

用于推荐的自传播图神经网络

摘要

在推荐系统中，用户偏好通过学习基于已观察的用户-物品交互数据的用户和物品嵌入（即节点表示）来建模，这些数据形成一个二分图。图神经网络（GNNs）通过探索图的拓扑结构来优化这些表示，传播邻居的嵌入到每个节点，从而重建其嵌入。现有的 GNN 存在两个主要缺陷：（1）图的不完整性：稀疏的观察图中存在大量缺失的连接，导致传播效果低效且有偏差；（2）传播权重问题：传播权重是通过粗略的预定义规则确定的，这些规则仅考虑节点的度数，未能充分考虑实际应用中的复杂性。本文作者提出了一种新的自传播图神经网络（SGNN），采用密集且数据驱动的传播机制。该方法从数据中学习传播矩阵，而不是依赖预定义的规则。为了解决推荐任务中图的稀疏性，作者提出了三种方法来补全图并构建传播矩阵：（1）基于推荐模型补全图；（2）基于谱聚类测量节点距离；（3）基于预测嵌入预测图中缺失的连接。在 SGNN 中，嵌入不仅可以传播到已观察到的邻居，还可以传播到潜在的、未观察到的邻居。传播权重是根据连接强度学习得到的，而不是基于预定义规则。通过在三个真实世界数据集上的综合实验，证明了 SGNN 在有效性和效率方面显著优于最近的、最先进的 GNN 模型。

关键词：图神经网络；自传播；协同过滤；物品推荐

1 引言

图神经网络（GNNs）由于其能够通过探索图的拓扑信息学习丰富的节点表示，在推荐系统中引起了广泛关注。GNNs 代表了一种新的神经网络范式，通过图中节点嵌入（或特征）的传播来优化每个节点的嵌入 [1]。在 GNN 中，嵌入通过单层传播被传播到一跳图邻居，并通过堆叠 L 层传播到 L 跳邻居。本文中，我们不区分 GNN 和具有一阶多项式滤波器的图卷积网络（GCNs），因为 GNN 中的传播与 GCN 中的图卷积是等价的 [2]。在推荐系统的背景下，我们通常使用一个二分的用户-物品交互图，其中节点表示用户和物品。GNN 通过图连接来平滑节点嵌入，并提供了一种显式的方式来利用协同信息。此外，GNN 为探索推荐任务中的深层结构提供了一种新方法——传统的神经网络侧重于学习用户和物品嵌入的非线性组合，而 GNN 则专注于学习嵌入，这是推荐模型中最关键的部分 [3]。因此，GNN 在建模用户偏好方面取得了显著的改进，并被广泛应用于此领域 [4]。

然而，GNN 也面临两个关键问题。第一个问题是：（1）用于传播的图极度不完整且有偏。在现实应用中，只有一小部分连接是可观察到的，这严重影响了传播的质量。。具体而言，缺失连接的问题导致了两个关键缺陷：（i）传播高度低效：图 1 中移除一个连接会导致图 2 中大部分传播树的移除。在推荐系统的背景下，图中大量的连接由于曝光不足而被遗漏，因此传

播树的不完整性问题非常严重，导致嵌入无法有效传播。(ii) 更糟糕的是，传播严重偏向，并且无法通过多跳传播来解决。深度 GNN 旨在通过探索多跳连接来缓解缺失连接的问题——尽管 u_1 与 i_2 之间的连接丢失，但物品 i_2 通过三跳链接与用户 u_1 建立了连接（见图 2）。然而，这种缓解远远不够——嵌入在一次传播后迅速衰减（衰减因子为 $\frac{1}{\sqrt{|N_i||N_j|}}$ ，而物品 i_2 在通过三跳传播到用户 u_1 后，其嵌入变得极为薄弱。尽管 i_1 和 i_2 对用户 u_1 来说同样重要，但由于缺失连接， i_2 在 GNN 中的贡献远低于 i_1 ，从而导致传播偏向问题。GNN 的另一个问题是：(2) 基于规则的传播权重是经验性的且粗略的： $\frac{1}{\sqrt{|N_i||N_j|}}$ 表示从节点 j 到节点 i 的传播权重。如我们所见，权重由邻居集的大小决定，而在实际应用中，这一问题更加复杂。例如，尽管篮球和马克杯的销售量相似（邻居集大小相似），但篮球对于用户偏好的信息贡献要大于马克杯。因此，对于用户节点，指向篮球的链接应比指向马克杯的链接权重更大。虽然 [4] 提出了一种注意力机制来计算链接的重要性，但这需要额外的计算。在我们的传播策略中，这一问题可以自然地解决，而无需额外的计算。此外，传统的传播策略仅根据一跳连接计算传播权重。而在我们的密集图中，传播权重，即每个链接的强度，是通过考虑多跳连接来预测的。

鉴于上述 GNN 的缺陷，作者提出了一种新的传播图，通过该传播图进行传播，而不是使用观察到的图。由于所有缺失的链接在新图中都被补全，因此我们的传播图是密集且无偏的。同时，由于每个链接的强度是从数据中学习的，因此我们的传播是数据驱动的。我们随后提出了一种自传播图神经网络 (SGNN)，将新型的传播策略集成到深度架构中。考虑到维持一个大规模且密集的图在空间和时间上的消耗是不可接受的，作者将图重构为低秩形式。与传统的 GNN 类似，作者通过传播嵌入来优化节点表示，并将每一层的嵌入结合起来作为预测嵌入。通过数据驱动、全面且无偏的传播，能够为推荐任务学习更好的用户和物品表示。

贡献简要总结如下：为了实现更好的传播，我们预测交互图中的缺失链接，并使用这个完整的图进行传播。作者提出了三种方法来补全图，并通过实验验证了其有效性。作者通过将密集传播集成到通用深度 GNN 结构中，设计了一个自传播图神经网络 (SGNN)。在 SGNN 中，每个节点都通过所有邻居（包括潜在的邻居）进行优化，其贡献是从数据中学习的。我们在三个真实世界数据集上进行了综合实验，证明了我们提出的方法的可行性和效率 [5]。

2 相关工作

近年来，推荐系统因其在各大平台上的广泛应用（如电商网站、视频流媒体服务、社交媒体应用等）及其带来的显著经济效益，受到了广泛关注。在推荐任务中，我们的目标是通过探索用户-物品图中的协同信息来建模用户偏好。为此，提出了几种协同过滤 (CF) 模型 [6]，这些模型侧重于利用用户-物品交互数据预测用户偏好，推动了推荐系统的发展。在各种 CF 方

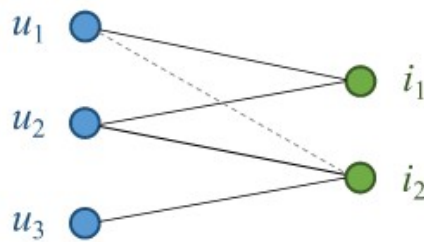


图 1. 二分图

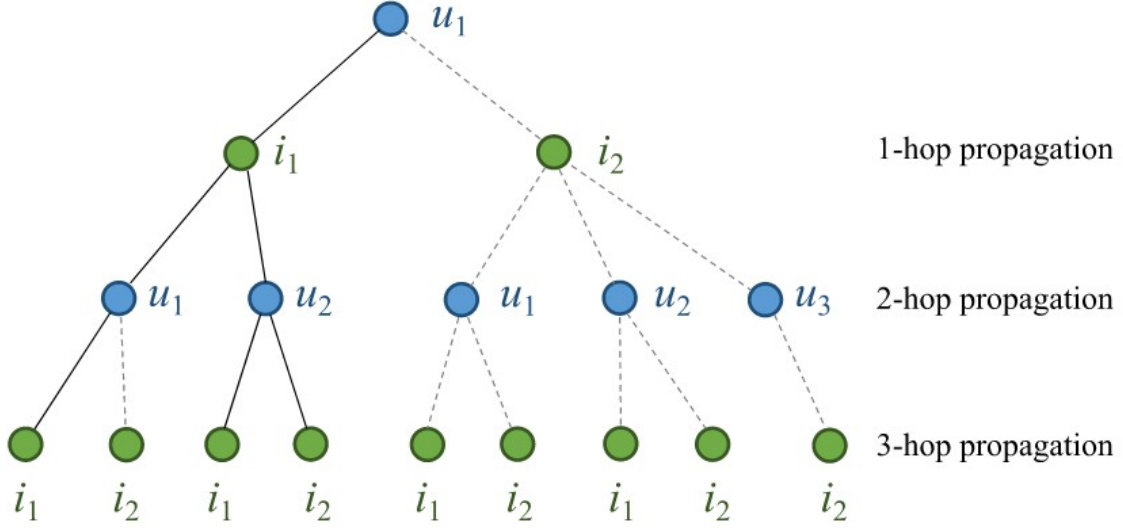


图 2. 传播树 (3 跳传播)

法中，矩阵分解 (MF) [7] 作为一种基础且高效的推荐模型，通过嵌入表示用户偏好，成为推荐系统中的核心方法之一。为了进一步提升性能，许多变种模型相继被提出 [8]。其中，基于神经网络和图的方法与本论文研究最为相关。

2.1 基于神经网络的推荐方法

基于神经网络的推荐方法：深度神经网络在提取高阶特征和做出预测方面表现出了强大的能力，因此在机器学习领域引起了广泛关注。在推荐任务中，研究者们也探索了多种利用神经网络的方法。[3], [9] 设计了一个深度全连接网络，通过非线性方式将用户和物品的嵌入结合起来，而 [10] 则利用卷积网络实现这一目标。[11] 研究了使用注意力机制来增强表示能力。[12], [13] 利用深度卷积神经网络 (CNNs) 提取视觉特征来建模用户的视觉偏好。[1] 通过对抗学习建模用户偏好，其中生成器为每个用户创建物品，判别器则判断给定物品是否为真实的正向物品还是生成的物品。尽管这些深度模型已被广泛研究，但它们关注的是推荐模型的外围方面，如侧面信息、嵌入组合和嵌入加权等。而本文的目标是利用深度架构学习更好的嵌入，因为嵌入是推荐模型中最为核心的部分。

2.2 基于神经隐式场的重构

推荐任务可以看作是一个链接预测任务——在其中我们只能观察到二分图中的一小部分链接，我们的目标是预测缺失的链接。在推荐的背景下，已经采用了多种图数据工具来探索图结构。[14] 利用图正则化来平滑嵌入。通过最小化图正则化，我们可以确保具有相似交互行为的节点在潜在的偏好空间中有相似的表示。[15] 从用户和物品超图中提取了谱特征。随后，探索了谱聚类来识别相似节点，从而提高排序质量。[16] 构建了一个超图，利用标签计算超图拉普拉斯矩阵进行图正则化。然后，在重排序任务中应用图正则化来平滑排序结果。尽管这些浅层模型已经得到了广泛研究，但它们不足以探索图中复杂的连接。

2.3 基于图的推荐方法

图神经网络 (GNNs) 提供了一种深入探索图结构的有效方法 [1]。嵌入通过单一的传播层传播到一跳邻居，并通过堆叠多层传播到多个跳的邻居。在推荐任务中，我们将每个节点的嵌入（即高阶偏好表示）传播给其邻居（具有相似交互行为的用户和物品），确保它们具有相似的偏好表示。如我们所见，GNN 明确地利用了协同信息。[1] 提出了一种用于显式反馈评分数据的 GNN 模型，其中节点嵌入通过基于评分等级 R 构建的 R 个图进行并行传播。[17] 提出了一种用于隐式反馈数据的多层 GNN。在每次传播过程中，考虑了元素级的偏好相似性，从而增强了模型的性能。[4] 提出了一种高效的 GNN 模型，专门用于大规模推荐。该模型利用传播丢弃来提高效率，同时利用注意力机制和自适应学习提升效果。由于嵌入是推荐模型中最重要的组成部分，因此 GNN 为探索推荐任务中的深层结构提供了一种更有效的方法。然而，如前所述，观察到的图非常稀疏，大多数链接缺失。现有的 GNN 通过构建多跳连接来缓解这个问题，但在长距离传播过程中，嵌入会显著衰减。此外，传播策略是粗略和经验性的，仍有很大的改进空间。为了应对这些问题，本文的作者设计了一种 GNN，通过从数据中学习完整的图并利用它进行传播。

3 预备知识

在本节中，我们以最简单但有效的图神经网络 (GNN) LightGCN [8] 为例，介绍一些关于现有 GNN 的预备知识。粗体大写字母表示矩阵，粗体小写字母表示向量，斜体字母表示标量。我们使用矩阵 $\mathbf{R} \in \mathbb{R}^{M \times N}$ 来表示用户与物品之间的交互（其中 M 为用户数量， N 为物品数量）。如果用户 u 与物品 i 发生过交互，则 $R_{ui} = 1$ ；否则， $R_{ui} = 0$ 。我们的任务是预测矩阵 \mathbf{R} 中缺失的值（即 0），并根据预测结果为每个用户返回排名靠前的物品。

嵌入矩阵 $E \in \mathbb{R}^{(M+N) \times K}$ ，其中 K 为嵌入的维度。 $E_{1:M}$ 表示用户嵌入，记作 E^U ， $E_{M+1:M+N}$ 表示物品嵌入，记作 E^I 。传播矩阵定义为 $P = D^{-1}A$ （左归一化）或 $P = D^{-1/2}AD^{-1/2}$ （对称归一化），其中 $A = \begin{bmatrix} 0 & R \\ R^\top & 0 \end{bmatrix}$ 是二分图的邻接矩阵， D 是对角度矩阵，其中第 i 个对角元素表示第 i 个节点的度： $D_{ii} = \frac{1}{\sqrt{N_i N_j}}$ ，其中 N_i 是节点 i 的邻居数量。

3.1 传播层

现在，我们说明在一个传播层中嵌入传播的策略。对于一对连接的节点 (i, j) ，我们定义从节点 j 到节点 i 的嵌入为：

$$\mathbf{e}_{i \leftarrow j} = \mathbf{P}_{ij} \mathbf{E}_j,$$

其中 P_{ij} 是传播权重：如果 $P = D^{-1}A$ ，则 $P_{ij} = \frac{1}{\sqrt{|N_i|}}$ 。如果 $P = D^{-1/2}AD^{-1/2}$ ，则 $P_{ij} = \frac{1}{\sqrt{|N_i||N_j|}}$ 。然后，将节点 i 所有邻居的嵌入聚合在一起，以精炼表示：

$$E'_i = \sum_{j \in N_i} \mathbf{e}_{i \leftarrow j},$$

其中 E'_i 表示节点 i 在传播后的嵌入。表示为矩阵形式，我们得到： $E' = PE$ 。

3.2 多层结构

在深度 GNN 中，一跳传播通过一个传播层实现，多跳传播通过堆叠多个传播层来实现。第 l 层的嵌入递归地表示为：

$$E^{(l)} = PE^{(l-1)},$$

其中 $1 \leq l \leq L$ (L 为传播层数), $E^{(0)}$ 表示嵌入层的嵌入。可以看出, 在 GNN 中, 嵌入可以通过图传播到较远的距离。每个节点都被 L 跳内的邻居增强, 并通过利用拓扑信息来精炼表示。预测嵌入 $E \in \mathbb{R}^{(M+N) \times K_1}$ 是每一层输出的组合: $E = \text{Comb}(E^{(0)}, \dots, E^{(L)})$, 其中 $\text{Comb}(\cdot)$ 是组合函数。在本文中, 我们遵循 [8] 使用加权和作为组合函数:

$$\mathbf{E} = \sum_{l=0}^L \frac{1}{l+1} \mathbf{E}^{(l)} \quad (1)$$

预测结果由预测嵌入给出:

$$\hat{\mathbf{R}} = \mathbf{E}^U \mathbf{E}^I{}^\top \quad (2)$$

3.3 模型学习

Bayesian Pairwise Learning to Rank (BPR) 方法 [7] 被广泛用作隐式反馈的损失函数, 其目标是最大化用户对一对正负物品偏好的差距:

$$\mathcal{L} = \sum_{(u,i,j) \in \mathcal{D}} -\ln \sigma(\hat{R}_{ui} - \hat{R}_{uj}) + \frac{\lambda}{2} \|\mathbf{E}^{(0)}\|_F^2 \quad (3)$$

其中, $\sigma(\cdot)$ 是 sigmoid 函数, \hat{R} 的定义如公式 (2) 所示; $\mathcal{D} = \{(u, i, j) | R_{ui} = 1 \wedge R_{uj} = 0\}$ 是成对的训练数据, 其中 R 是观测到的交互矩阵; $\|\cdot\|_F$ 表示矩阵的 Frobenius 范数。在 LightGCN 中, $\mathbf{E}^{(0)}$ 是模型的可训练参数。

这里, 我们回顾图 1 中的二分图。在该示例中, 传播深度设置为 1。白色条表示原始嵌入, 灰色条表示传播后的嵌入, 蓝色条表示合并后的嵌入。如图 1 所示, 每个节点的嵌入通过观测到的链接传播到邻居节点。论文采用对称归一化, 线上的数字表示传播权重。

3.4 自传播的图神经网络

3.4.1 自传播

图结构是我们需要学习的目标, 同时也是我们用于传播的工具。受到这一启发, 我们使用新学到的图作为传播的工具。与现有 GNN 中经验性且稀疏的传播矩阵 $P = D^{-1}A$ 或 $P = D^{-1/2}AD^{-1/2}$ 不同, 我们的自传播方法中的传播矩阵 P 是数据驱动且稠密的。传播权重 P_{ij} 是节点 i 和节点 j 之间的预测相似度。图 3 展示了我们的自传播过程。使用图 1 中所示的二分图作为例子, 传播深度设置为 1。线上数字是数据驱动的传播权重。论文作者的模型潜在的链接也被考虑在内, 传播权重是通过数据构建的。

在实际应用中, 传播矩阵 P 是非常大的且稠密的。考虑到 P 占用大量空间存储, 且传播操作 PE 计算起来非常耗时, 我们需要在模型训练过程中提供一种高效的方式来存储和更新

P 。由于 P 表示节点之间的相似度，它应该是一个低秩矩阵，因此我们通过矩阵分解来重构它：

$$P = nFF^\top$$

其中 $F \in \mathbb{R}^{(M+N) \times K_2}$ 被称为传播嵌入，用以区分预测嵌入 E ， n 是一个训练得到的数值，用来控制传播嵌入的大小。 F 编码了图的连接信息，而 $P_{ij} = nF_iF_j^\top$ 则是节点 i 和节点 j 相似度的预测值。

在传播操作中，我们可以通过改变计算顺序，避免显式构建 P ： $PE = nF^\top E$ 。由于中间矩阵 $F^\top E \in \mathbb{R}^{K_1 \times K_2}$ 是一个小矩阵，这大大减少了传播过程中的空间和时间消耗。

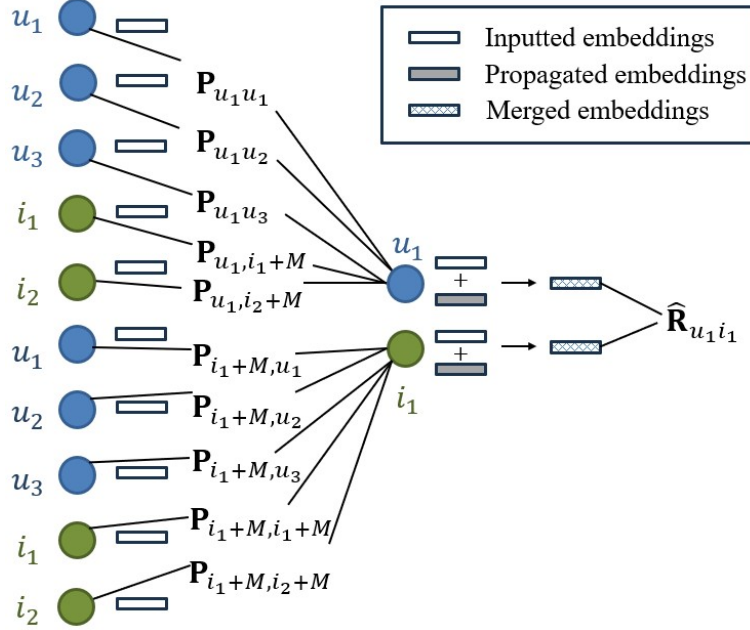


图 3. SGNN 框架概览图

3.4.2 多层结构

在本小节中，我们介绍了具有自传播的 GNN 网络结构，称为 SGNN。在本文中，我们遵循 LightGCN [8] 的结构来设计我们的模型。在 LightGCN 中，去除了 GNN 中一些常用的组件，如变换矩阵 [4]、激活函数和非线性聚合函数。图 3 展示了 SGNN 的概览。此外，我们使用加权和替代了拼接方式来结合来自不同层次的嵌入。SGNN 是一个多层结构。它包含一个嵌入层和 L 个传播层。在每个传播层中，前一层的嵌入作为输入，通过图传播并在每个节点上进行聚合，从而得到当前层的输出嵌入。

对于第 l 层传播，输出的嵌入为：

$$E^{(l)} = \tanh(nF(F^\top E^{(l-1)})) \quad (4)$$

[8] 提到，sigmoid 和 ReLU 函数对偏好预测有负面影响，而我们发现 \tanh 函数对于嵌入是一个很好的激活函数。预测嵌入由式 (1) 给出，预测结果由式 (2) 给出。

3.5 构建传播嵌入

为了构建节点相似度预测矩阵 \mathbf{P} ，论文作者提出了三种方法：

3.5.1 推荐模型

推荐系统的设计目标是预测用户-项目交互图中的缺失链接，因此我们可以使用推荐模型来预测 \mathbf{P} 。我们称结合推荐模型的 SGNN 为 SGNN_RM。对于用户 u 和项目 i ，其相似度为： $P_{u,M+i} = nf(F_u, F_{M+i}, \Theta)$ ，其中 $f(\cdot, \Theta)$ 是推荐模型， Θ 是网络参数， F_u 和 F_{M+i} 分别是用户和项目嵌入。我们可以选择不同的推荐模型 $f(\cdot, \Theta)$ ，例如矩阵分解 (MF) [6]、宽深结构 [3] 和 GNNs [2]。如前所述，我们需要对 P 进行矩阵分解，以避免显式构建它，因此我们采用矩阵分解 (MF)： $f(F_u, F_{M+i}, \Theta) = F_u F_{M+i}^\top$ 为了学习稠密图，我们通过最小化下式来更新 F ： $\frac{1}{2} \|\mathbf{R} - \mathbf{F}^U \mathbf{F}^I{}^\top\|_F^2$ 其中 $F^U = F_{1:M}$ 表示用户传播嵌入， $F^I = F_{M+1:M+N}$ 表示项目传播嵌入。

稠密传播矩阵可以表示为： $\mathbf{P} = n\mathbf{F}\mathbf{F}^\top = n \begin{bmatrix} \mathbf{F}_u \mathbf{F}_u{}^\top & \mathbf{F}_u \mathbf{F}_I{}^\top \\ \mathbf{F}_I \mathbf{F}_u{}^\top & \mathbf{F}_I \mathbf{F}_I{}^\top \end{bmatrix}$ ，其中 $\mathbf{F}^U \mathbf{F}^U{}^\top$ 、 $\mathbf{F}^U \mathbf{F}^I{}^\top$ 和 $\mathbf{F}^I \mathbf{F}^I{}^\top$ 分别记录了用户-用户传播、用户-项目传播和项目-项目传播的权重。在我们的模型中，用户-项目传播由交互矩阵 \mathbf{R} 进行监督。为了更好地学习用户-用户和项目-项目传播，我们还可以通过最小化以下损失函数，使用社交网络 \mathbf{S} 和知识图谱 \mathbf{K} 作为监督信号： $\frac{1}{2} \|\mathbf{S} - \mathbf{F}^U \mathbf{F}^U{}^\top\|_F^2$ 和 $\frac{1}{2} \|\mathbf{K} - \mathbf{F}^I \mathbf{F}^I{}^\top\|_F^2$

对于多层 SGNN_RM，第 l 层的传播由公式 (4) 给出；预测嵌入由公式 (1) 给出；偏好预测由公式 (2) 给出；损失函数为：

$$\mathcal{L} = \sum_{(u,i,j) \in \mathcal{D}} -\ln \sigma(\hat{\mathbf{R}}_{ui} - \hat{\mathbf{R}}_{uj}) + \frac{1}{2} \|\mathbf{R} - \mathbf{F}^U \mathbf{F}^I{}^\top\|_F^2 + \frac{\lambda}{2} (\|\mathbf{E}^{(0)}\|_F^2 + \|\mathbf{F}\|_F^2) \quad (5)$$

其中 D 是训练数据集， \hat{R}_{ui} 是对用户 u 和项目 i 的预测， \mathbf{F}^U 和 \mathbf{F}^I 是用户和项目的传播嵌入， λ 是正则化项的系数。通过这种方式，从数据中共同学习节点表示和传播。在 SGNN_RM 中， $E^{(0)}$ ， F ，和 n 是可训练的参数。

3.5.2 谱特征

另一种广泛使用的衡量节点相似度的方法是谱聚类 [18]。在谱聚类算法中，通过特征分解分解图的归一化邻接矩阵，得到特征向量，这些修剪后的特征向量随后用于聚类。作者探索了谱特征在构建传播中的应用，并称该模型为 SGNN_SF。二分图的邻接矩阵为： $\mathbf{A} = \begin{bmatrix} 0 & \mathbf{R} \\ \mathbf{R}^\top & 0 \end{bmatrix}$ ，对其进行对称归一化： $\tilde{\mathbf{A}} = D^{-\frac{1}{2}} \mathbf{A} D^{-\frac{1}{2}}$ ，其中 D 为度矩阵，归一化后的传播矩阵 $\tilde{\mathbf{A}}$ 是常规 GNN 中的传播矩阵。特征分解为： $\tilde{\mathbf{A}} = \Phi \Lambda \Phi^\top$ ，其中 $\Phi \in \mathbb{R}^{(M+N) \times (M+N)}$ 是特征向量矩阵， $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{M+N}) \in \mathbb{R}^{(M+N) \times (M+N)}$ 是特征值矩阵，特征值按降序排列。谱特征 \mathbf{F} 定义为前 K_2 个特征向量： $\mathbf{F} = \Phi_{*,1:K_2}$ 。当 $\tilde{\mathbf{A}}$ 为稀疏矩阵时，可以通过 Lanczos 方法高效构造 \mathbf{F} [?]。如果有社交网络 \mathbf{S} 和知识图谱 \mathbf{K} 可用，我们可以通过增强用户-用户和项目-项目的连接来进一步改进。在这种情况下，邻接矩阵为： $\mathbf{A} = \begin{bmatrix} \mathbf{S} & \mathbf{R} \\ \mathbf{R}^\top & \mathbf{K} \end{bmatrix}$ 。

由于谱特征能够很好地控制幅度，因此我们去掉公式 (3) 中的 n ，因此传播为：

$$E^{(l)} = \tanh(\mathbf{F}(\mathbf{F}^\top E^{(l-1)}))$$

类似于 SGNN_RM，预测嵌入由公式 (1) 给出；偏好预测由公式 (2) 给出；损失函数由

公式 (3) 给出。在 SGNN_SF 中, $E^{(0)}$ 是可训练的参数, \mathbf{F} 是预构建的并且在训练期间保持固定。

3.5.3 预测嵌入

论文作者还设计了 SGNN_PE, 用于基于预测嵌入构建传播嵌入: $\mathbf{F}^{(l)} = \sum_{l'=0}^{l-1} \frac{1}{1+l'} E^{(l')}$ 对于多层 SGNN_PE, 第 l 层的传播为:

$$E^{(l)} = \tanh(n\mathbf{F}^{(l)} (\mathbf{F}^{(l)\top} E^{(l-1)}))$$

预测嵌入由公式 (1) 给出; 偏好预测由公式 (2) 给出; 损失函数由公式 (3) 给出。在 SGNN_PE 中, 节点表示和传播是由相同的嵌入来建模的。 $E^{(0)}$ 和 n 是可训练的参数。

4 复现细节

4.1 与已有开源代码对比

论文已有源码, 更改小部分代码适应版本与环境, 复现原论文。

4.2 实验

本节通过与一些最新的模型进行比较, 展示了论文提出的模型的实用性。利用两个真实世界的数据集验证模型的有效性。

4.2.1 数据集

本文采用了两个真实世界的数据集: *Amazon* 和 *Movielens* 来训练所有模型。

- *Amazon* 数据集 [19] 是从电子商务网站 *Amazon.com* 收集的用户评论数据。在本文中, 我们采用了“电子”类别的 5-core 版本 (在使用 n -core 过滤时, 我们移除与交互次数少于 n 的用户和物品)。
- *Movielens* 数据集 [20] 是通过电影网站 *movielens.umn.edu* 收集的, 我们使用了 1M 版本。该电影评分数据集被广泛用于评估协同过滤算法。

这两个数据集均为显式反馈数据 (评分数据)。为了构造隐式反馈, 如果用户 u 对物品 i 进行了评分, 则将交互 (u, i) 设置为“1”, 否则为“0”。我们分别对 *Amazon* 和 *Movielens* 数据集进行了 10-core 和 20-core 过滤以减少数据的稀疏性。过滤后的数据集如表 1 所示。

每个数据集随机划分为训练集 (80%)、验证集 (10%) 和测试集 (10%)。我们在训练集上训练模型, 在验证集上确定所有超参数, 并在测试集上报告性能。

4.2.2 方法

对比以下方法, 以证明论文所提出的模型的有效性。

表 1. 数据集统计

数据集	交互数	用户数	物品数	稀疏率
Amazon	347,393	20,247	11,589	99.8519%
MovieLens	995,154	6,022	3,043	94.5694%

- **MF**: 该矩阵分解 (Matrix Factorization) 方法是一个基础但具有竞争力的物品推荐基线方法 [6]。该方法通过学习嵌入 (embeddings) 来编码用户偏好, 预测结果由嵌入在潜在空间中的距离决定。
- **GCMC**: 图卷积矩阵补全 (Graph Convolutional Matrix Completion) 方法利用图自编码器 (graph auto-encoder) 通过图传播特征 [1]。在实验中, 特征是可学习的嵌入。考虑到 GCMC 是为显式反馈 (explicit feedback) 设计的, 我们设置 $R = 1$ (即只有一个评分等级)。
- **NGCF**: 神经图协同过滤 (Neural Graph Collaborative Filtering) 是针对隐式反馈 (implicit feedback) 最有效的多层 GNN 模型之一 [17]。NGCF 通过堆叠 L 层将嵌入传播到节点的 L 跳图邻居。聚合方式包括求和和逐元素乘积。
- **LightGCN**: 轻量图卷积网络 (Light Graph Convolutional Network) 是当前最先进的 GNN 推荐模型之一。LightGCN 简单且有效, 移除了 GNN 中一些常用的组件, 例如变换矩阵、激活函数和非线性聚合函数。此外, 用加权求和替代了嵌入层的拼接操作。
- **LCFN**: 低通协同过滤网络 (Low-pass Collaborative Filter Network) 方法使用原始图卷积来设计深层网络, 并通过低通图滤波器使其适用 [21]。卷积核从数据中可训练, 因此 LCFN 可以从图中提取高级特征。
- **SGNN**: 自传播图神经网络 (Self-propagation Graph Neural Network) 是本文提出的 GNN 模型。SGNN 首先预测并完成用户-物品图, 然后通过完成的图传播嵌入。由于该图通过考虑多跳传播进行重构, SGNN 提供了一种利用多跳传播的新方法。论文作者验证了 SGNN 的三个变体的有效性: SGNN_RM、SGNN_SF、SGNN_PE。

需要注意的是, MF 和 GCMC 是为显式反馈设计的, 作者在实验中使用 BPR 来优化所有模型。

4.2.3 评估指标

为了在隐式反馈场景下评估我们提出的模型和基线的性能, 我们对验证集/测试集中的每位用户进行全物品排序, 并向用户推荐前 k 个物品。然后, 我们采用两个指标: $F1\text{-score}@k$ 和归一化折扣累计增益 (*Normalized Discounted Cumulative Gain*, $NDCG@k$) 来评估推荐质量。 $F1\text{-score}$ 定义为精确率 (Precision) 和召回率 (Recall) 的调和平均数, 被广泛用于测试二分类器的性能。 $NDCG$ 是一个位置敏感的指标, 通常用于衡量排序质量。在验证集上调参时, 我们设置 $k = 2$; 在测试集上测试时, 我们设置 $k = \{2, 5, 10, 20, 50, 100\}$ 。

4.2.4 参数设置

为了公平比较，作者在实验中使用以下策略对所有模型进行调参：最大迭代次数设置为 200。在每次迭代中，枚举所有正样本以训练模型，然后进行测试。学习率 η 和正则化系数 λ 通过在粗粒度范围 $\{0.0001, 0.001, 0.01\} \otimes \{0.001, 0.01, 0.1\}$ 上进行网格搜索来确定，然后在粗调结果的基础上进行细粒度调节。例如，如果某个模型在 $\eta = 0.01$ 和正则化系数 $\lambda = 0.1$ 时获得最佳性能，作者将其在细粒度范围 $\{0.002, 0.005, 0.01, 0.02, 0.05\}$ 和 $\{0.02, 0.05, 0.1, 0.2, 0.5\}$ 上进行调优。批次大小设置为 10,000，以利用 GPU。作者评估不同维度的预测和传播嵌入 K_1 和 K_2 ，范围为 $\{8, 16, \dots, 128\}$ 。层数 L 在范围 $\{1, 2, \dots, 5\}$ 内确定。作者在每个参数设置下运行所有模型 5 次进行模型调参，并进行 10 次测试，报告平均性能。所有深度模型的嵌入层和 SGNN 的传播嵌入均通过预训练的矩阵分解 (MF) 初始化。

4.2.5 传播选择

在本小节中，我们从 SGNN_RM、SGNN_SF 和 SGNN_PE 中选择最有效的模型作为最终的 SGNN。在验证集上的性能如表 2 所示。表 2 显示 SGNN_SF 表现最佳。有两个可能的原因可以解释实验结果：(1) 嵌入在学习用户偏好方面效果很好，而谱特征在编码图结构方面效果很好。(2) 端到端模型 (SGNN_RM 和 SGNN_PE) 的表现不如两步模型 (SGNN_SF)，因为端到端模型难以训练。为了验证哪个原因是真正的原因，作者通过预训练的 MF 初始化 SGNN_RM 的传播嵌入，并在训练期间固定它们，作者没有观察到明显的改进。因此，原因 (1) 是真正的原因。

表 2. 推荐性能 (Amazon, 验证集)

模型	F_1 -score	NDCG@2
SGNN_RM	0.01572	0.01984
SGNN_SF	0.02071	0.02616
SGNN_PE	0.01575	0.01957

5 实验结果分析

在本小节中，报告了 SGNN 及其基线模型的预测性能，以展示其有效性。所有模型在两个数据集上的性能如表 3 所示。特别值得注意的是，我们遵循文献 [21] 的做法，将所有模型的预测嵌入的维度固定为 128，以进行公平比较。在 GCMC、NGCF 和 LCFN 中，来自不同层的嵌入通过连接组合，因此嵌入层 $\mathbf{E}^{(0)}$ 的嵌入维度设置为 $\frac{128}{L+1}$ 。我们可以看到，在这种情况下，GCMC 和 NGCF 相较于浅层 MF 并没有取得明显的改进。

接着讨论两个有趣的观察结果。首先，比较 *Amazon* 和 *MovieLens*，很明显，稀疏度更高的数据集 (*Amazon*) 更难以预测。然而，高级模型在 *Amazon* 上相较于基础模型的改进比在 *MovieLens* 上的要大。原因可能是 *MovieLens* 数据集的稀疏度较低，因此观察到的数据足以监督模型训练 (*MovieLens* 中未曝光的正样本较少)，在这种情况下，连接信息的帮助较小。另一个引起我们兴趣的观察结果是，随着推荐列表长度 k 的增加，所有模型之间的性能差距都在缩小。可能的原因是，所有模型都是根据 $F1$ -score@2 进行调整的，因此在前两项推

表 3. 推荐性能 (测试集)

数据集	评估指标		MF	GCMC	NGCF	LightGCN	LCFN	SGNN
<i>Amazon</i>	<i>F1-score@</i>	2	0.01321	0.01136	0.00889	0.01655	0.01404	0.01797
		5	0.01439	0.01195	0.00914	0.01701	0.01462	0.01789
		10	0.01340	0.01099	0.00825	0.01520	0.01346	0.01565
		20	0.0113	0.00942	0.00676	0.01255	0.01141	0.01283
		50	0.00826	0.00687	0.00498	0.00884	0.00832	0.00906
		100	0.00612	0.00520	0.00380	0.00643	0.00616	0.00658
	NDCG@	2	0.01680	0.01465	0.01092	0.02086	0.01790	0.02265
		5	0.02285	0.01924	0.01502	0.02785	0.02384	0.02951
		10	0.02910	0.02420	0.01906	0.03454	0.03008	0.03609
		20	0.03644	0.03045	0.02338	0.04234	0.03760	0.04382
		50	0.04812	0.04019	0.03059	0.05418	0.04960	0.05590
		100	0.05837	0.04944	0.03768	0.06450	0.06017	0.06632
<i>Movielens</i>	<i>F1-score@</i>	2	0.07320	0.05898	0.03488	0.07301	0.07570	0.08125
		5	0.11330	0.09180	0.05310	0.11306	0.11467	0.12562
		10	0.14111	0.11250	0.06707	0.13821	0.14069	0.15267
		20	0.15675	0.12471	0.07884	0.15209	0.15592	0.16658
		50	0.15095	0.12342	0.07889	0.14632	0.14947	0.15747
		100	0.12767	0.10826	0.07228	0.12420	0.12658	0.13143
	NDCG@	2	0.24796	0.21937	0.14161	0.24870	0.25174	0.26738
		5	0.23153	0.19974	0.12673	0.23213	0.23291	0.25072
		10	0.23173	0.19314	0.12150	0.22938	0.23235	0.25070
		20	0.24861	0.20129	0.12665	0.24391	0.25014	0.26796
		50	0.29914	0.24076	0.14990	0.29241	0.30033	0.31948
		100	0.35095	0.28865	0.18283	0.34377	0.35191	0.36981

荐上表现良好,但对于较大的 k 则调整不佳。这就是 SGNN 在 k 增加时相对于 LightGCN 和 LCFN 的性能提升减少的原因。这并不是因为 SGNN 在 k 较大时无效,而是因为在这种情况下没有得到很好的调整。实际上,当 k 较大时,LightGCN 和 LCFN 相对于 MF 的改进也是边缘的。因此,前两项推荐实验结果最具说服力。为了在较大的 k 下获得更好的性能,我们可以根据 $F1\text{-score}@100$ 重新调整所有模型。

6 总结与展望

在本次复现工作中,复现出来的结果如表3所示,其中的 NGCF 的测试结果与原文有所出入,初步考虑可能是设备原因所致,但在整个阅读文献与复现过程中我收获颇丰,对于推荐系统的相关研究有所了解,也为后续研究工作打下基础。论文关注图神经网络 (GNNs) 中的偏差和不完全传播问题。考虑到图既是需要预测的训练目标,也是使用的工具,我们使用

从数据中学习到的完整图来传播嵌入。由于学习一个密集的传播矩阵需要大量的空间和时间，作者通过矩阵分解（MF）来分解它，并提出了三种密集传播的方法：(1) 基于推荐模型完成图；(2) 基于谱聚类；(3) 基于预测嵌入。实验表明，第二种方法最为有效。通过堆叠多个层，嵌入可以在图中长距离传播。考虑到以上所有因素，作者提出了自传播图神经网络（SGNN），并针对隐式反馈进行成对学习优化。实验表明，SGNN 显著优于现有的最先进的 GNNs。对于未来的工作，可以将社交网络和知识图谱整合到模型中。

参考文献

- [1] R. V. D. Berg, T. N. Kipf, and M. Welling. Graph convolutional matrix completion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018.
- [2] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
- [3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.
- [4] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 974–983, 2018.
- [5] Wenhui Yu, Xiao Lin, Jinfei Liu, Junfeng Ge, Wenwu Ou, and Zheng Qin. Self-propagation graph neural network for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 34(12):5993–6002, 2022.
- [6] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81(2009):1–10, 2009.
- [7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AAAI, 2009.
- [8] 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*, 2020.
- [9] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep

- learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.
- [10] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T.-S. Chua. Outer product-based neural collaborative filtering. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2227–2233, 2018.
 - [11] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 335–344, 2017.
 - [12] Ruining He and Julian McAuley. Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.
 - [13] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. Aesthetic-based clothing recommendation. In *Proceedings of the 2018 world wide web conference*, pages 649–658, 2018.
 - [14] Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. *Advances in neural information processing systems*, 28, 2015.
 - [15] Wenhui Yu and Zheng Qin. Spectrum-enhanced pairwise learning to rank. In *The world wide web conference*, pages 2247–2257, 2019.
 - [16] Fuli Feng, Xiangnan He, Yiqun Liu, Liqiang Nie, and Tat-Seng Chua. Learning on partial-order hypergraphs. In *Proceedings of the 2018 world wide web conference*, pages 1523–1532, 2018.
 - [17] 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.
 - [18] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. 14th Int. Conf. Neural Inf. Process. Syst. Natural Synthetic*, pages 849–856, 2001.
 - [19] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 507–517, 2016.
 - [20] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):19:1–19:19, 2016.
 - [21] W. Yu and Z. Qin. Graph convolutional network for recommendation with low-pass collaborative filters. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 10936–10945, 2020.