_{c \in \mathcal{N}_c} \mathbf{e}_u^{(c,k)} \tag{7}$$

$$\mathbf{e}_i^{(k)} = \sum_{c \in \mathcal{N}_c} \mathbf{e}_i^{(c,k)} \tag{8}$$

$\mathbf{e}_u^{(k)}$ 和 $\mathbf{e}_i^{(k)}$ 分别表示从第 k 层邻居节点聚合的用户 u 和物品 i 的特征。值得注意的是，原文平等地对待不同集群特定图中的信息，因为不同的集群中的节点分配不同的概率，可以捕获不同的重要性。

3.4 层组合与预测

原文采用了与 LightGCN [5] 相同的层组合和预测方法。具体来说，在执行 k 层图卷积后，通过结合所有层的嵌入，得到用户 u 和物品 i 的最终嵌入:

$$\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)} \tag{9}$$

$$\mathbf{e}_i = \sum_{k=0}^{K} \alpha_k \mathbf{e}_i^{(k)} \tag{10}$$

其中 α_k 统一设置为 $1/(K+1)$。在学习了用户和物品的表示后，对给定用户 u 和目标物品 i 的预测评分为:

$$\hat{r}_{ui} = \mathbf{e}_u^T \mathbf{e}_i \tag{11}$$

3.5 软聚类

原文引入了一种软聚类方法来捕获用户的多个兴趣并识别它们之间的共同兴趣。由于用户通常在物品的不同方面表现出多重兴趣，原文将用户和物品节点分类为具有不同关联程度

的多个集群，而不是将它们分配给单个集群。以用户 u 和物品 i 为例，首先融合图结构和 ID 嵌入：

$$F_u = \sigma\left(W_1\left(\mathbf{e}_u^{(0)} + \mathbf{e}_u^{(1)}\right) + b_1\right) \tag{12}$$

$$F_i = \sigma\left(W_1\left(\mathbf{e}_i^{(0)} + \mathbf{e}_i^{(1)}\right) + b_1\right) \tag{13}$$

其中 F_u 和 F_i 是通过特征聚合获得的用户 u 和物品 i 的增强特征。$W_1 \in \mathbb{R}^{d\times d}$，$b_1 \in \mathbb{R}^{1\times d}$ 分别是权重矩阵和偏置向量。采用 LeakyReLU 函数作为激活函数。随后，使用单层神经网络将这些增强的特征转换为预测向量：

$$H_u = \sigma(W_2 F_u + b_2) \tag{14}$$

$$H_i = \sigma(W_2 F_i + b_2) \tag{15}$$

其中 H_u 和 H_i 分别是用户和物品的预测向量。类似地，$W_2 \in \mathbb{R}^{d\times|\mathcal{N}_c|}$，$b_2 \in \mathbb{R}^{1\times|\mathcal{N}_c|}$ 是权重矩阵和偏置向量。值得注意的是，预测向量的维数等于集群特定图的数量，该值由预选的超参数决定。

聚类方法的一个挑战是离散聚类分配的不可微性，这使得基于梯度的优化方法难以应用。为了解决这个问题，原文在提出的方法中采用了 Gumbel-Softmax 技术 [8] 来实现软聚类。由于 Gumbel-Softmax 是可微的，它能够直接融入涉及神经网络的端到端学习框架中。换句话说，聚类方法可以与推荐模型一起进行联合优化。

Gumbel-Softmax 函数提供了对离散分布的平滑可微近似，并由 logits 和温度参数进行参数化。在该的方法中，神经网络的最后一层生成所有集群的 logits。然后，这些 logits 被输入到 Gumbel-Softmax，从而为每个用户或物品生成软聚类分配。例如，给定的 logits 如 H_u，Gumbel-Softmax 函数在多个集群上输出一个概率分布 P_u，数学上定义为：

$$P_u = \text{Softmax}\left(\frac{H_u + g}{\tau}\right) \tag{16}$$

其中 g 是 Gumbel 噪声，τ 是温度。Gumbel 噪声可以通过 Gumbel 分布生成，或者使用变换 - $\log(-\log(U)$ 从均匀噪声 U 生成。

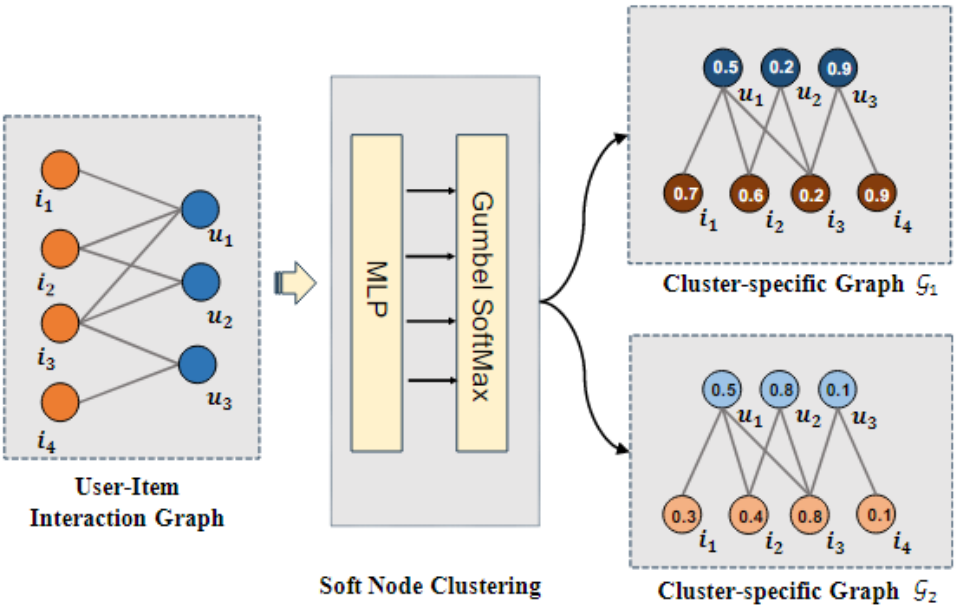

图 2. 软聚类示意图

在使用软聚类技术之后，特别是使用 Gumbel-Softmax 技巧，原文将原始用户-物品交互图重构为一系列集群特定图。与可以使用阈值来确定硬聚类分配的传统方法不同，原文的方法中原始图中的每个节点都根据其软聚类概率分配给所有集群。图 2 为软聚类示意图，来自原文 [11]。

3.6 损失函数定义

原文的目标是通过计算用户的偏好分数，向目标用户推荐物品列表。与 LightGCN 类似，本文也使用贝叶斯个性化排名（BPR）损失，使得模型更倾向于正确物品的得分高于错误物品。为了训练这个模型，构建一个 $\{u, i^+, i^-\}$ 的三元组，其中 i^+ 表示有交互的物品，i^- 表示没有交互的物品，则损失函数为：

$$\arg\min \sum_{(u,i^+,i^-)\in\mathcal{O}} -\ln \phi\left(\hat{r}_{ui^+} - \hat{r}_{ui^-}\right) + \lambda\|\Theta\|_2^2 \tag{17}$$

其中 $\mathcal{O}$ 表示训练集，定义为 $\mathcal{O} = \{(u, i^+, i^-) \mid (u, i^+) \in \mathcal{R}^+, (u, i^-) \in \mathcal{R}^-\}$。这里，$\mathcal{R}^+$ 表示训练数据集中观察到的交互，而 $\mathcal{R}^-$ 表示一组采样的未观察到的交互。此外，λ 表示正则化权重，Θ 代表模型参数。原文采用 L_2 正则化来减轻过拟合。

4 复现细节

4.1 与已有开源代码对比

本次复现的模型是 ClusterGCF 模型，该模型有公开的代码，本次复现主要以公开代码为基础进行的。

表 1. 数据集具体信息

Dataset	Users	Items	Interactions	Density
Gowalla	29,858	40,981	1,027,370	0.00084
Yelp2018	31,831	40,841	1,666,869	0.00128
Amazon - Book	52,643	91,599	2,984,108	0.00062

4.2 实验环境搭建

为了搭建实验环境，本文使用 Python3.9 作为主要的编程语言。其中调用到的 Python 库包括了 pandas,scipy,numpy,tqdm,tensorboardX,scikit-learn 等。

4.3 实验数据集介绍

关于数据集的选择和预处理，本文采用 Gowalla, Yelp2018 和 Amazon-book 三个数据集进行实验。数据集的具体信息如表 1 所示。

Gowalla 数据集是从 Gowalla 获得的一个包含社交媒体签到数据的数据集，用户在 Gowalla 平台上的签到记录和社交网络关系揭示了他们在现实世界中的活动和偏好。研究者可以通过分析用户的地理位置历史、签到地点和社交连接，为用户提供个性化的地理位置推荐。Yelp2018 数据集取自 2018 年版 Yelp 挑战。其中，数据集包含了用户对商家的详细评分和评论，以及地理位置等信息。通过分析用户行为和情感反馈。Amazon-book 数据集是从一个名为 Amazon-review 的广泛使用的产品推荐数据集选取的。

对于每个数据集，为了确保数据集的质量，本文只保留至少有 10 个交互的用户和物品。接着，随机选择每个用户的 80% 的历史交互作为训练集，并将剩余的作为测试集。训练过程中，随机从训练集中选择 10% 的交互作为验证集来调整超参数。对于每个观察到的用户-物品交互，本文将其视为正样本。

5 实验结果分析

实验采用 Recall@K 和 NDCG@K 两个广泛使用的评价指标来评估推荐性能，K 设置为 20。其中，Recall@K 是一个衡量推荐系统在前 K 个推荐物品中是否包含用户感兴趣物品的指标。而 NDCG@K 是一个综合考虑推荐列表中物品的排名和相关性的指标，它反映了推荐结果的排序质量。总的来说，两个评价指标越大，代表着推荐精度越好。

关于 ClusterGCF 模型的参数设置，嵌入维度固定为 64。在优化方面，默认学习率为 0.001。在 Office 数据集上，mini-batch 大小为 1024；相对地，在 Gowalla 和 Yelp208 数据集上，mini-batch 大小增加到 2048，以加速模型训练。对于 L_2 正则化系数 λ，其取值范围为 $\{1e^{-6}, 1e^{-5}, \cdots, 1e^{-1}\}$。嵌入传播层数的取值范围为 $\{1, 2, \cdots, 8\}$。对于聚类数量, 在 $\{2, 3, 4\}$ 范围内进行选取。此外，温度系数的调节范围为 $\{1e^{-2}, 1e^{-1}, \cdots, 1e^{2}\}$。本文采用早停策略，确保对关键参数进行了调优。经过多次实验，本文发现正则化系数 λ 取 $1e^{-4}$，聚类数量取 3，温度系数取 $1e^{-1}$ 时 ClusterGCF 具有最高性能。

将 NDCG，LightGCN 和 ClusterGCF 三个模型在 Gowalla, Yelp2018 和 Amazon-book (office) 三个数据集进行实验，得到实验结果。

表 2. 实验结果

	Gowalla		Yelp2018		Office	
Model	Recall@20	NDCG	Recall@20	NDCG	Recall@20	NDCG
NGCF	0.1570	0.1327	0.0581	0.0475	0.1257	0.0742
LightGCN	<u>0.1809</u>	<u>0.1542</u>	<u>0.0633</u>	<u>0.0512</u>	<u>0.1319</u>	<u>0.0783</u>
ClusterGCF	**0.1907**	**0.1621**	**0.0706**	**0.0583**	**0.1397**	**0.0846**

从表 2 中可以看出 LightGCN 始终优于 NGCF。它通过消除冗余操作并降低计算复杂度来简化 NGCF。这展示了 GCN 的强度，并强调了将高阶信息整合到表示学习中的重要性。ClusterGCF 方法在所有指标始终优于所有数据集的所有基线模型。与 LightGCN 相比，Recall 最多提高了 11.53%，而 NDCG 最多提高了 13.87%，性能提升比较显著。总的来说，ClusterGCF 有效地避免了噪声的影响，并从高阶邻居节点中捕获更有价值的信息，验证了其在推荐性能方面上的有效性。

6 总结与展望

本文复现了 ClusterGCF 模型，并探讨了其在推荐系统中的应用。通过基于用户-物品交互数据构建特定于集群的图，ClusterGCF 有效地结合了软聚类方法和高阶图卷积操作，从而实现了用户与物品间复杂关系的建模。复现过程中，本文验证了模型在提升推荐性能方面的有效性。然而，在实现过程中也发现了一些不足之处，例如，模型的计算复杂度较高，尤其是在处理大规模用户和物品数据时，特定于集群的图构建和训练时间消耗较大。此外，软聚类方法对超参数的选择较为敏感，这对模型的稳定性提出了挑战。

在未来的研究中可以探索更高效的聚类算法或图简化方法，以降低计算复杂度。其次，还可以进一步研究 ClusterGCF 在跨域推荐等场景中的表现。通过这些优化，相信 ClusterGCF 能在推荐系统中发挥更大的潜力。

参考文献

[1] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[2] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 27–34, 2020.

[3] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. Graph trend filtering networks for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 112–121, 2022.

[4] Xiangnan He and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 355–364, 2017.

[5] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[7] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, pages 193–201, 2017.

[8] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[10] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. Interest-aware message-passing gcn for recommendation. In *Proceedings of the web conference 2021*, pages 1296–1305, 2021.

[11] Fan Liu, Shuai Zhao, Zhiyong Cheng, Liqiang Nie, and Mohan Kankanhalli. Cluster-based graph collaborative filtering. *ACM Transactions on Information Systems*, 42(6):1–24, 2024.

[12] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. Ultragcn: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1253–1262, 2021.

[13] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[14] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.