基于图学习的联邦推荐算法研究与实现

摘要

在当今时代，推荐系统已被广泛地应用于各种在线服务平台，在海量物品中为用户挑选和推荐符合其兴趣和需求的物品，有效地缓解了信息过载的问题。然而，随着人们隐私保护意识的提高和相关法律法规的颁布，严重依赖于用户隐私数据的传统集中式推荐系统面临困境。为此，研究人员提出了将联邦学习和推荐系统相结合的联邦推荐，在保护用户隐私数据的同时，还能够训练出相对有效的推荐模型。早期的联邦推荐算法大多基于矩阵分解这类技术，这类联邦推荐算法发展至今，其推荐性能已达瓶颈，难以再有提升。因此，在保证用户的隐私安全的同时，设计一个性能更强的联邦推荐算法意义重大。

为了进一步提高联邦推荐算法的性能，本文复现了一个基于图神经网络的联邦推荐算法框架FedPerGNN，并将它和一个性能优异的基于图神经网络的集中式推荐算法LightGCN 相结合，提出了一个基于图学习的跨用户联邦推荐算法FedLightGCN，并应用到单类协同过滤问题上。在成功设计算法后，我们进行了性能对比实验和消融实验来验证FedLightGCN和其内部组件的有效性，并通过一系列的超参数实验来探究了FedLightGCN的推荐性能、隐私保护和通信代价这三者之间的关系。

通过理论分析和实验证明，我们提出的FedLightGCN的推荐性能普遍优于基于矩阵分解的联邦推荐算法，其通信代价较FedPerGNN也有一定程度的下降，同时能够较好地保护用户的隐私数据。FedLightGCN的推荐性能往往和隐私保护、通信代价相互制约，因此我们在选择超参数时应当平衡好这三者之间的关系。

关键词：推荐系统；联邦学习；图学习；单类协同过滤

1 引言

1.1 联邦推荐算法的研究背景

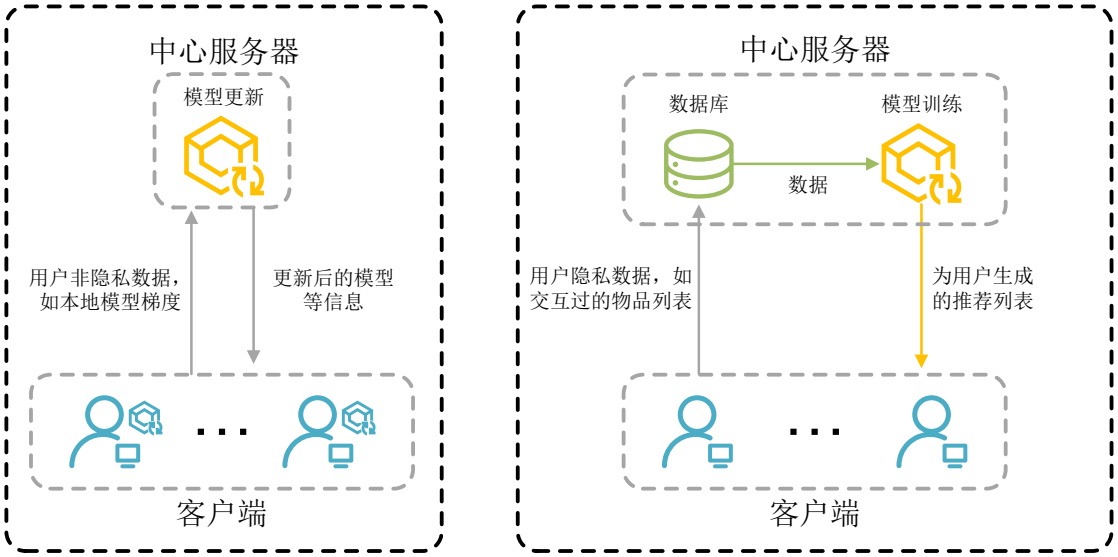
推荐系统（Recommender System, RS）是一种基于数据分析的信息过滤技术，通过对用户历史行为和偏好的分析和一定的算法，快速、准确地在海量物品中过滤掉用户不太可能产生交互行为的物品，并为用户推荐符合其兴趣和需求的物品，从而大大缓解了信息过载[23]的问题。在当今大数据和人工智能时代，推荐系统已被广泛地应用于各个领域，成为电子商务、新闻资讯和影音娱乐等众多在线服务平台的核心技术和重要引擎，极大地促进了商业发展和社会进步。

传统的推荐系统需要在中心服务器上集中存储和处理所有用户的数据。在过去，由于大众隐私保护意识的薄弱和相关法律法规的不健全，这种集中存储和处理数据的行为是普遍存

在的。然而，近年来，随着通用数据保护条例（General Data Protection Regulation, GDPR）[29]等保护隐私和数据安全的法律法规的相继颁布，以及人们隐私保护意识的提高，机构或组织收集用户隐私数据的难度大大提高，而不充分的数据难以训练出有效的算法模型，这给以数据驱动为主的传统集中式推荐系统带来了巨大的挑战 [13]。

为了解决这个问题，微众银行将谷歌提出的分布式学习框架——联邦学习 [17]应用到推荐系统中，并提出了联邦推荐系统 [28, 31]的概念。不同于传统的推荐系统需要在中心服务器集中存储和处理用户的隐私数据，联邦推荐系统不需要用户设备（也称为客户端）上传本地隐私数据。在联邦推荐系统中，客户端在本地执行模型训练和推荐过程，并上传梯度等非隐私数据至中心服务器，而中心服务器则聚合接收到的梯度等信息来更新全局模型，并发放更新后的模型等必要信息到客户端，来共同优化推荐模型，在达到令人满意的推荐效果的同时，保护了用户的隐私安全。联邦推荐系统与传统推荐系统的区别如图 1所示。

联邦推荐使得机构或组织能够在不侵犯用户隐私、符合相关法律法规的情况下，构建满足用户个性化需求的推荐模型，具有重大现实意义，其应用前景和学术价值巨大。因此，联邦推荐迅速引起了众多研究人员的关注和参与，成为推荐系统中一个备受瞩目的新兴子领域。



（a）联邦推荐系统　　（b）传统推荐系统

图 1. 联邦推荐系统与传统推荐系统的区别

1.2　联邦推荐算法的研究现状

在联邦推荐的早期发展阶段，其相关工作主要集中在将基于矩阵分解的协同过滤推荐算法（如PMF [18]、SVD++ [9]等）进行联邦化，从而产生了第一批联邦推荐算法。其中比较典型的联邦推荐算法包括应用于跨用户联邦推荐问题上的FCF [1]、FedRec [14]和FedRec++ [12]等，它们在满足中心服务器联合多方用户共同训练模型的需求、保证良好的推荐效果的同时，还保护了用户的隐私数据，如用户的评分数据和评分行为等。除此以外，还有应用于跨组织联邦推荐问题的 FedMF [3]、FCMF [27]等算法，它们使得不同的组织之间能够在不泄露隐私数据和商业机密的情况下进行知识迁移与共享，共同提高推荐效果。

作为联邦推荐领域的早期探索，上述研究具有重要的意义，然而随着时间的推移，它们的问题也开始显现：基于矩阵分解的协同过滤算法 [33]是一类已发展成熟的经典推荐算法，其简易性使其无法利用“用户——物品”的高阶交互信息和难以融合辅助信息如社交网络等，严重限制了以其为基础的首批联邦推荐算法的性能。

由于基于矩阵分解的联邦推荐算法的性能提升已达瓶颈，研究人员开始将目光转移到更具潜力的基于图学习的联邦推荐算法的研究。图学习是指在“图”这一数据结构上进行的机器学习，在后面的2.1节中会进行详细介绍。受到深度学习的启发，图学习领域发展出了一个分支——图神经网络，其强大的表征能力和对高阶交互信息的有效利用，使得它在传统集中式推荐系统中的应用取得了巨大的成功，这自然引起了联邦推荐领域的研究者们的注意。自2021年起，已有多篇基于图神经网络的联邦推荐算法研究的论文发表，比如基于图神经网络的联邦推荐通用算法框架——FedPerGNN [26]，将社交网络以图数据结构的形式融入联邦推荐的FeSoG [15]，和融合表示学习、用户聚类及模型自适应的PerFedRec [16]等算法。

基于图神经网络的联邦推荐算法，将图神经网络强大的表征能力和联邦推荐优异的隐私保护能力相结合，展现了巨大的发展前景和商业潜力。因此，研究基于图学习的联邦推荐算法意义重大，可以在保护参与方数据隐私的同时，进一步提高推荐性能，扩展联邦推荐的应用场景和价值。

1.3 本文的主要工作

本文所关注的推荐问题是，基于隐式反馈的物品排序问题，也称为单类协同过滤问题，其详细定义在3.1节中。对于该推荐问题，基于图神经网络的推荐算法表现优异，但它们对全局图的需求使其无法保护用户的隐私数据。而基于矩阵分解的联邦推荐算法，虽然可以保护用户的隐私数据，但由于结构简单、无法利用高阶信息等原因，往往推荐性能欠佳。

为了解决上述问题，本文复现了1.2节中提到的基于图神经网络的联邦推荐算法框架FedPerGNN [26]，并将它和一个性能优异的基于图神经网络的推荐算法 LightGCN [6]相结合，提出了一个基于图学习的跨用户联邦推荐算法FedLightGCN，并将之应用到单类协同过滤问题上。

在成功设计FedLightGCN算法后，我们在两个公开数据集Gowalla和Yelp2018上进行了大量的实验，包括：（1）性能对比实验。将FedLightGCN和基于深度学习的集中式推荐算法、基于矩阵分解的联邦推荐算法进行推荐性能的比较，验证了FedLightGCN的有效性；（2）消融实验。针对FedLightGCN中的一个重要组件——保护隐私的图扩展方法，设计了消融实验，验证了该组件的有效性；（3）超参数实验。探究了FedLightGCN的推荐性能和隐私保护、通信代价之间的关系。

通过理论分析和实验证明，我们提出的FedLightGCN具有如下几个特点：（1）推荐性能良好。虽然FedLightGCN的推荐性能相比于LightGCN有一定的损失，但相较于传统的联邦推荐算法则有较大的提升；（2）保护用户隐私。FedLightGCN确保了用户的原始敏感数据不被直接泄露和中心服务器难以反推用户的敏感数据，很好地保护了用户的隐私；（3）通信代价降低。相较于采用GAT [24]作为骨干模型的FedPerGNN，FedLightGCN采用了没有额外模型参数的LightGCN 作为骨干模型，这在一定程度上降低了客户端和服务器之间的通信代价。

2 相关工作

本文的重点在于研究、设计和实现一个基于图学习的联邦推荐算法，因此除推荐系统外，与本文相关的研究领域还有图学习和联邦学习。接下来，将分别介绍图学习和联邦学习的定义，及两者在推荐系统中的应用。

2.1 图学习

2.1.1 图学习的定义

图学习，即指在“图”这一数据结构上进行的机器学习。图的定义如下：一个图可以被表示为$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$，其中$\mathcal{V} = \{v_1, \cdots, v_N\}$是大小为$N = |\mathcal{V}|$的节点集合，$\mathcal{E} = \{e_1, \cdots, e_M\}$是大小为$M = |\mathcal{E}|$的边集合。

图描述了实体之间的两两关系，被广泛地应用于各个领域，作为这些领域中真实数据的表示方法。例如，在社会科学中，社交网络就是一个以用户作为节点、用户之间的社会关系（如关注、屏蔽等）作为边的图。又比如在化学中，化合物可以被表示为以原子为节点、以化学键为边的图。图的泛用性和潜在价值，使得图学习受到越来越多其他学科领域的关注。

图学习可大致分为四类：（1）基于图信号处理的方法。即将传统信号处理的概念扩展到图上，利用一些经典的信号处理技术如傅里叶变换等来分析和挖掘图中所蕴含的隐式信息；（2）基于矩阵分解的方法。该方法利用一个矩阵来表示图的某些特征如节点间的相似度等，而节点的嵌入则可以通过分解这个矩阵来得到。（3）基于随机游走的方法。该方法通过一定的游走策略在相连的节点之间不断游走采样，在生成节点序列的同时保留节点之间的原始关系。生成的节点序列被用于训练节点的特征向量，并给下游任务使用。在基于随机游走的方法中，具有代表性的算法包括DeepWalk [19]、node2vec [5]和LINE [22]等；（4）基于深度学习的方法。该方法也可以被称为图神经网络，是目前图学习中最热门的子领域之一，接下来我们将重点介绍它。

2.1.2 图神经网络

深度学习作为近十几年最热门的研究领域之一，其在各类任务中均表现优异，这自然引起了图学习领域的研究人员的注意。通过将各类神经网络应用到图上，图学习发展出了一个新分支——图神经网络 [32]。初期的图神经网络是在递归神经网络、循环神经网络和前馈神经网络的基础上进行修改的，虽然取得了一定的成功，但其本质上也只是在图上建立状态转换系统并不断迭代直至收敛，相应的可扩展性和表征能力有限。

随后，有研究人员将卷积操作引入到图神经网络中，并通过一系列近似和简化，提出了图卷积神经网络（Graph Convolutional Network, GCN） [8]。其核心组件——图卷积层通过共享的权重矩阵对节点特征进行变换，并聚合每个节点的相邻节点变换后的特征。经过k层这样的图卷积层后，每个节点可以融合其k阶邻域内所有节点的信息，从而捕获高阶邻域信息，大大增强了表征能力和模型性能，GCN也因此在半监督分类任务上较其他方法取得了大幅的性能提升。在GCN之后，也有学者将注意力机制引入图神经网络，提出了图注意力网络（Graph Attention Network, GAT） [24]，通过为节点的各相邻节点分配不同的权重，来区分它们的重要程度，从而获得了更好的表征能力。

GCN和GAT的提出大大促进了图神经网络的发展，此后陆续有新的图神经网络模型被提出，并在各类图相关的任务上表现优异，这也让越来越多其他领域的研究人员开始关注这种强有力的图学习方法，其中就包括推荐系统领域。

2.1.3 图神经网络在推荐系统中的应用

在推荐系统中，“用户——物品”的交互行为本身也可以表示为用户节点与交互过的物品节点相连的二分图的形式，因此推荐系统领域的研究人员开始尝试将大获成功的GCN等图神经网络应用到推荐系统中。

有研究人员通过将GCN应用到推荐系统的单类协同过滤问题上，提出了NGCF [25]，并取得了一定的成功。然而，从GCN中继承而来的特征变换和非线性激活这两个组件，实际上限制了NGCF在单类协同过滤这类简单场景下的性能，有研究人员通过消融实验发现了这一点，并提出了结构更为轻便简单、性能更加优秀的LightGCN [6]。LightGCN舍弃了 NGCF中冗余的特征变换和非线性激活这两个设计，大幅简化了模型结构，并在单类协同过滤问题上取得了更好的表现。

然而，LightGCN依赖于“用户——物品”的全局图，使得它需要收集用户的敏感数据，无法保护用户的隐私安全。正是基于该原因，本文将它和基于图神经网络的联邦推荐算法框架FedPerGNN相结合，在维持其推荐性能的同时保护了用户的隐私数据。

2.2 联邦学习

2.2.1 联邦学习的定义

联邦学习（Federated Learning, FL）本质上是一种新型的分布式学习框架，由谷歌于2017年首次提出 [17]。与传统的分布式机器学习将任务和数据分配到多个计算节点来提高模型训练效率的目的不同，联邦学习更侧重于保护参与节点的数据隐私 [34]，在此基础上再考虑计算效率和通信代价等。

一个联邦学习系统往往包含一个中心服务器和多个客户端，其中客户端既可以是个人用户，也可以是不同的机构或组织等。客户端的隐私敏感数据通常只保存在本地，不会以任何方式上传给中心服务器，而中心服务器则通过一定的协议来联合客户端进行模型训练，以此提高模型性能。该训练协议一般可概括为：（1）中心服务器在客户端集合中随机选择一部分客户端；（2）被选中的客户端从中心服务器中下载当前的全局模型参数；（3）被选中的客户端利用下载的模型和自己的本地数据来进行模型训练，并将训练过程中产生的中间结果如模型梯度等非隐私敏感的数据上传至中心服务器，在上传之前通常还会对这些数据进行加密加噪来进一步确保安全性；（4）中心服务器聚合从客户端收集到的模型梯度等数据，并进行全局模型的更新；（5）重复上述4步，直到模型收敛至期望值。

联邦学习发展至今，又可细分为横向联邦学习和纵向联邦学习。横向联邦学习，适用于各个参与方的数据特征空间重叠较多，而样本重叠较少的情况，例如某个电商平台中的各个用户，他们的数据都具有统一的形式（数据特征空间重叠），但每个用户之间又是独立的个体（样本不重叠）；而纵向联邦学习，则适用于参与方的样本重叠较多，而数据特征空间重叠较少的情况，例如同一地区的银行和医院的客户数据，由于两者在同一地区而客户重叠较多，但他们记录存储的客户数据形式却相差较大。联邦学习因为其兼顾模型性能和隐私保护

的特性，受到推荐系统领域的研究人员的关注，接下来我们介绍联邦学习在推荐系统中的应用。

2.2.2 联邦学习在推荐系统中的应用

正如前面的1.1节所述，联邦推荐就是联邦学习在推荐系统中的应用，其中横向联邦学习一般对应着跨用户联邦推荐，而纵向联邦学习则对应着跨组织联邦推荐。相关的联邦推荐算法已在1.2节中进行过介绍，此处不再赘述。

早期的基于矩阵分解的联邦推荐算法往往性能有限，因此本文将图神经网络和联邦推荐相结合，在保护用户隐私的同时，提高模型的推荐性能。

3 本文方法

3.1 研究问题定义

本文研究的问题是在跨用户联邦推荐场景下，基于隐式反馈的物品排序问题。下面分别给出两者的详细定义。

基于隐式反馈的物品排序问题，也称为单类协同过滤问题，是指通过研究用户过往的单一类型的隐式反馈行为，来预测这些用户未来的反馈行为，其中隐式反馈是指用户对物品的点击、浏览、收藏、加入购物车和购买等行为，它们无法明确表示用户对物品的喜好程度，区别于能明确表示用户对物品喜好程度的显式反馈，如用户对物品的数值评分和点赞等行为。该问题具体到本文，可表示如下：$\mathcal{U} = \{1, 2, 3, \ldots, n\}$和$\mathcal{I} = \{1, 2, 3, \ldots, m\}$表示用户集合和物品集合，其中$n$代表用户数量，$m$代表物品数量。对于每个用户$u$，我们将他曾经交互过（即存在隐式反馈）的所有物品，表示为物品集合$\mathcal{I}_u$，同时可知$\mathcal{I} = \bigcup_{u \in \mathcal{U}} \mathcal{I}_u$。而我们的目的是在该用户未交互过的物品集合$\mathcal{I} \setminus \mathcal{I}_u$中，预测他/她在未来最有可能进行交互的$topK$个物品，表示为物品集合$\mathcal{I}_u^{re}$，其中$topK$是由我们自己设置的超参数，可以是5、10、20等整数。

跨用户联邦推荐场景是指存在一个中心服务器和多个用户（即客户端，下面不再区分两者），这些客户端在该中心服务器的协调下共同训练算法模型。需要注意的是，客户端的本地隐私敏感数据，如用户对物品的评分数据和交互行为等，不能以任何方式泄露给中心服务器。具体到本文，即每个用户u的交互物品列表$\mathcal{I}_u$不能以任何方式泄露给中心服务器。

综上所述，本文研究的问题可表示为：存在一个中心服务器、n个客户端$\mathcal{U} = \{1, 2, 3, \ldots, n\}$和 m个物品$\mathcal{I} = \{1, 2, 3, \ldots, m\}$，这些客户端在中心服务器的调度下共同训练算法模型。对于每一个用户u，其交互过的物品（也称为正物品）集合表示为$\mathcal{I}_u$，且$\mathcal{I}_u$不能以任何方式泄露给中心服务器。最终，我们要在他/她未交互过的物品（也称为负物品）集合$\mathcal{I} \setminus \mathcal{I}_u$中，预测其在未来最有可能进行交互的$topK$个物品，即在客户端本地生成推荐物品列表$\mathcal{I}_u^{re}$。

3.2 模型总体框架

FedLightGCN的总体框架如图2所示。接下来，我们将介绍总体框架中的相关细节。

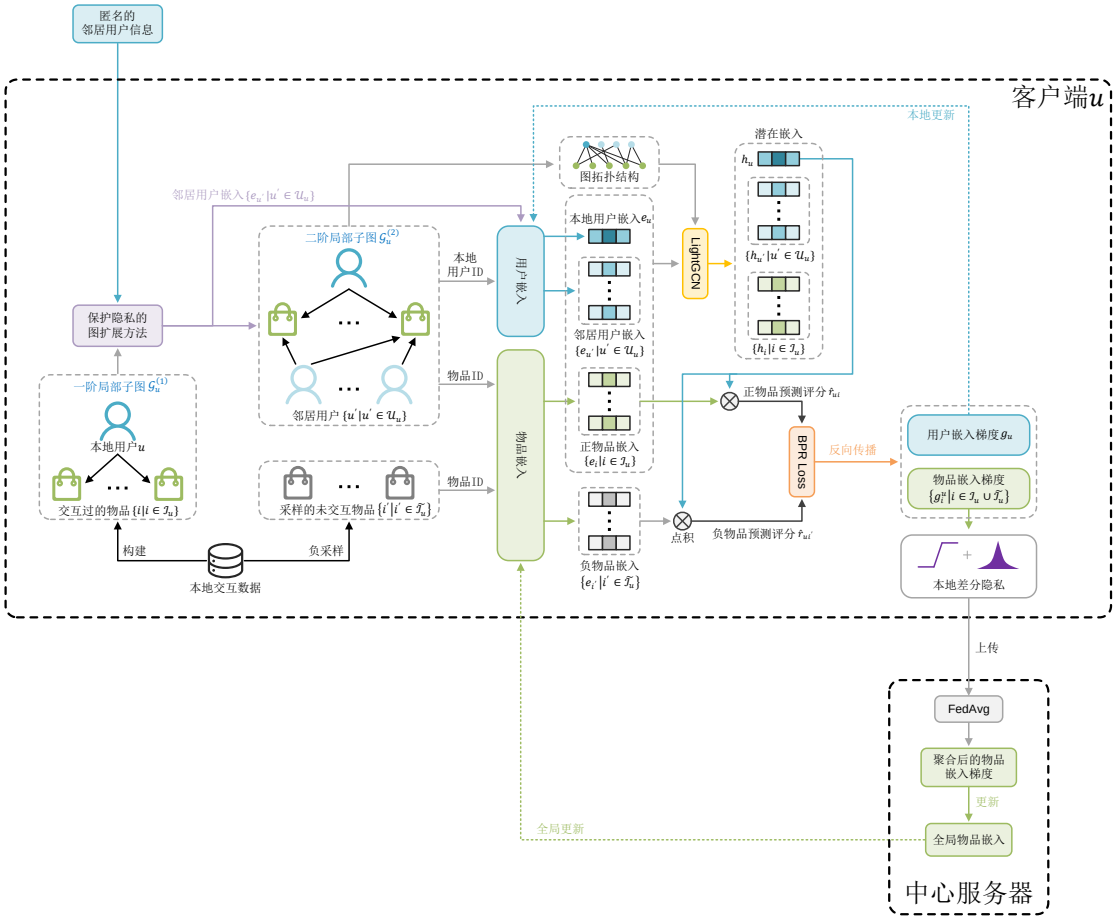


图 2. FedLightGCN的总体框架

在开始模型训练之前，中心服务器和客户端需要先进行初始化。对于中心服务器而言，它需要初始化每个物品i的嵌入$e_i \in \mathbb{R}^{1\times d}$，其中$d$表示嵌入的维度，并将所有物品嵌入$\{e_i|i \in \mathcal{I}\}$发送给所有客户端。与中心服务器类似，每个客户端$u$也需要初始化其所代表的用户的嵌入$e_u \in \mathbb{R}^{1\times d}$，此外还需要根据自己存储在本地的交互物品列表$\mathcal{I}_u$来构建一阶局部子图$\mathcal{G}_u^{(1)}$。

这里解释一下嵌入和一阶局部子图的含义。在推荐系统中，嵌入通常是指一个用户或物品被映射出的一个低维向量$\mathbb{R}^{1\times d}$，该嵌入通过特定初始化、神经网络训练等方式对该用户或物品的潜在特征进行编码，包含了大量有价值的信息，有助于推荐系统更好地理解用户和物品之间的关系。而客户端u中的一阶局部子图$\mathcal{G}_u^{(1)}$，如图2所示，则是指仅包含该客户端所代表的用户节点、曾经交互过的物品节点（即一阶邻域）和这些节点之间的边的局部子图，即$\mathcal{G}_u^{(1)} = \left\{\mathcal{V}_u^{(1)} = \{u\} \cup \mathcal{I}_u, \mathcal{E}_u^{(1)}\right\}$，区别于包含所有用户节点和物品节点的全局图。

完成初始化后，便可以开始模型的训练。考虑到效率和实际情况，在每一轮训练时，并不是所有客户端都参与本次训练，而是中心服务器从客户端集合$\mathcal{U}$中随机选取一个子集$\bar{\mathcal{U}}$，让该子集中的所有客户端参与本次训练。

对于被选中的客户端u，虽然它已经构建了一阶局部子图，但该子图所包含的信息较少，不利于模型训练。因此，我们参考FedPerGNN中保护隐私的图扩展方法 [26]，将客户端中的一阶局部子图扩展成二阶局部子图，该图扩展方法的相关细节在3.3节中。在该二阶局部子图中，除了原先的本地用户节点u及u交互过的物品节点集合$\mathcal{I}_u$外，还新增了与物品节点交互过的其他用户节点的集合$\mathcal{U}_u = \left\{u' \middle| u' \in \bigcup_{i\in\mathcal{I}_u} \mathcal{U}_i, u' \neq u\right\}$，其中$\mathcal{U}_i$表示与物品$i$进行过交互的用

户集合。我们将这些新增的用户节点称为u的邻居用户，它们与物品节点共同组成了u的完整二阶邻域，因此扩展后的图被称为二阶局部子图$\mathcal{G}_u^{(2)} = \left\{\mathcal{V}_u^{(2)} = \{u\} \cup \mathcal{I}_u \cup \mathcal{U}_u, \mathcal{E}_u^{(2)}\right\}$。

在图扩展后，一个嵌入层将本地用户节点u、u交互过的物品节点$\{i \,|i \in \mathcal{I}_u\}$和邻居用户节点 $\{u' \,|u' \in \mathcal{U}_u\}$映射为相应的节点嵌入$e_u$、$\{e_i \,|i \in \mathcal{I}_u\}$和 $\{e_{u'} \,|u' \in \mathcal{U}_u\}$。

接下来，骨干模型LightGCN将上述节点嵌入和二阶局部子图的拓扑结构（邻接矩阵和度矩阵）作为输入，对它们进行相应的卷积操作，并输出对应的节点潜在嵌入h_u、$\{h_i \,|i \in \mathcal{I}_u\}$和$\{h_{u'} \,|u' \in \mathcal{U}_u\}$。这些潜在嵌入聚合了高阶邻域节点嵌入的信息，具有更强大的表征能力。LightGCN的相关细节在3.4节中。

最终，用户u对某个物品i的预测评分（喜好程度或交互的可能性）可量化地表示为用户潜在嵌入h_u和该物品嵌入e_i的点积，即：

$$\hat{r}_{ui} = h_u {e_i}^T \tag{1}$$

通过式1，客户端u可以计算对所有未交互过的物品的预测评分$\{\hat{r}_{ui} \,|i \in \mathcal{I} \setminus \mathcal{I}_u\}$，取其中评分最高的$topK$个物品，即可在本地生成推荐物品列表$\mathcal{I}_u^{re}$。

在模型训练方面，我们选择BPR Loss [21]，它是一种基于成对偏好假设的损失函数，其思想是最大化正物品和负物品之间的预测评分之差，原始公式如下：

$$L_{BPR} = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}_u} \sum_{i' \in \mathcal{I} \setminus \mathcal{I}_u} \left[-\ln \sigma\left(\hat{r}_{ui} - \hat{r}_{ui'}\right) + \alpha(\|e_u\|^2 + \|e_i\|^2 + \|e_{i'}\|^2)\right] \tag{2}$$

其中，σ是sigmoid函数，即$\sigma(x) = \frac{1}{1+e^{-x}}$；$\alpha$为$L_2$正则化系数。

观察式2可以发现，每个客户端需要计算其每个正物品和每个负物品之间的预测评分之差，时间复杂度较高，不符合实际情况。因此，对于客户端u来说，它会从$\mathcal{I} \setminus \mathcal{I}_u$中随机采样一个子集$\widetilde{\mathcal{I}_u}$，再为$\widetilde{\mathcal{I}_u}$中的每个负物品$i'$随机采样一个正物品$i$，组成正、负物品对$(i, i')$，所有的正、负物品对记作集合$\mathcal{B}_u$。由于负采样的数量一般远大于正物品的总数$|\mathcal{I}_u|$，因此可以粗略认为$\{i \,|(i, i') \in \mathcal{B}_u\} = \mathcal{I}_u$。在客户端采样得到$\mathcal{B}_u$后，会通过下式计算BPR Loss：

$$L_{BPR,u} = \frac{1}{|\mathcal{B}_u|} \sum_{(i,i') \in \mathcal{B}_u} \left[-\ln \sigma\left(\hat{r}_{ui} - \hat{r}_{ui'}\right) + \alpha(\|e_u\|^2 + \|e_i\|^2 + \|e_{i'}\|^2)\right] \tag{3}$$

计算完损失后，对其进行反向传播，得到本地用户嵌入的梯度g_u和物品（包括交互过的物品和负采样的物品）嵌入的梯度 $\left\{g_i^u \,\middle|\, i \in \mathcal{I}_u \cup \widetilde{\mathcal{I}_u}\right\}$。客户端通过下式来更新本地用户嵌入：

$$e_u = e_u - \gamma g_u \tag{4}$$

其中，γ为学习率。

而梯度$\left\{g_i^u \,\middle|\, i \in \mathcal{I}_u \cup \widetilde{\mathcal{I}_u}\right\}$和梯度对应的下标$\left\{i \,\middle|\, i \in \mathcal{I}_u \cup \widetilde{\mathcal{I}_u}\right\}$则需要上传给中心服务器。需要注意的是，尽管这些梯度不是用户原始的隐私数据，其隐含的敏感信息大大减少，但还是存在一定的泄露隐私的可能 [30]，即中心服务器可以通过这些梯度信息反推用户的原始信息如交互物品列表等。因此，在上传这些梯度之前，客户端会对其进行本地差分隐私 [20]操作，进一步降低隐私泄露的风险，如下所示：

$$g_i^u = clip\left(g_i^u, \delta\right) + Laplace\left(0, \lambda\right) \tag{5}$$

其中$clip\left(g_i^u, \delta\right)$表示将$g_i^u$限制在$[-\delta, \delta]$的范围内，Laplace$(0, \lambda)$表示均值为0、强度为$\lambda$的拉普拉斯噪声。另外，由于我们选择了没有额外模型参数（如权重矩阵等）的LightGCN作为骨干模型，因此客户端不需要像FedPerGNN那样上传模型梯度，这在一定程度上降低了和服务器之间的通信代价。

中心服务器收到参与本次训练的所有客户端上传的梯度及其下标后，使用FedAvg [17]对这些梯度进行聚合，并更新相应的物品嵌入：

$$
\begin{aligned}
g_i &= \frac{\sum_{u \in \bar{\mathcal{U}}} g_i^u}{|\bar{\mathcal{U}}|} \\
e_i &= e_i - \gamma g_i
\end{aligned}
\tag{6}
$$

最后，服务器将更新后的物品嵌入发送给所有客户端，并开始新一轮的模型训练。

3.3 保护隐私的图扩展方法

基于图神经网络的集中式推荐算法将用户对物品的交互数据储存在中心服务器，因此可以轻易地构建出“用户——物品”的全局图。然而，当我们需要保护用户的交互数据，让它们只存储在用户本地而非中心服务器时，中心服务器会因为缺少用户数据而无法构建全局图，用户也只能根据本地存储的交互数据构建一阶局部子图。这样的一阶局部子图所蕴含的信息较少，不利于模型训练。

因此，我们参考FedPerGNN中保护隐私的图扩展方法 [26]，如图3所示，将客户端中的一阶局部子图扩展成二阶局部子图，以增加图中所蕴含的信息。接下来，我们将介绍该方法的相关细节。

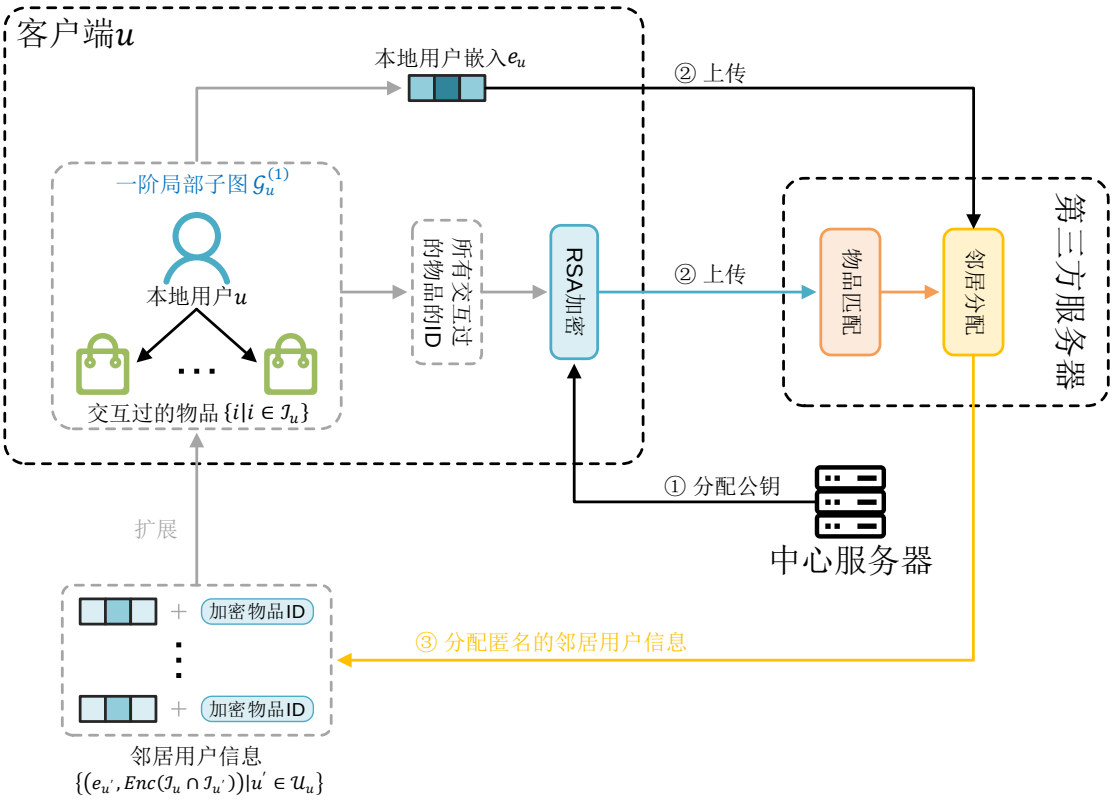


图 3. FedLightGCN的总体框架

在开始之前，我们需要先引入一个第三方服务器，并假设它和负责模型训练的中心服务器互不知晓、互不干扰。接下来，中心服务器首先生成一个公钥，并将它分配给所有客户端。每个客户端u通过该公钥对自己所有的交互物品ID$\{i \,|i \in \mathcal{I}_u\}$进行RSA加密，然后将加密后的交互物品ID $\{Enc\,(i)\,|i \in \mathcal{I}_u\}$和自己的本地用户嵌入$e_u$发送给第三方服务器。

第三方服务器收到从客户端发送过来的加密物品ID和用户嵌入后，可以通过匹配物品ID密文来找到对同一物品进行过交互的用户（他们互为对方的邻居用户），并为每位用户u分配他们的邻居用户信息，即邻居用户嵌入和共同交互过的物品的ID密文$\{(e_{u'}, Enc\,(\mathcal{I}_u \cap \mathcal{I}_{u'}))\,|u' \in \mathcal{U}_u\}$。

用户收到自己的邻居用户信息后，便可以将原先的一阶局部子图扩展为二阶局部子图了。如此一来，客户端中的图的信息便增多了，更有利于模型的训练。

需要注意的是，客户端中的邻居用户嵌入是静态、固定的，因此我们需要每隔一定周期进行一次所有客户端的图扩展。这里我们遵循FedPerGNN原论文 [26]中的设置，每隔一个epoch，即$\left\lfloor \frac{n}{|\mathcal{U}|} \right\rfloor$次训练后，进行一次所有客户端的图扩展。

3.4 LightGCN

在这一节中，我们将介绍LightGCN的基本原理，以及原始的LightGCN和FedLightGCN中的骨干模型LightGCN的区别。

假设一个K层的LightGCN，其输入是某个图$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$的拓扑结构（具体来说是邻接矩阵和度矩阵）和节点相应的嵌入 $\{e_v\,|v \in \mathcal{V}\}$。它会对图中所有节点进行$K$次图卷积操作，让节点聚合其$K$阶邻域的信息。具体到图中的某个节点$v$，其第$k+1$层的图卷积公式如下：

$$e_v^{(k+1)} = \sum_{v' \in \mathcal{N}_v} \frac{1}{\sqrt{|\mathcal{N}_v|}\sqrt{|\mathcal{N}_{v'}|}} e_{v'}^{(k)} \tag{7}$$

其中，$e_v^{(k+1)}$是节点v经过第$k+1$层图卷积后得到的嵌入，$e_v^{(0)} = e_v$；$\mathcal{N}_v$是和节点v相连的节点的集合。

在得到所有层的节点嵌入后，我们对其进行加权平均来得到最终的节点潜在嵌入：

$$h_v = \sum_{k=0}^{K} \beta_k e_v^{(k)} \tag{8}$$

这里我们遵循LightGCN原始论文 [6]中的设置，将β_k统一设置为$1/\,(K+1)$。

为了提高计算效率，我们可以将上述公式写成矩阵形式来实现计算的并行化：

$$\begin{aligned} E^{(k+1)} &= \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right) E^{(k)} \\ H &= \beta_0 E^{(0)} + \beta_1 E^{(1)} + \ldots + \beta_K E^{(K)} \end{aligned} \tag{9}$$

其中，$E^{(k)} \in \mathbb{R}^{|\mathcal{V}| \times d}$是第$k$层图卷积得到的节点嵌入矩阵；$A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$是邻接矩阵；$D \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$是度矩阵；$H \in \mathbb{R}^{|\mathcal{V}| \times d}$是节点潜在嵌入矩阵。

原始的LightGCN和FedLightGCN中的LightGCN的区别在于，前者的输入是包含所有用户节点和物品节点的全局图，而后者的输入是客户端u构建的二阶局部子图 $\mathcal{G}_u^{(2)} = \left\{\mathcal{V}_u^{(2)} = \{u\} \cup \mathcal{I}_u \cup \mathcal{U}_u, \mathcal{E}_u^{(2)}\right\}$。

3.5 隐私保护分析

在这一节中，我们将分析FedLightGCN对用户隐私的保护情况，即在整个训练过程中，用户u的交互物品列表$\mathcal{I}_u$是否会泄露给服务器和其他用户。

用户隐私的泄露只有可能发生在客户端和服务器进行数据交互的时候，具体到本文，即：（1）客户端上传物品嵌入梯度和梯度对应的下标给中心服务器；（2）客户端上传用户嵌入和加密的交互物品ID给第三方服务器；（3）第三方服务器发送邻居用户信息给客户端。

对于第1处，首先客户端发送给中心服务器的物品嵌入梯度对应的下标中，包含了交互过的物品和负采样的物品，因此中心服务器无法从这里推断出客户端的交互物品列表。其次，虽然中心服务器存在通过物品嵌入梯度反推用户的交互物品列表的可能性 [30]，但这个可能性很小。这是因为FedLightGCN中的物品嵌入梯度计算较为复杂，中心服务器难以模拟反推，而且客户端上传梯度之前还进行了本地差分隐私操作，进一步提高了中心服务器反推原始信息的难度。

对于第2处，客户端发送给第三方服务器的交互物品ID是经过加密的，而第三方服务器并没有相应的私钥，无法解密得到用户的交互物品列表，因此隐私也没有泄露。

对于第3处，第三方服务器发送给客户端的邻居用户信息不包含邻居用户的ID，信息是匿名的，客户端不知道得到的加密交互物品列表属于哪个用户，因此隐私也没有泄露。

综上所述，FedLightGCN很好地保护了用户的隐私。

4 复现细节

4.1 与已有开源代码对比

本文复现的文章FedPerGNN虽然有TensorFlow版开源代码[1]，但本人没有参考该开源代码，而是根据自己对文章的理解，用PyTorch重新编写了相关代码。

4.2 创新点

本文与复现原文相比，创新点如下：

- 研究问题不同。原文研究的是评分预测问题，较为陈旧，而本文研究的则是物品排序问题，更具代表性同时更贴近当前本领域的研究趋势。
- 所用backbone不同。原文使用的backbone为GAT，无法很好地适配推荐系统中的物品排序任务，且额外的模型参数导致了通信代价的增加。而本文更换backbone为当前更轻量和性能更好LightGCN，以此提高推荐精度和降低通信代价。
- 数据集不同。本文选取了物品排序任务中较常使用的2个数据集，而不是原文所用的数据集。

[1] https://github.com/wuch15/FedPerGNN

5 实验结果分析

5.1 实验设置

5.1.1 数据集选择

在本文的实验中，我们选择了两个知名的公开数据集Gowalla和Yelp2018，它们被广泛地应用于单类协同过滤问题上。其中，Gowalla数据集是从一个社交网站Gowalla上收集得到的，它包含了用户通过签到来进行分享的几百万个地理位置信息。而Yelp2018数据集则是来自2018年的Yelp挑战赛，其中本地商业场所如餐厅、酒吧等被视为推荐系统中的物品。

为了方便对比，我们不使用上述数据集的原始版本，而是使用LightGCN原始论文处理过的版本，它删去了上述两个原始数据集中交互少于10次的用户和物品，以此保证数据集的质量。这两个数据集的统计信息如表1所示：

表 1. 实验数据集的统计信息

数据集	用户数量	物品数量	交互次数	稠密度
Gowalla	29,858	40,981	1,027,370	0.00084
Yelp2018	31,668	38,048	1,561,406	0.00130

5.1.2 评价指标选择

在评价指标方面，我们选择了在物品排序问题中常用的两个指标——召回率$Recall$和归一化折损累计增益$NDCG$，它们的计算公式如下：

$$\begin{aligned} Recall@topK &= \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{|\mathcal{I}_u^{re}\cap\mathcal{I}_u^{te}|}{|\mathcal{I}_u^{te}|} \\ NDCG@topK &= \frac{1}{|\mathcal{U}^{te}|}\sum_{u\in\mathcal{U}^{te}}\frac{\sum_{k=1}^{topK}\frac{I\left(\mathcal{I}_u^{re}(k)\in\mathcal{I}_u^{te}\right)}{\log(k+1)}}{\sum_{k=1}^{topK}\frac{1}{\log(k+1)}} \end{aligned} \tag{10}$$

其中，$\mathcal{U}^{te}$是测试集中的用户集合；$\mathcal{I}_u^{te}$是测试集中用户u交互过的物品的集合；$I(\cdot)$是指示函数，当且仅当x为真时，$I(x)=1$；否则，$I(x)=0$；$\mathcal{I}_u^{re}(k)$是推荐物品列表$\mathcal{I}_u^{re}$中的第k个物品。

$Recall$衡量了推荐物品列表中的物品出现在测试集的比例，而$NDCG$则根据推荐物品在$\mathcal{I}_u^{re}$中的排位来区分它们的贡献。这两个评价指标越大，代表算法的推荐性能越好。

5.1.3 基准算法选择

为了能够直观地看出FedLightGCN推荐性能的好坏，我们需要挑选一些推荐算法与其进行对比，我们把这些用于对比的算法称作基准算法。我们分别选择了集中式推荐算法和联邦推荐算法中具有代表性的部分算法作为基准算法。

集中式推荐算法中，我们选择了Mult-VAE [11]和LightGCN [6]。Mult-VAE是一个基于变分自编码器的协同过滤算法，在多个数据集上都取得了比基于矩阵分解的推荐算

法更好的结果。LightGCN则是我们在上面已提及多次的基于图神经网络的推荐算法，也是FedLightGCN的骨干模型。

而在联邦推荐算法方面，我们选择了FCF [1]、FedMF [3]和FedeRank [2]。三者都是基于矩阵分解的联邦推荐算法，但FCF和FedMF采用了MSE Loss，而FedeRank采用了BPR Loss。

5.1.4 模型参数设置

对于FedLightGCN，我们将嵌入的维度d设置为64，并使用Xavier方法 [4]初始化嵌入，骨干模型的层数K设置为2层，每轮训练选取的客户端数量$\left|\bar{\mathcal{U}}\right|$设置为512，客户端的负采样数量$\left|\widetilde{\mathcal{I}}_u\right|$设置为2048，推荐物品列表的长度$topK$设置为20，并训练1000个epoch。训练时的优化器选择Adam优化器 [7]，学习率γ设置为0.001。对于L_2正则化系数α、本地差分隐私中的梯度裁剪阈值δ、拉普拉斯噪声强度λ，在Gowalla数据集中我们设置为0.001、5×10^{-4}和10^{-5}，在Yelp2018数据集中则设置为0.01、3×10^{-3}和10^{-4}。早停和验证策略均遵循LightGCN原论文 [6]中的设置。

对于基准算法，我们遵循这篇论文 [10]中的设置。在接下来的实验中，除非有特别说明，否则都按照本节内容进行参数设置。

5.2 性能评估

FedLightGCN与基准算法的性能对比结果如表2所示，其中LightGCN-K表示K层的LightGCN。

表 2. FedLightGCN与基准算法的性能对比

数据集	Gowalla		Yelp2018	
算法	Recall@20	NDCG@20	Recall@20	NDCG@20
Mult-VAE	0.1641	0.1335	0.0584	0.0450
LightGCN-1	0.1755	0.1492	0.0631	0.0515
LightGCN-2	0.1777	0.1524	0.0622	0.0504
LightGCN-3	**0.1823**	**0.1555**	**0.0639**	**0.0525**
FCF	0.0703	0.0588	0.0282	0.0235
FedMF	0.0727	0.0583	0.0250	0.0207
FedeRank	0.1440	0.1164	0.0503	0.0405
FedLightGCN	**0.1661**	**0.1390**	**0.0595**	**0.0485**

由上表可以看出，在与集中式推荐算法的对比中，FedLightGCN的推荐性能略好于Mult-VAE，但与LightGCN则有一定的差距。即使是1层的LightGCN，它的推荐性能仍然要优于FedLightGCN。

这种性能损失是合理和可接受的，也是大多数联邦推荐算法的其中一个问题，因为对传统推荐算法进行联邦化时往往需要舍弃部分信息、加入噪声等来保护用户隐私，导致算法性能下降。具体到本文，FedLightGCN对比原始的LightGCN存在如下局限：（1）

FedLightGCN和LightGCN-1/2在前向传播方面是等价的，但在进行反向传播时，邻居用户嵌入的梯度由于邻居匿名而无法传递给相应客户端进行更新，导致只更新了本地用户嵌入，嵌入更新不完全；（2）FedLightGCN为了保护用户隐私，在上传物品嵌入梯度的时候进行了梯度裁剪和加噪；（3）FedLightGCN只能构建二阶局部子图，这在一定程度上限制了骨干模型的层数。以上局限都导致了FedLightGCN的性能损失。

而对比联邦推荐算法FCF、FedMF和FedeRank，FedLightGCN的推荐性能则明显更加优秀。毕竟前者都是基于矩阵分解的联邦推荐算法，无法充分利用“用户——物品”的交互信息，而FedLightGCN融合了图神经网络，能够更加充分地利用隐式信息。

5.3 消融实验

在这一节中，我们进行了一个消融实验，来验证FedLightGCN中的“保护隐私的图扩展方法”这一组件的有效性，实验结果如下：

表 3. 消融实验结果

图扩展	Gowalla		Yelp2018	
	Recall@20	NDCG@20	Recall@20	NDCG@20
有	**0.1661**	0.1390	**0.0595**	**0.0485**
无	0.1638	**0.1402**	0.0582	0.0475

由上表可以看出，该图扩展方法确实可以在一定程度上提升FedLightGCN的推荐性能，并且提升幅度也和LightGCN-1到LightGCN-2的提升幅度大致相当，实验结果在合理的范围内。

5.4 超参数实验

联邦推荐算法的推荐性能、隐私保护和通信代价这三者往往是我们关注的重点，因此在本节中，我们将进行一系列的超参数实验，来探究FedLightGCN的推荐性能、隐私保护和通信代价这三者之间的关系。

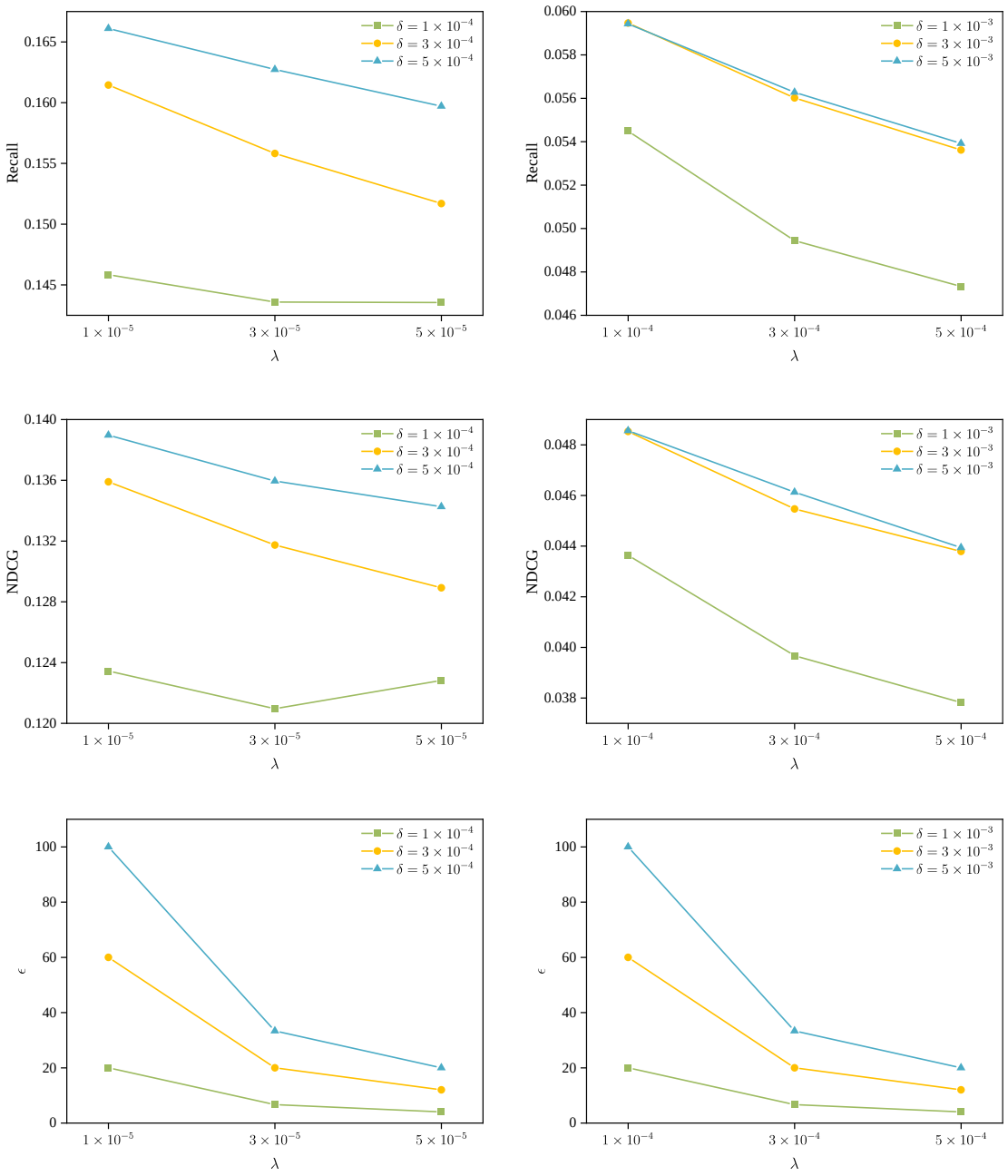
5.4.1 隐私保护和推荐性能之间的关系

从3.5节中我们对FedLightGCN的隐私保护分析可知，中心服务器如果想要得到用户隐私，只有通过客户端上传的物品嵌入梯度反推原始信息这一种方式，因此我们会在上传梯度之前对其进行本地差分隐私操作，来进一步保护隐私。其中，隐私预算ϵ可以用来衡量本地差分隐私的隐私保护程度，其公式 [20]如下：

$$\epsilon = \frac{2\delta}{\lambda} \tag{11}$$

ϵ的值越小，代表隐私保护程度越高，即较小的梯度裁剪阈值δ和较大的拉普拉斯噪声强度λ更有利于保护用户的隐私。

为了探究FedLightGCN的隐私保护和推荐性能之间的关系，我们通过改变FedLightGCN中δ和λ的取值，进行了一系列实验，实验结果如下：



（a）Gowalla　　（b）Yelp2018

图 4. 隐私保护和推荐性能的实验结果

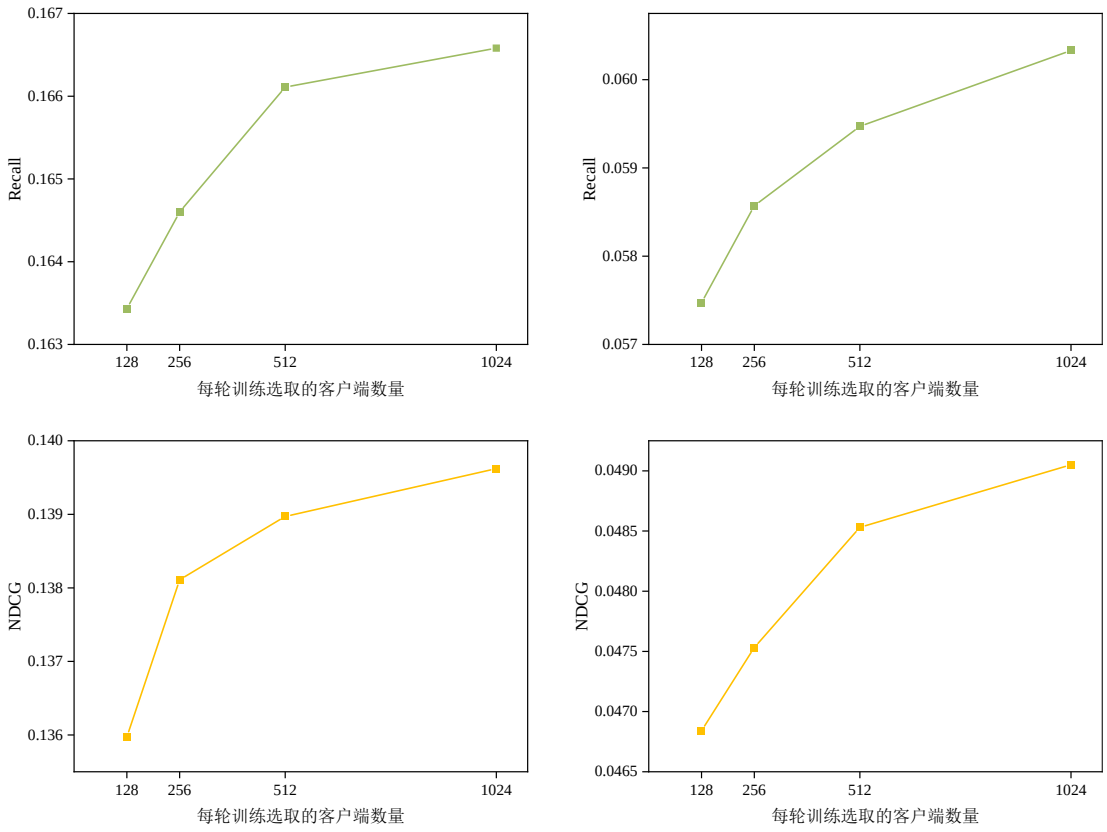
观察图4（a）可知，在Gowalla数据集的实验结果中，隐私保护和推荐性能大致呈负相关，即梯度裁剪阈值δ越小，拉普拉斯噪声强度 λ越大，则隐私预算ϵ越小，隐私保护程度越高，但推荐性能也随之下降。Yelp2018数据集的实验结果与之类似，此处不再赘述。

由于FedLightGCN的隐私保护和推荐性能呈负相关，因此我们要合理地选择本地差分隐私中的参数δ和λ，不应过大或过小，要平衡好算法的隐私保护程度和推荐性能。

5.4.2 通信代价和推荐性能之间的关系

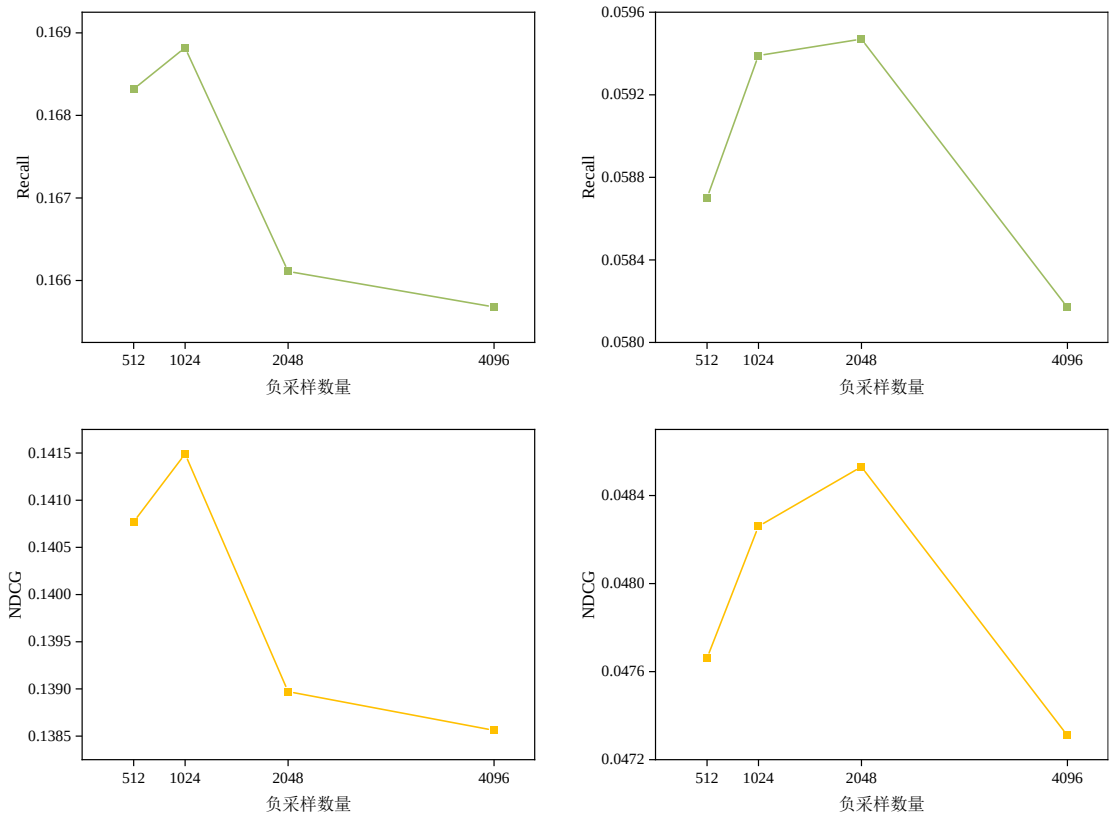
FedLightGCN中的通信代价主要来源于训练时客户端和中心服务器之间的数据交互，每轮训练选取的客户端数量 $|\bar{\mathcal{U}}|$越多，负采样数量$|\tilde{\mathcal{I}}_u|$越大，则通信代价越大。

我们通过改变$|\bar{\mathcal{U}}|$和$|\tilde{\mathcal{I}}_u|$的取值，进行了一系列实验来探究FedLightGCN的通信代价和推荐性能之间的关系，实验结果如下：



（a）Gowalla　　（b）Yelp2018

图 5. 选取的客户端数量和推荐性能的实验结果



（a）Gowalla　　（b）Yelp2018

图 6. 负采样数量和推荐性能的实验结果

观察图5（a）和图6（a）可知，在Gowalla数据集的实验结果中，每轮训练选取的客户端数量越大，推荐性能往往越好；而负采样数量越大，推荐性能往往会先提高后下降。Yelp2018数据集的实验结果与之类似。因此我们要合理地选择这两个参数，既要保证推荐性能，又要防止通信代价过大。

6　总结与展望

6.1　总结

本文首先介绍了联邦推荐算法的研究背景和现状，随后引出了该研究领域的相关工作，包括图学习和联邦学习。接下来，本文围绕基于矩阵分解的早期联邦推荐算法性能欠佳的问题，复现了一个基于图神经网络的联邦推荐算法框架FedPerGNN。在此基础上，将它和一个性能优异的基于图神经网络的集中式推荐算法LightGCN相结合，进而提出了一个基于图学习的跨用户联邦推荐算法FedLightGCN，并将之应用到单类协同过滤问题上。在成功设计算法后，本文详细介绍了FedLightGCN及其内部组件，并进行了一系列实验来探究FedLightGCN的性能和特性。

最后，通过理论分析和实验证明，我们提出的FedLightGCN具有如下几个特点和贡献：（1）FedLightGCN的推荐性能普遍优于基于矩阵分解的联邦推荐算法，证明了基于图学习的联邦推荐算法的有效性；（2）FedLightGCN的通信代价较FedPerGNN有一定程度的下降，更有利于实际部署；（3）FedLightGCN能够较好地保护用户的隐私数据，符合用户意愿和相

关法律法规。FedLightGCN的推荐性能往往和隐私保护、通信代价相互制约，因此我们在选择超参数时应当平衡好这三者之间的关系。

6.2 展望

尽管FedLightGCN在推荐性能、隐私保护和通信代价这三方面都取得了一定程度的进步，但仍然存在不少局限：（1）它没有解决邻居用户嵌入梯度无法回传、本地差分隐私引入噪声和无法构建全局图等问题，这在一定程度上限制了它的性能；（2）如果中心服务器和第三方服务器相互知晓并勾结，则用户的隐私数据会被泄露。

因此在未来的研究中，我们会尝试使用损失更低的加密方式、更巧妙安全的图扩展方法等来解决上述问题，实现一个性能更强、安全性更高的基于图学习的联邦推荐算法。

参考文献

[1] Muhammad Ammad-Ud-Din, Elena Ivannikova, Suleiman A Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*, 2019.

[2] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Antonio Ferrara, and Fedelucio Narducci. Federank: User controlled feedback with federated recommender systems. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 32–47. Springer, 2021.

[3] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *IEEE Intelligent Systems*, 36(5):11–20, 2020.

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[6] 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.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[10] Junyi Li and Heng Huang. Fedgrec: Federated graph recommender system with lazy update of latent embeddings. *arXiv preprint arXiv:2210.13686*, 2022.

[11] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[12] Feng Liang, Weike Pan, and Zhong Ming. Fedrec++: Lossless federated recommendation with explicit feedback. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4224–4231, 2021.

[13] Feng Liang, Enyue Yang, Weike Pan, Qiang Yang, and Zhong Ming. Survey of recommender systems based on federated learning. *Scientia Sinica Informationis*, 52(5):713–741, 2022.

[14] Guanyu Lin, Feng Liang, Weike Pan, and Zhong Ming. Fedrec: Federated recommendation with explicit feedback. *IEEE Intelligent Systems*, 36(5):21–30, 2020.

[15] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S Yu. Federated social recommendation with graph neural network. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–24, 2022.

[16] Sichun Luo, Yuanzhang Xiao, and Linqi Song. Personalized federated recommendation via joint representation learning, user clustering, and model adaptation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4289–4293, 2022.

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[18] Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. *Advances in neural information processing systems*, 20, 2007.

[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[20] Tao Qi, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. Privacy-preserving news recommendation model learning. *arXiv preprint arXiv:2003.09592*, 2020.

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[22] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[23] Alvin Toffler. *Future shock*. Bantam, 1984.

[24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[25] 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.

[26] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Tao Qi, Yongfeng Huang, and Xing Xie. A federated graph neural network framework for privacy-preserving personalization. *Nature Communications*, 13(1):3091, 2022.

[27] Enyue Yang, Yunfeng Huang, Feng Liang, Weike Pan, and Zhong Ming. Fcmf: Federated collective matrix factorization for heterogeneous collaborative filtering. *Knowledge-Based Systems*, 220:106946, 2021.

[28] Liu Yang, Ben Tan, Vincent W Zheng, Kai Chen, and Qiang Yang. Federated recommendation systems. *Federated Learning: Privacy and Incentive*, pages 225–239, 2020.

[29] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[30] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

[31] 张洪磊, 李浥东, 邬俊, 陈乃月, and 董海荣. 基于隐私保护的联邦推荐算法综述. 自动化学报, 48(9):2142–2163, 2022.

[32] 徐冰冰, 岑科廷, 黄俊杰, 沈华伟, and 程学旗. 图卷积神经网络综述. 计算机学报, 43(5):755–780, 2020.

[33] 李改 and 李磊. 基于矩阵分解的协同过滤算法. 计算机工程与应用, 47(30):4–7, 2011.

[34] 汤凌韬, 陈左宁, 张鲁飞, and 吴东. 联邦学习中的隐私问题研究进展. 软件学报, 34(1):197–229, 2021.