

# 基于二部图对比学习的特征增强推荐算法

余鹏, 杨佳琦, 陈欣然, 贺超波\*

(华南师范大学计算机学院, 广东 广州 510631)

**摘要:** 信息过载已成为大数据时代面临的一个普遍问题, 推荐算法是解决该问题的有效手段。现有的推荐算法具有不同程度的有效性, 但仍面临着如何学习更高质量的项目和用户特征以提升推荐性能的挑战。提出一种基于二部图对比学习的特征增强推荐算法 FRBGCL。设计一个项目特征初始化模块, 利用图卷积网络(GCN)进行各类项目关系二部图的表示学习, 并使用基于注意力机制的特征融合策略获取项目初始特征。此外, 在构建用户-项目二部图的基础上, 设计图对比学习模块进一步增强项目 and 用户特征, 进而提升推荐算法性能。在 XuetangX、Last.fm 和 Yelp2018 3 个数据集上的实验结果表明, 在选择最优参数的情况下, FRBGCL 的 Top20 推荐结果与次优算法相比, 召回率分别提升 2.1%、6.8%、11.6%, 归一化折损累计增益(NDCG)分别提升 1.8%、6.1%、13.1%, 命中率(HR)分别提升 1.7%、7.8%、8.4%。

**关键词:** 深度学习; 推荐算法; 二部图; 对比学习; 图卷积网络

中图分类号: TP391.3

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0069099

## Feature-enhanced Recommendation Algorithm Based on Bipartite Graph Contrastive Learning

YU Peng, YANG Jiaqi, CHEN Xinran, HE Chaobo\*

(School of Computer Science, South China Normal University, Guangzhou 510631, Guangdong, China)

**【Abstract】** Recommendation algorithms are effective in addressing information overload, a common problem in the era of big data. Existing recommendation algorithms have different degrees of effectiveness but still face the challenge of learning higher quality items and user features to enhance recommendation performance. Therefore, this paper proposes a Feature-enhanced Recommendation algorithm based on Bipartite Graph Contrastive Learning (FRBGCL). An item feature initialization module is designed that can use Graph Convolutional Network (GCN) for the representation learning of bipartite graphs of all types of item relationships, and an attention mechanism-based feature fusion strategy is adopted to obtain the initial features of items. In addition, a graph Contrastive Learning (CL) module is designed based on the construction of user-item bipartite graphs, which can further enhance item and user features, leading to an improvement in recommendation performance. On three datasets, XuetangX, Last.fm, and Yelp2018, compared with the suboptimal algorithm, FRBGCL improves the Top20 recommendation results by 2.1%, 6.8%, and 11.6% for recall; 1.8%, 6.1%, and 13.1% for Normalized Discounted Cumulative Gain (NDCG); and 1.7%, 7.8%, and 8.4% for Hit Rate (HR), with optimal parameter selection.

**【Key words】** deep learning; recommendation algorithm; bipartite graph; Contrastive Learning (CL); Graph Convolution Network (GCN)

## 0 引言

推荐算法在商品推荐<sup>[1]</sup>、新闻推荐<sup>[2]</sup>和社交媒体推荐<sup>[3]</sup>等领域有着广泛应用, 作为一种信息过滤技术, 它根据用户的个人偏好和交互历史, 自动地推荐给目标用户其可能感兴趣的物品、内容和服务等<sup>[4]</sup>。在过去的几十年中, 推荐算法得到了广泛研究, 并且大量推荐算法被提出, 如基于用户的协同过滤<sup>[5]</sup>、基于卷积神经网络<sup>[6]</sup>和基于图的

推荐算法等。基于图的推荐算法将用户和项目数据建模成各种图结构, 如异质图<sup>[7-8]</sup>、知识图谱<sup>[9-10]</sup>、二部图<sup>[11]</sup>等, 通过学习用户或项目之间的相似性进行推荐。总体而言, 基于图的推荐算法能够有效地捕捉用户和物品之间的潜在关联, 提高推荐的准确性, 因此, 研究者提出了越来越多的基于图的推荐算法。其中, 由于二部图更适合建模用户与项目之间的交互关系并且可扩展性强, 因此基于二部图的推荐算法受到更多关注。

收稿日期: 2023-12-26 修回日期: 2024-02-27

基金项目: 国家自然科学基金面上项目(62077045); 广东省自然科学基金(2019A1515011292)。

通信作者 E-mail: \* hechaobo@foxmail.com

二部图将用户和物品划分成 2 个不同的集合,图中的边只连接 2 个不同集合中的节点,而不连接同一集合中的节点。目前基于二部图的推荐算法主要是利用图卷积网络(GCN)来实现。WANG 等<sup>[11]</sup>提出了 NGCF 算法,通过使用 GCN 来聚集邻域节点的信息用于推荐任务。HE 等<sup>[12]</sup>则是在 NGCF 的基础上证明了特征变换和非线性激活 2 个部分的无效和冗余,基于此提出了 LightGCN,通过将节点的最终嵌入构造为节点在所有层上的嵌入的加权和,有效控制过拟合。此外,MA 等<sup>[13]</sup>利用注意力机制来学习二部图中不同邻居的不同权重,从而有效聚合节点邻居的特征。

虽然上述基于二部图的推荐算法具有较好的推荐性能,但仅考虑了项目和用户之间稀疏的直接交互关系,无法很好地捕捉到用户的个性化兴趣或项目的独有特征;数据源中隐含额外项目交互关系数据,如项目与项目类型等,其中可能包含了项目数据之间的隐含关系;用户的社交信息中包含用户的兴趣信息,忽略这些信息会导致学习到的用户和项目特征不够准确且不具有区分性,影响了推荐算法的性能。此外,利用多源信息进行训练,通常能够更好地捕捉用户和项目之间的复杂关系,提升推荐效果。

为解决上述问题,本文提出了一种基于二部图对比学习(CL)的特征增强推荐算法(FRBGCL)。本文主要工作如下:

1)为了缓解数据稀疏问题,从数据源中提取出隐含的项目交互关系数据用于获取增强的项目特征。设计一个项目特征初始化模块,通过结合 GCN 和注意力机制用于获取融合了项目交互关系的增强项目特征,并将得到的项目特征作为推荐任务中的初始项目特征。

2)在推荐任务中引入对比学习,使用视图间的对比学习和视图内的对比学习作为辅助任务来获得更具鉴别力的用户和项目特征,并通过联合对比损失和推荐损失对模型进行优化。

3)在 3 个不同类型的真实数据集上进行了大量实验。实验结果表明,FRBGCL 优于目前的代表性方法,在 Recall@K、NDCG@K 和 HR@K 等指标上均有明显的提升。

## 1 相关工作

近年来,随着图深度学习研究的流行,越来越多的科研人员关注基于图的算法。目前大多数关于图深度学习的研究都集中在监督学习场景并且

严重依赖标签,因此存在过于依赖监督学习情况导致算法泛化性较差以及手动标签的收集和注释成本高昂等问题。由于对比学习能够缓解数据稀疏等问题<sup>[14]</sup>,使算法从未标记的数据中学习更多特征表示,实现更好的性能和提高算法的鲁棒性,因此大量研究将对比学习引入各类图学习的任务中<sup>[15]</sup>。

在图表示学习方面,GCC<sup>[16]</sup>利用随机游走作为增强来提取节点的上下文信息,然后使用 MoCo<sup>[17]</sup>对比框架将节点的表示与其对应的表示进行对比。对于属性图,GraphSAGE<sup>[18]</sup>将相关方法扩展到属性图,并提出了一种新的利用图神经网络来计算节点嵌入的方法,它同样也使用了基于随机游走的方法作为其内部采样策略。对于异质图表示学习,SELR<sup>[19]</sup>通过对元路径进行采样以捕获上下文信息,它由一个主链接预测任务和几个元路径预测辅助任务组成,以强制同一元路径内的节点共享更紧密的语义信息。此外,HeCo<sup>[20]</sup>则是在异构图上进行对比,从网络模式和元路径 2 个角度生成 2 个对比视图,编码器通过最大化同一节点的嵌入之间的表征来进行训练。

最近,越来越多的研究者将对比学习引入基于二部图的推荐算法中。WU 等<sup>[21]</sup>提出将对比学习用于二部图结构并实现了 SGL 方法,利用辅助自监督任务的自辨别特性来加强节点表示学习,通过使用 3 种操作算子来获取不同的视图,提高算法的准确性和鲁棒性。YU 等<sup>[22]</sup>在前人的基础上提出了一种将均匀噪声添加到嵌入空间以创建对比视图的推荐算法 SimGCL,通过学习更统一的用户或项目表示来进行推荐,从而隐式地缓解流行度偏差和平滑地调整学习表示的均匀性。XSimGCL<sup>[23]</sup>是在 SimGCL 的基础上提出的,它通过向学习到的用户和项目嵌入添加均匀噪声来实现更有效的数据增强。与 SimGCL 不同的是,XSimGCL 在最终层对比时引入了跨层对比度,可以以更简单的架构获得优于 SimGCL 的性能。JIANG 等<sup>[24]</sup>提出自适应图对比学习算法 AdaGCL,该算法使用 2 个自适应对比视图生成器进行数据增强,具体是使用 2 个可训练的视图生成器来创建自适应对比视图。通过 2 种自适应对比视图,AdaGCL 将额外的高质量训练信号引入,从而有助于缓解数据稀疏和噪声问题。

综上所述,对比学习在图深度学习研究中得到了大量应用,主要是通过数据表示之间的相似性和差异性学习到有用的特征表示。已有的基于

二部图对比学习的推荐算法主要是通过对用户和项目的交互进行操作,而忽略了其他的隐含交互关系,因此相关算法的推荐性能仍有进一步的提升空间。基于此,FRBGCL 将结合对比学习的优势,利用项目的额外交互关系来更准确地进行推荐。

## 2 FRBGCL 算法设计

### 2.1 FRBGCL 框架

本文提出了一种结合对比学习和 GCN 的特征增强推荐算法 FRBGCL,图 1 展示了算法的整体框架(彩色效果见《计算机工程》官网 HTML 版,下同)。

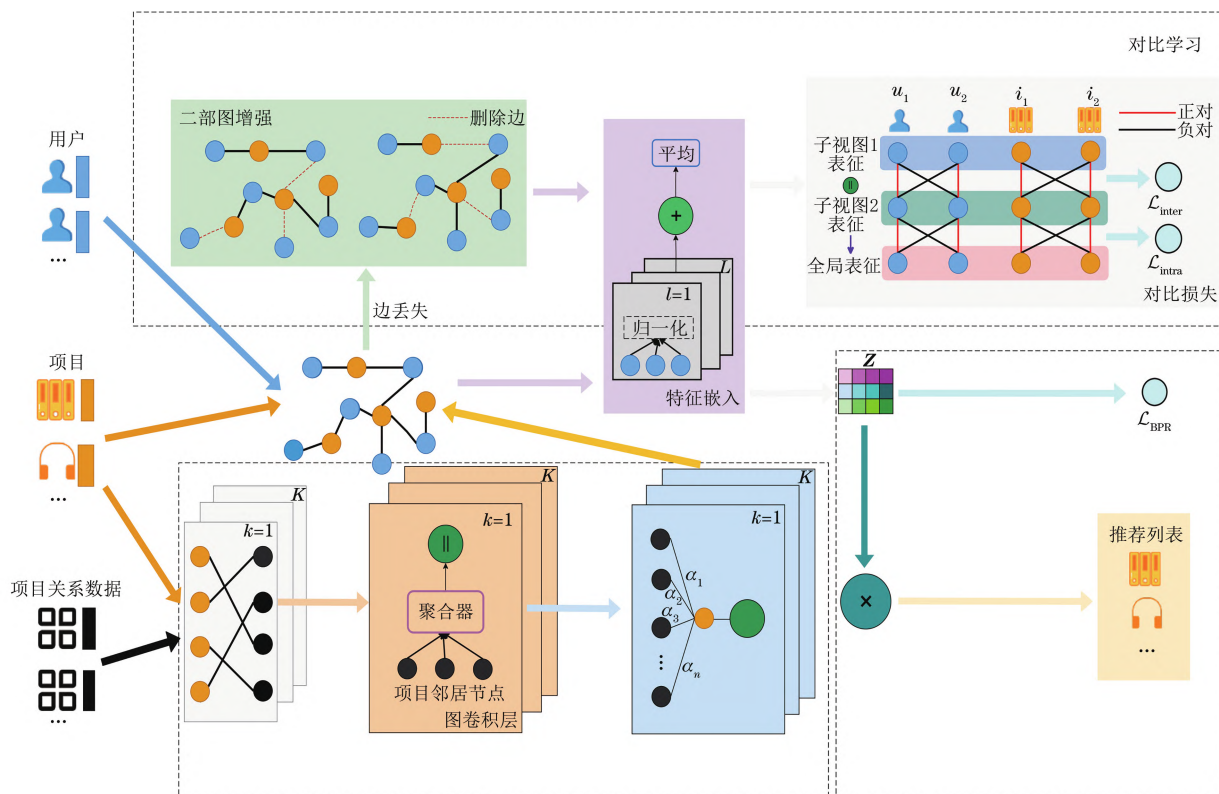


图 1 FRBGCL 模型框架

Fig.1 Framework of FRBGCL model

FRBGCL 模型中的主要模块包括:

1)项目特征初始化模块:首先通过 GCN 来获得融合了项目外部交互关系的项目特征,然后利用改进的图注意力机制来自适应地平衡各类项目特征信息的贡献。

2)特征嵌入模块:利用 GCN 完成用户和项目的特征聚合与嵌入操作,得到最终的用户特征和项目特征。

3)对比学习模块:通过边丢失算子构建子视图进行二部图增强操作,并有意地减少高阶节点的影响来减轻度偏差,同时构建视图间和视图内部 2 种损失函数进行对比学习。

4)项目推荐模块:通过上述步骤得到的用户与项目特征完成内积操作,获取最终推荐给目标用户的项目列表。

### 2.2 项目特征初始化

本文将构造一个项目特征初始化模块,通过从数据源中提取出的项目和项目外部交互关系构建项

目关系的稀疏矩阵来建模二部图。该模块首先通过图卷积层进行特征融合操作,结合节点周围所有的邻居特征和当前节点特征的均值进行聚合,得到融合后的项目特征。其次通过图注意力机制来处理项目和其交互数据的特征,主要作用是邻域中的项目和项目交互数据指定不同的权值,通过学习节点之间的关系权重得到增强的项目特征表示。

在图卷积层中,通过对项目特征和项目外部交互数据特征进行融合操作,获取融合了项目外部交互数据的项目特征。具体来说,图卷积层利用了 GCN 来进行特征融合操作,GCN 选择对应的聚合方式后,利用节点的邻居特征得到融合后的项目特征。在推荐任务中,激活函数具有负面影响<sup>[12]</sup>,因此,本文算法中图卷积层不使用激活函数。给定项目集合  $I$  和项目外部交互数据集  $T$ ,首先将项目特征和其外部交互数据映射在同一维度空间,得到维度相同的项目特征  $\mathbf{z}_{ik}^{(I)}$  和项目外部交互数据特征  $\mathbf{z}_{ik}^{(T)}$ :

$$\mathbf{z}_{ik}^{(l)} = \text{aggregate}(\{\mathbf{z}_{jk}^{(l)}, \forall j \in T\}) \quad (1)$$

$$\mathbf{z}_{ik}^{(l)} = \mathbf{W} \cdot \text{concat}(\mathbf{z}_{ik}^{(l)}, \mathbf{z}_{ik}^{(l)}) \quad (2)$$

式中: aggregate 为聚合器, 本文算法选择的是将邻居节点的特征和自身节点的特征求均值;  $\mathbf{z}_{ik}^{(l)}$  和  $\mathbf{z}_{ik}^{(l)}$  是第  $k$  个项目-项目交互数据二部图中得到的项目特征;  $\mathbf{W}$  是权重矩阵; concat 是拼接函数。

在文献[25]的基础上, 本文设计了一个改进的图注意力机制用于注意力层, 它可以考虑多个不同的关注方向, 以更好地捕捉不同方向上的信息。通过项目关系矩阵  $\mathbf{R}_u \in \mathbf{R}^{n \times j}$  ( $n$  表示项目数量,  $j$  表示项目交互数据数量) 得到项目-项目交互数据特征的相关值, 并将其映射到对应维度  $d$  上最终得到注意力分数  $\omega_{it}$ , 公式如下所示:

$$\omega_{it} = \frac{\mathbf{z}_{ik}^{(l)} \mathbf{z}_{ik}^{(l)\top} \mathbf{R}_u}{\sqrt{d}} \quad (3)$$

在式(3)的基础上, 通过项目和项目外部交互数据的相关性, 计算项目特征  $i$  和项目外部交互数据特征  $t$  的注意力权重, 最终的注意力权重  $\alpha_{it}$  由 Softmax 函数归一化注意力分数  $\omega_{it}$  得到:

$$\alpha_{it} = \text{Softmax}(\omega_{it}) = \frac{\exp(\omega_{it})}{\sum_{t \in T} \exp(\omega_{it})} \quad (4)$$

最终通过从式(4)中得到的注意力权重  $\alpha_{it}$  获得具有不同特征影响项目特征  $\mathbf{z}_{ik}^{(l)}$ :

$$\mathbf{z}_{ik}^{(l)} = \sum_{t \in T} \alpha_{it} \mathbf{z}_{ik}^{(l)} + \mathbf{z}_{ik}^{(l)} \quad (5)$$

由于本文算法考虑的是多个外部数据, 因此需要在最后对每个项目关系交互数据二部图求得的项目特征进行融合操作:

$$\mathbf{z}_i^{(l)} = \sum_{k=1}^K \mathbf{z}_{ik}^{(l)} \quad (6)$$

式中:  $\mathbf{z}_i^{(l)}$  为最终求得的项目特征;  $K$  为项目-项目交互数据二部图的个数。

### 2.3 特征嵌入

特征嵌入模块主要使用一种简化的 GCN, 原因是 GCN 表现出对非欧几里得数据强大的建模能力, 有利于表现数据之间的关系。因此, 为将用户-项目二部图中的隐式信息有效作用于推荐算法中, 本文通过一种使用简单加权和聚合器的 GCN, 利用二部图的结构特征来学习节点的表示。为了实现这一点, 算法迭代地执行 GCN, 通过聚合邻居的特征作为目标节点的新表示。GCN 的邻域聚合可以表示为:

$$\mathbf{z}_u^{(l+1)} = \sum_{i \in N_u} \frac{1}{\sqrt{|N_i|} \sqrt{|N_u|}} \mathbf{z}_i^{(l)} \quad (7)$$

$$\mathbf{z}_i^{(l+1)} = \sum_{u \in N_i} \frac{1}{\sqrt{|N_i|} \sqrt{|N_u|}} \mathbf{z}_u^{(l)} \quad (8)$$

式中:  $\frac{1}{\sqrt{|N_i|} \sqrt{|N_u|}}$  是对称归一化项, 主要作用是避免了嵌入规模随 GCN 的运算而增大;  $N_u$  表示与用户  $u$  有交互关系的相关项目的集合;  $N_i$  表示与项目  $i$  有交互关系的相关用户的集合。

设  $\mathbf{z}_u^{(0)}$  是所有用户的初始化嵌入,  $\mathbf{z}_i^{(0)}$  是所有项目的初始化嵌入, 其中项目的嵌入来自于项目特征初始化模块。高层嵌入可以经由式(7)和式(8)来计算。在经过  $l$  层 GCN 之后, 进一步联合计算在每一层获得的特征嵌入以形成用户和项目的最终表示:

$$\begin{aligned} \mathbf{z}_u &= \sum_{l=0}^L \alpha_l \mathbf{z}_u^{(l)} \\ \mathbf{z}_i &= \sum_{l=0}^L \alpha_l \mathbf{z}_i^{(l)} \end{aligned} \quad (9)$$

式中:  $\alpha_l \geq 0$  表示第  $l$  层嵌入在构成最终嵌入中的重要性。设置  $\alpha_l = \frac{1}{L+1}$ , 其中,  $L$  为 GCN 的总层数。

### 2.4 图对比学习

图对比学习模块中需要执行的是基于图的对比学习方法, 主要作用是: 完成节点自我辨别, 从而提供辅助监督信号; 通过边丢失算子来有意减小高阶节点的影响和缓解度偏差; 利用节点多重视图的不同局部结构和邻域增强算法性能。该模块主要包含二部图增强和对比损失 2 个部分。

#### 2.4.1 二部图增强

二部图结构中 2 种类型的节点是有内在关联的, 具体到推荐算法中即为用户和项目之间存在交互关系。此外, 这些交互数据是较为稀疏的, 故在其他领域如计算机视觉和自然语言处理等使用的相关操作算子不适用于基于二部图的推荐算法。因此, 需要设计出一种适用于二部图推荐算法的操作算子。

二部图结构主要反映的是用户和项目交互关系, 其中包含协同过滤的信息。例如: 用户的第 1 跳邻近节点包含了该用户节点和相邻项目节点的信息, 即与该用户交互过的历史项目可以被看作是该用户之前存在的特征; 用户的第 2 跳邻近节点可以看作是与该用户行为类似的用户, 即两者同样与某个项目有过交互; 此外, 用户到项目的高阶路径反映了用户对项目的潜在兴趣; 项目侧的信息和用户侧的信息类似。基于此, 本文在二部图结构上设计了

一种基于边丢失的操作算子来创建不同的节点视图。通过边丢失,算法会学习到更具区分性的特征,从而提高了算法的泛化性。此外,由于用户-项目交互数据通常是稀疏且含有噪声的,因此通过边丢失操作能够降低噪声影响。具体操作如下:给定二部图  $G$ ,通过随机删除图  $G$  中的部分边来改动图  $G$  的连通性,具体为丢弃二部图  $G$  的一些连接边,其中每条连接边都以一个随机删除概率  $\rho$  被丢弃。独立操作可以表示为:

$$\begin{aligned} o_1(G) &= (V, \mathbf{M}_1 \odot \rho) \\ o_2(G) &= (V, \mathbf{M}_2 \odot \rho) \end{aligned} \quad (10)$$

式中:  $\mathbf{M}_1, \mathbf{M}_2 \in \{0, 1\}^{|\rho|}$  是二部图的边集合的 2 个掩码向量,通过随机删除比率  $\rho$  生成;  $\odot$  表示 2 个向量的乘积。将该二部图增强操作独立地应用于每次迭代的图结构,即在一个新的训练迭代开始时为每个节点生成 2 个不同的子视图。对于 2 个独立的过程 ( $o_1$  和  $o_2$ ),边丢失率  $\rho$  保持相同的值。

#### 2.4.2 对比损失

视图间节点对比损失函数  $\mathcal{L}_{\text{inter}}$  是在建立了子视图之后,将相同节点的视图视为正对(即  $\{(\mathbf{z}'_u, \mathbf{z}''_u) \mid u \in U\}$ ),以及将任何不同节点的视图视为负对(即  $\{(\mathbf{z}'_u, \mathbf{z}''_v) \mid u, v \in U, u \neq v\}$ )。正对的辅助监督促进同一节点的不同视图之间的一致性,而负对的辅助监督则是强化不同节点之间的不一致性,最后最大化正对的一致性并最小化负对的不一致性来实现视图间节点对比损失。用户侧的损失函数可以表示为:

$$\mathcal{L}_{\text{inter}}^{\text{user}} = \frac{1}{n} \sum_{u \in U} -\ln \frac{\exp(s(\mathbf{z}'_u, \mathbf{z}''_u)/\tau)}{\sum_{v \in U} \exp(s(\mathbf{z}'_u, \mathbf{z}''_v)/\tau)} \quad (11)$$

式中:  $n$  是节点个数;函数  $s$  度量 2 个向量之间的相似性,本文算法中设置为内积操作;  $\tau$  是温度系数。

视图内节点对比损失函数  $\mathcal{L}_{\text{intra}}$  是在建立子视图后,通过约束每个节点的最终表示,接近其一致表示的同时远离其不一致表示来完成视图内节点对比损失。具体地,对于每个用户节点  $u$ ,通过结合子视图表示  $\mathbf{z}_{u,1}, \mathbf{z}_{u,2}$  来生成最终表示,其被公式化为:

$$\mathbf{z}_u = g_2(g_1(\mathbf{z}_{u,1} \parallel \mathbf{z}_{u,2})) \quad (12)$$

式中:  $g_1, g_2$  是 2 个多层感知机;  $\mathbf{z}_u$  是用户节点  $u$  的最终表示。

对于用户节点  $u$  的最终表示  $\mathbf{z}_u$ ,将所有视图中  $u$  的一致表示视为视图内正样本  $\mathbf{z}''_u$ ,并且将所有视图中的其他节点的不一致表示视为视图内负样本  $\mathbf{z}''_v$ 。形式上,用户侧的视图内节点对比损失函数定义如下:

$$\mathcal{L}_{\text{intra}}^{\text{user}} = \frac{1}{n} \sum_{u \in U} -\ln \frac{\sum_{m=1}^M \exp(s(\mathbf{z}_u, \mathbf{z}''_u)/\tau)}{\sum_{v \in U} \sum_{m=1}^M \exp(s(\mathbf{z}_u, \mathbf{z}''_v)/\tau)} \quad (13)$$

式中:  $M$  表示子视图总数量。

类似地,可以得到项目侧的对比损失  $\mathcal{L}_{\text{inter}}^{\text{item}}$  和  $\mathcal{L}_{\text{intra}}^{\text{item}}$ 。结合上述内容,可以得到最终的对比学习的损失函数为:

$$\mathcal{L}_{\text{SSL}} = \mathcal{L}_{\text{inter}}^{\text{user}} + \mathcal{L}_{\text{intra}}^{\text{user}} + \mathcal{L}_{\text{inter}}^{\text{item}} + \mathcal{L}_{\text{intra}}^{\text{item}} \quad (14)$$

#### 2.5 项目推荐

项目推荐建立在最终的表示之上,用来预测推荐项目被相关用户接受的可能性。本文算法采用的解决方案是内积,即通过用户和项目的表征来求得:

$$\hat{y}_{ui} = \mathbf{e}_u \mathbf{e}_i^T \quad (15)$$

为了优化算法参数,本文算法选择了贝叶斯个性化排名(BPR)损失<sup>[26]</sup>,其基本思想是:推荐算法会将用户喜欢的项目排在用户不喜欢的项目前,即通过对得到的预测值与真实值之间进行对比,使推荐结果更接近真实值。BPR 损失可形式化为:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,j) \in S} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \quad (16)$$

式中:  $S$  是用户与项目的交互数据;  $i$  表示与用户  $u$  有过交互记录的项目;  $j$  表示与用户  $u$  未有过交互记录的项目;  $\sigma$  是 Sigmoid 函数;  $\hat{y}_{ui}$  和  $\hat{y}_{uj}$  分别表示正样本和负样本的分值。

通过利用多任务训练策略联合式(14)和式(16)来优化推荐模型,即:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \lambda_1 \mathcal{L}_{\text{SSL}} + \lambda_2 \|\theta\|_2^2 \quad (17)$$

式中:  $\theta$  是算法参数的集合;  $\lambda_1, \lambda_2$  为超参数;  $\|\theta\|_2^2$  表示 L2 正则化,用于防止过拟合现象的发生。

#### 2.6 时间复杂度分析

假设用户-项目二部图中的节点和边的数量分别为  $V_1$  和  $E_1$ ,项目-项目交互数据二部图中的节点和边的数量分别为  $V_2$  和  $E_2$ ,  $d$  表示嵌入大小,  $L$  表示 GCN 层数,  $\rho'$  表示边丢失的保留概率,可以得出特征融合模块的时间复杂度为  $\mathcal{O}(V_2 d + E_2 L d)$ ,特征嵌入模块的时间复杂度为  $\mathcal{O}(\rho' L d E_1)$ ,图对比模块的时间复杂度为  $\mathcal{O}(E_1 d V_1)$ ,综上 FRBGCL 的时间复杂度约为  $\mathcal{O}(V_2 d + E_2 L d + E_1 d (\rho' L + V_1))$ 。

### 3 实验结果与分析

本文提出的 FRBGCL 算法使用 PyTorch 框架实现,所有的实验都在一台 CPU 为 Intel® Core™

i7-12700H, 显卡为 NVIDIA RTX 3060, 内存为 6 GB 的机器上运行。实验环境为 Python 3.7, 运行工具为 PyCharm 2021。为了评估算法的推荐性能, 在实验中对数据集进行随机划分, 将 80% 的数据划分为训练集, 将 20% 的数据划分为测试集, 并将最好的实验结果通过加粗标识, 次优的实验结果通过下划线标识。

### 3.1 实验设置和评价指标

为了测试本算法的性能, 本文采用了 3 个来自不同领域且被广泛使用的公开数据集, 分别是 XuetaangX、Last. fm 和 Yelp2018, 表 1 给出 3 个数据集的具体信息。

表 1 数据集统计信息

Table 1 Datasets statistics

单位: 个

数据集	用户数	项目数	专业/ 类型数	用户-项目 交互数
XuetaangX	17 789	1 182	24	206 090
Last. fm	1 892	12 523	4 081	71 064
Yelp2018	31 668	38 048	10 944	1 561 406

本文主要使用了召回率( $\text{Recall}@K$ )、归一化折损累计增益( $\text{NDCG}@K$ )和命中率( $\text{HR}@K$ )3 个指标来评估算法的效果, 其中,  $K$  是推荐项目数量, 3 个指标均为越大推荐性能越好。3 个指标计算方式如下:

$$r_{\text{Recall}@K} = \frac{|N(K) \cap N(u)|}{|N(u)|} \quad (18)$$

$$r_{\text{NDCG}@K} = \frac{\sum_{i=1}^K \frac{2^{s_{\text{rel}}(i)} - 1}{\text{lb}(i+1)}}{|s_{\text{REL}K}| \sum_{i=1}^K \frac{s_{\text{rel}}(i)}{\text{lb}(i+1)}} \quad (19)$$

$$r_{\text{HR}@K} = \frac{1}{K} \sum_{i=1}^K j_{\text{hit}}(i) \quad (20)$$

式中:  $r_{\text{Recall}@K}$ 、 $r_{\text{NDCG}@K}$ 、 $r_{\text{HR}@K}$  分别表示  $\text{Recall}@K$ 、 $\text{NDCG}@K$  和  $\text{HR}@K$ ;  $N(K)$  表示测试集中推荐给用户的前  $K$  个项目;  $N(u)$  表示测试集中的与用户  $u$  有过交互的项目;  $s_{\text{rel}}(i)$  表示在第  $i$  个位置上的相关性得分;  $|s_{\text{REL}K}|$  表示按  $s_{\text{rel}}(i)$  从大到小的顺序排序, 取前  $K$  个结果组成的集合;  $j_{\text{hit}}(i)$  表示第  $i$  个项目是否包含在算法推荐的项目列表中, 如果第  $i$  个项目在推荐列表中, 则  $j_{\text{hit}}(i) = 1$ , 否则  $j_{\text{hit}}(i) = 0$ 。

### 3.2 基准方法与对比结果

本文使用了 6 个基准方法, 包括基于矩阵分解的算法 MF、基于协同过滤的算法 DirectAU、基于 GCN 的算法 LightGCN、基于图对比学习的算法

SGL、SimGCL 和 XSimGCL。具体介绍如下:

MF<sup>[27]</sup>: MF 通过构造用户-项目交互矩阵, 使用矩阵分解方法来获取用户和项目特征用于推荐。

DirectAU<sup>[28]</sup>: DirectAU 是从对齐和超球面上的均匀性的角度来衡量协同过滤中的表征质量用于推荐。

LightGCN<sup>[12]</sup>: LightGCN 是一种使用轻量级 GCN 操作的推荐算法, 其将节点的最终嵌入构造为节点在所有层上的嵌入的加权和。

SGL<sup>[21]</sup>: SGL 是一个基于图对比学习的推荐算法, 通过辅助的自监督任务来加强节点表示的学习。

SimGCL<sup>[22]</sup>: SimGCL 是一个图对比学习的推荐算法, 通过在嵌入空间中添加均匀噪声创建对比视图来实现对比学习。

XSimGCL<sup>[23]</sup>: XSimGCL 采用基于噪声的嵌入增强来生成对比视图以实现对比操作。

根据表 2~表 4 的实验结果可知, 在 6 个基准方法中, 基于协同过滤的算法 DirectAU 的效果是最差的, 原因在于仅考虑协同过滤的推荐算法对用户与项目之间的信息的挖掘不够充分, 并且 DirectAU 忽略了高阶邻域的信息, 导致算法效果不好。矩阵分解推荐算法 MF 效果也不如基于图结构的推荐算法, 其原因可能是无法充分捕捉每个用户的独特兴趣和偏好, 导致推荐结果不佳。在 4 个基于图结构的推荐方法中, SGL 的效果优于 SimGCL、LightGCN 和 XSimGCL。可能的原因是: SimGCL 和 XSimGCL 在图对比学习部分中对对比学习方面的效果不如 SGL, 而 LightGCN 并没有考虑图对比学习的相关方法, 所以对图结构的学习效果较差, 因此在所有基线算法中 SGL 的效果最好。

本文提出的算法 FRBGCL 在 3 个不同类型的数据集上的所有指标均优于其他基线算法。从表 2~表 4 中可以看出: 随着  $K$  值的增大,  $\text{Recall}@K$ 、 $\text{NDCG}@K$  和  $\text{HR}@K$  指标均在提升, 且在  $K = 20$  时, 各个算法性能达到最佳。在 XuetaangX 数据集上, FRBGCL 和次优算法 SGL 相比,  $\text{Recall}@20$ 、 $\text{NDCG}@20$  和  $\text{HR}@20$  指标分别提升了 2.1%、1.8% 和 1.7%; 在 Last. fm 数据集上, FRBGCL 和次优算法 SGL 相比,  $\text{Recall}@20$ 、 $\text{NDCG}@20$  和  $\text{HR}@20$  指标分别提升了 6.8%、6.1% 和 7.8%; 在 Yelp2018 数据集上, FRBGCL 和次优算法 SGL 相比,  $\text{Recall}@20$ 、 $\text{NDCG}@20$  和  $\text{HR}@20$  指标分别提升了 11.6%、13.1% 和 8.4%。总体来说, 基于图结构的推荐算法性能大部分优于 MF 和 DirectAU, 而在基于图结构的方法中,

LightGCN 只考虑到用户-项目二部图内部所含信息,没有考虑到项目外部交互信息的属性。此外,SGL、SimGCL 以及 XSimGCL 都只考虑了视图间

节点对比。本文提出的算法 FRBGCL 结合了项目的外部交互信息,并引入视图内对比来增强对比学习效果,与基准方法相比有了较大的提升。

表 2 XuetaangX 数据集实验结果

Table 2 Experimental results in XuetaangX dataset

算法	Recall@K			NDCG@K			HR@K		
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20
MF	0.162 0	0.234 7	0.241 2	0.174 3	0.201 1	0.285 1	0.301 5	0.400 8	0.411 6
DirectAU	0.187 9	0.259 3	0.263 1	0.196 9	0.223 4	0.290 8	0.334 9	0.432 1	0.430 3
LightGCN	0.371 9	0.457 3	0.543 8	0.346 2	0.378 3	0.397 4	0.535 4	0.639 3	0.729 2
SGL	<u>0.378 6</u>	<u>0.464 7</u>	<u>0.567 5</u>	<u>0.354 7</u>	<u>0.386 1</u>	<u>0.420 6</u>	<u>0.544 2</u>	<u>0.649 0</u>	<u>0.754 6</u>
SimGCL	0.299 4	0.341 9	0.391 4	0.281 7	0.298 9	0.313 9	0.408 2	0.474 7	0.553 8
XSimGCL	0.350 7	0.431 0	0.526 9	0.327 8	0.357 6	0.388 9	0.502 0	0.605 5	0.712 2
FRBGCL	<b>0.388 0</b>	<b>0.475 5</b>	<b>0.579 4</b>	<b>0.363 7</b>	<b>0.395 1</b>	<b>0.428 1</b>	<b>0.559 9</b>	<b>0.663 3</b>	<b>0.767 7</b>

表 3 Last.fm 数据集实验结果

Table 3 Experimental results in Last.fm dataset

算法	Recall@K			NDCG@K			HR@K		
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20
MF	0.029 9	0.035 7	0.064 2	0.030 1	0.037 2	0.071 5	0.094 3	0.150 9	0.234 7
DirectAU	0.031 7	0.048 5	0.074 5	0.031 8	0.039 9	0.085 5	0.102 2	0.179 2	0.263 6
LightGCN	0.077 7	0.122 6	0.176 2	0.138 1	0.137 1	0.147 3	0.343 8	0.434 7	0.517 8
SGL	<u>0.086 2</u>	<u>0.124 0</u>	<u>0.187 9</u>	<u>0.141 0</u>	<u>0.138 4</u>	<u>0.151 7</u>	<u>0.358 5</u>	<u>0.435 5</u>	<u>0.527 9</u>
SimGCL	0.044 1	0.069 7	0.165 3	0.068 7	0.071 1	0.104 7	0.192 2	0.266 8	0.361 7
XSimGCL	0.070 7	0.103 6	0.151 8	0.106 1	0.109 3	0.117 9	0.287 3	0.376 5	0.470 4
FRBGCL	<b>0.092 8</b>	<b>0.139 4</b>	<b>0.200 6</b>	<b>0.154 1</b>	<b>0.154 8</b>	<b>0.161 0</b>	<b>0.377 1</b>	<b>0.464 7</b>	<b>0.569 3</b>

表 4 Yelp2018 数据集实验结果

Table 4 Experimental results in Yelp2018 dataset

算法	Recall@K			NDCG@K			HR@K		
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20
MF	0.013 8	0.025 1	0.045 5	0.026 7	0.029 5	0.036 8	0.115 7	0.191 4	0.305 9
DirectAU	0.016 2	0.031 1	0.057 8	0.034 5	0.035 3	0.047 1	0.152 1	0.220 7	0.358 8
LightGCN	0.019 8	0.034 1	0.059 4	0.037 1	0.041 1	0.048 8	0.161 1	0.245 2	0.373 4
SGL	<u>0.022 1</u>	<u>0.039 1</u>	<u>0.067 5</u>	<u>0.041 3</u>	<u>0.044 6</u>	<u>0.054 9</u>	<u>0.169 0</u>	<u>0.271 9</u>	<u>0.402 8</u>
SimGCL	0.020 9	0.037 3	0.063 1	0.039 5	0.0427	0.052 0	0.1629	0.261 8	0.386 2
XSimGCL	0.021 4	0.038 1	0.064 2	0.040 1	0.043 4	0.052 7	0.165 7	0.267 1	0.391 8
FRBGCL	<b>0.025 6</b>	<b>0.044 7</b>	<b>0.075 3</b>	<b>0.048 9</b>	<b>0.051 1</b>	<b>0.062 1</b>	<b>0.193 5</b>	<b>0.301 8</b>	<b>0.436 8</b>

### 3.3 消融实验

为了研究不同组件对于最终推荐性能的贡献,本节设计了 3 个不同的算法变体:FRBGCL-WT 表示算法 FRBGCL 不使用项目特征融合;FRBGCL-WC 表示删除视图内节点对比部分;FRBGCL-WTC 表示删除项目特征融合和视图内节点对比 2 个部

分。3 个数据集的消融实验结果分别如表 5~表 7 所示。

从表 5~表 7 中可以看出,项目特征融合和视图内节点对比 2 个部分都对推荐效果起到了积极作用。此外还可以得知:在 XuetaangX 数据集上,视图内节点对比部分的效果要强于特征融合部分;而在



Last.fm 数据集和 Yelp2018 数据集上则相反,特征融合部分的效果要强于视图内节点对比部分的效果。结合表 1 中的信息可以得出可能的原因在于

Last.fm 数据集和 Yelp2018 数据集中的项目种类数较多,项目特征融合部分效果更好。此外,增加的视图内节点对比部分对 3 个数据集都有积极的作用。

表 5 XuetangX 数据集消融实验结果

Table 5 Ablation experiment results in XuetangX dataset

算法	Recall@K			NDCG@K			HR@K		
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20
FRBGCL-WT	0.387 6	0.475 2	0.579 1	0.363 0	0.394 4	0.427 7	0.559 0	0.661 3	0.767 1
FRBGCL-WC	0.386 4	0.475 3	0.578 5	0.362 3	0.394 0	0.426 9	0.557 8	0.659 9	0.766 3
FRBGCL-WTC	0.385 1	0.467 8	0.578 3	0.361 4	0.386 7	0.422 3	0.553 5	0.656 8	0.757 4
FRBGCL	<b>0.388 0</b>	<b>0.475 5</b>	<b>0.579 4</b>	<b>0.363 7</b>	<b>0.395 1</b>	<b>0.428 1</b>	<b>0.559 9</b>	<b>0.663 3</b>	<b>0.767 7</b>

表 6 Last.fm 数据集消融实验结果

Table 6 Ablation experiment results in Last.fm dataset

算法	Recall@K			NDCG@K			HR@K		
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20
FRBGCL-WT	0.091 7	0.138 7	0.197 7	0.152 1	0.153 7	0.160 1	0.376 1	0.463 0	0.565 1
FRBGCL-WC	0.092 0	0.139 0	0.199 1	0.152 6	0.154 3	0.160 8	0.376 6	0.464 1	0.566 1
FRBGCL-WTC	0.090 5	0.137 3	0.197 9	0.150 5	0.152 5	0.159 1	0.374 6	0.462 2	0.562 0
FRBGCL	<b>0.092 8</b>	<b>0.139 4</b>	<b>0.200 6</b>	<b>0.154 1</b>	<b>0.154 8</b>	<b>0.161 0</b>	<b>0.377 1</b>	<b>0.464 7</b>	<b>0.569 3</b>

表 7 Yelp2018 消融实验

Table 7 Ablation experiment results in Yelp2018 dataset

算法	Recall@K			NDCG@K			HR@K		
	K=5	K=10	K=20	K=5	K=10	K=20	K=5	K=10	K=20
FRBGCL-WT	0.024 2	0.042 2	0.072 0	0.046 1	0.048 6	0.059 4	0.185 4	0.291 4	0.426 3
FRBGCL-WC	0.024 5	0.042 8	0.072 4	0.045 9	0.048 9	0.059 8	0.186 0	0.292 3	0.427 3
FRBGCL-WTC	0.023 9	0.042 0	0.071 1	0.045 7	0.047 9	0.058 4	0.185 1	0.290 5	0.419 5
FRBGCL	<b>0.025 6</b>	<b>0.044 7</b>	<b>0.075 3</b>	<b>0.048 9</b>	<b>0.051 1</b>	<b>0.062 1</b>	<b>0.193 5</b>	<b>0.301 8</b>	<b>0.436 8</b>

### 3.4 参数设置分析

在确保实验数据集和实验环境相同的条件下,本文分析研究了 3 个参数( $\lambda_1$ 、温度系数  $\tau$  和 GCN 层数  $L$ ) 在 3 个不同数据集上对本文算法的性能影响,选取的衡量指标为 Recall@20 和 NDCG@20。

#### 3.4.1 超参数 $\lambda_1$

超参数  $\lambda_1$  的作用是控制对比学习相关模块占联合学习任务的比例。为了探究  $\lambda_1$  值对算法性能结果的影响,将提出的算法分别在不同的  $\lambda_1$  值下进行实验, $\lambda_1$  的取值设置为  $\{0, 0.1, 0.2, 0.5, 1.0\}$ ,并对相关结果进行比较分析,实验结果如图 2~图 4 所示。由实验结果可以看出:总体趋势是随着  $\lambda_1$  增大,算法性能呈现先增后减的情况,当  $\lambda_1 = 0.1$  时,算法取得最佳效果;此外,对比学习模块所占比例过大,算法的性能反而会下

降。因此,将  $\lambda_1$  设置成一个合适的值可以使得对比学习模块发挥最大作用,从而使得算法取得最佳性能。

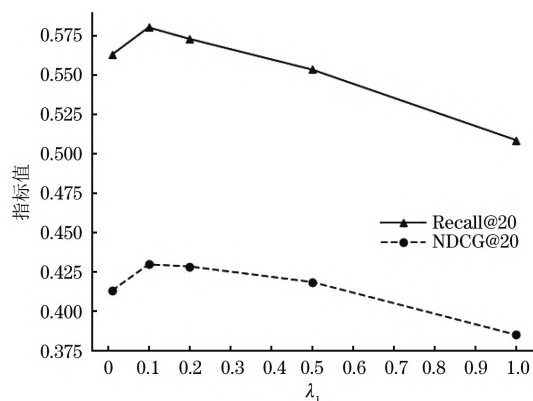
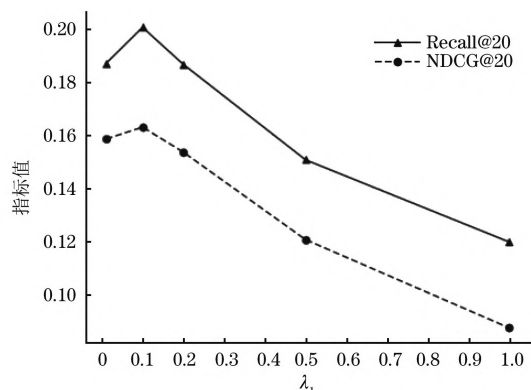
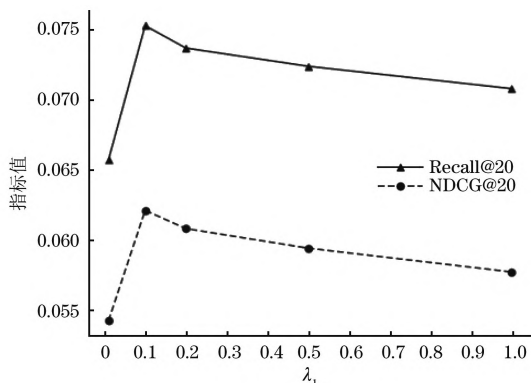


图 2 XuetangX 数据集上不同  $\lambda_1$  值的影响

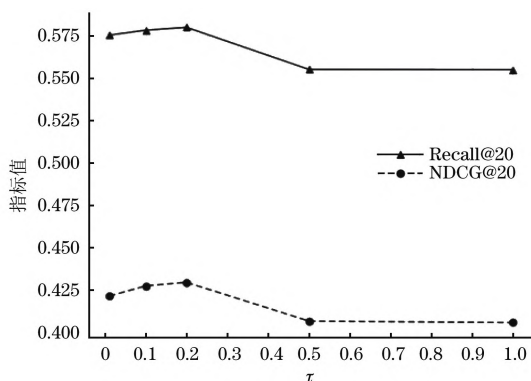
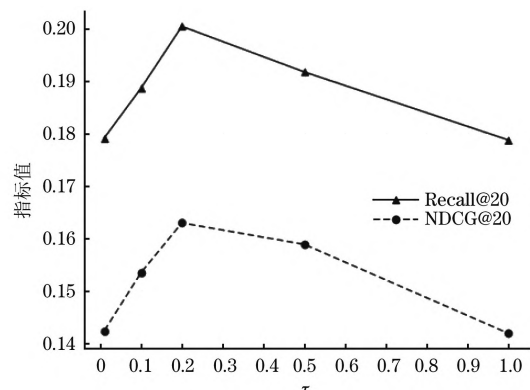
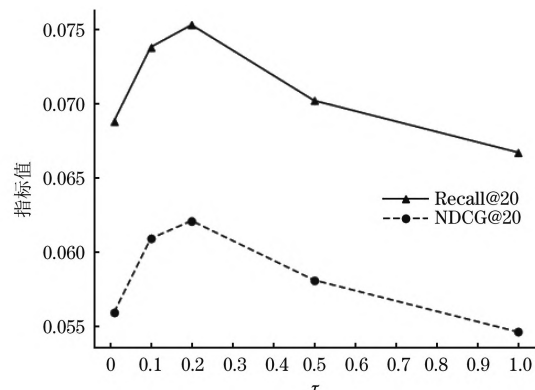
Fig.2 Impact of different  $\lambda_1$  values in XuetangX dataset



图 3 Last.fm 数据集上不同  $\lambda_1$  值的影响Fig.3 Impact of different  $\lambda_1$  values on Last.fm dataset图 4 Yelp2018 数据集上不同  $\lambda_1$  值的影响Fig.4 Impact of different  $\lambda_1$  values in Yelp2018 dataset

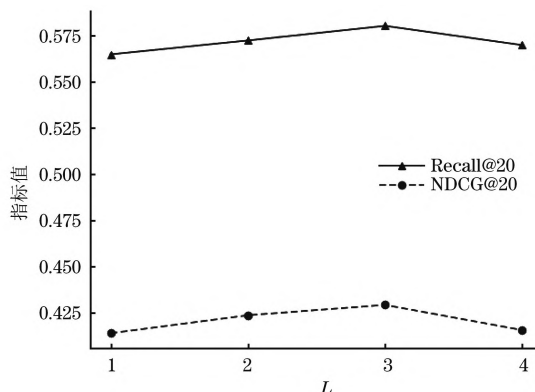
### 3.4.2 温度系数 $\tau$

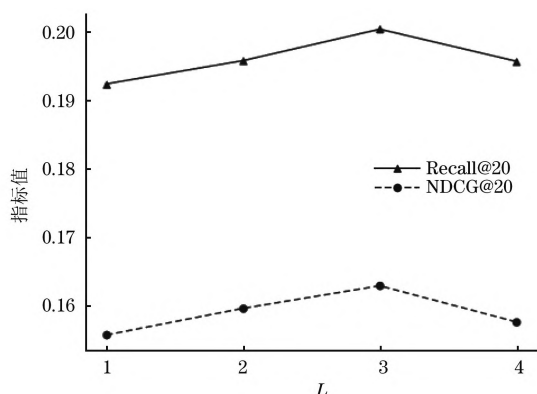
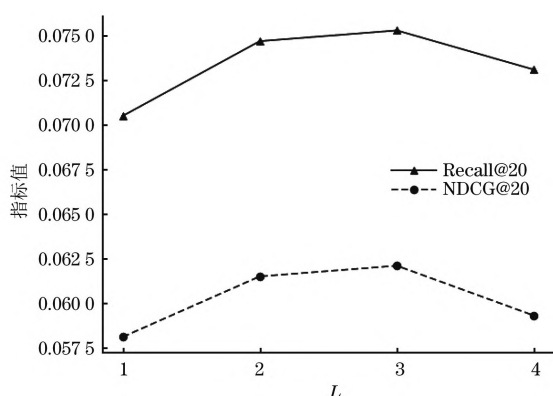
温度系数  $\tau$  在寻找负样本的过程中发挥了重要作用,具体作用是使对比学习中负样本和正样本距离更远,增大相似性得分的差值,明显区分正负样本。因此,通过设置  $\tau = \{0, 0.1, 0.2, 0.5, 1.0\}$  来分析  $\tau$  值的影响。图 5~图 7 分别为不同  $\tau$  值对实验结果的影响折线图。从图中可以看出:随着  $\tau$  值的增加,Recall@20 和 NDCG@20 这 2 个指标都表现出先增大后减小的情况;此外,当  $\tau = 0.2$  时,算法性能达到最优,增大或减小  $\tau$  都会降低算法的性能。

图 5 XuetangX 数据集上不同  $\tau$  值的影响Fig.5 Impact of different  $\tau$  values in XuetangX dataset图 6 Last.fm 数据集上不同  $\tau$  值的影响Fig.6 Impact of different  $\tau$  values in Last.fm dataset图 7 Yelp2018 数据集上不同  $\tau$  值的影响Fig.7 Impact of different  $\tau$  values in Yelp2018 dataset

### 3.4.3 图卷网络积层数 $L$

GCN 层数  $L$  在聚合用户和项目信息方面起着至关重要的作用,本文主要研究  $L = \{1, 2, 3, 4\}$  时对算法性能的影响。图 8~图 10 分别是不同  $L$  值对实验结果的影响。从图中可以看出:增加 GCN 的层数通常可以增强算法的表示能力,能够学习到更好的节点表示;但随着网络层数的增加,算法的参数数量也随之增加,可能会产生过拟合现象;当  $L = 3$  时,算法性能达到最佳,增加或减少 GCN 层数都会影响算法的效果。

图 8 XuetangX 数据集上不同  $L$  值的影响Fig.8 Impact of different  $L$  values in XuetangX dataset

图 9 Last.fm 数据集上不同  $L$  值的影响Fig. 9 Impact of different  $L$  values in Last.fm dataset图 10 Yelp2018 数据集上不同  $L$  值的影响Fig. 10 Impact of different  $L$  values in Yelp2018 dataset

## 4 结束语

本文提出了一种基于二部图对比学习的特征增强推荐算法 FRBGCL, 有效地利用了二部图结构的特性以及图对比学习的优势。FRBGCL 首先通过引入项目特征初始化模块, 将项目外部交互关系数据属性融入到项目特征的学习中, 从而获得增强的项目特征作为推荐任务的项目初始化特征。然后利用 GCN 来学习用户和项目特征的嵌入, 并通过图对比学习来探索节点之间的内在联系, 结合视图间节点对比和视图内节点对比来增强相关节点的表示学习从而提升算法性能。在 3 个基准数据集上进行了大量实验, 结果证明了本文算法在推荐性能方面的优势。在未来的工作中, 将考虑加入用户的社交信息进一步增强其推荐性能。

### 参考文献

[1] PATRO S G K, MISHRA B K, PANDA S K, et al. Cold start aware hybrid recommender system approach for E-commerce users[J]. *Soft Computing*, 2023, 27(4): 2071-2091.

[2] ALAM M, IANA A, GROTE A, et al. Towards analyzing the bias of news recommender systems using sentiment and

stance detection[C]//*Proceedings of the Web Conference*. New York, USA: ACM Press, 2022: 448-457.

- [3] YANG T T, YANG F, MEN J Q. Recommendation content matters! Exploring the impact of the recommendation content on consumer decisions from the means-end chain perspective [J]. *International Journal of Information Management*, 2023, 68: 102589.
- [4] KO H, LEE S, PARK Y, et al. A survey of recommendation systems: recommendation models, techniques, and application fields[J]. *Electronics*, 2022, 11(1): 141.
- [5] 张帅, 高旻, 文俊浩, 等. 基于自监督学习的去流行度偏差推荐方法[J]. *电子学报*, 2022, 50(10): 2361-2371.
- ZHANG S, GAO M, WEN J H, et al. Self-supervised learning for alleviating popularity bias in recommender systems[J]. *Acta Electronica Sinica*, 2022, 50(10): 2361-2371. (in Chinese)
- [6] EZALDEEN H, MISRA R, BISOY S K, et al. A hybrid E-learning recommendation integrating adaptive profiling and sentiment analysis[J]. *Journal of Web Semantics*, 2022, 72: 100700.
- [7] CAI D S, QIAN S S, FANG Q, et al. User cold-start recommendation via inductive heterogeneous graph neural network[J]. *ACM Transactions on Information Systems*, 2023, 41(3): 1-27.
- [8] ZHANG Y M, WU L F, SHEN Q, et al. Graph learning augmented heterogeneous graph neural network for social recommendation[J]. *ACM Transactions on Recommender Systems*, 2023, 1(4): 1-22.
- [9] 曹宗胜, 许倩倩, 李朝鹏, 等. 基于对偶四元数的协同知识图谱推荐模型[J]. *计算机学报*, 2022, 45(10): 2221-2242.
- CAO Z S, XU Q Q, LI Z P, et al. Dual quaternion based collaborative knowledge graph modeling for recommendation [J]. *Chinese Journal of Computers*, 2022, 45(10): 2221-2242. (in Chinese)
- [10] ZHAO N, LONG Z, WANG J, et al. AGRE: a knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder[J]. *Knowledge-Based Systems*, 2023, 259: 110078.
- [11] WANG X, HE X N, WANG M, et al. Neural graph collaborative filtering [C]//*Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, USA: ACM Press, 2019: 165-174.
- [12] HE X N, DENG K, WANG X, et al. LightGCN: simplifying and powering graph convolution network for recommendation[C]//*Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, USA: ACM Press, 2020: 639-648.
- [13] MA X T, DONG L Y, WANG Y Q, et al. AIRC: attentive implicit relation recommendation incorporating content information for bipartite graphs[J]. *Mathematics*, 2020, 8(12): 2132.
- [14] 刘斌. 基于对比学习的推荐算法研究[D]. 武汉: 华中科技大学, 2023.
- LIU B. Research on recommendation algorithm based on comparative learning[D]. Wuhan: Huazhong University of Science and Technology, 2023. (in Chinese)
- [15] LIU Y X, JIN M, PAN S R, et al. Graph self-supervised learning: a survey[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2022, 35(6): 5879-5900.
- [16] QIU J Z, CHEN Q B, DONG Y X, et al. GCC: graph contrastive coding for graph neural network pre-training[C]//*Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, USA: ACM Press, 2020: 1150-1160.

- [17] HE K M, FAN H Q, WU Y X, et al. Momentum contrast for unsupervised visual representation learning [C] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Washington D. C., USA: IEEE Press, 2020: 9729-9738.
- [18] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[EB/OL]. [2023-10-11]. <https://arxiv.org/abs/1706.02216v4>.
- [19] HWANG D, PARK J, KWON S, et al. Self-supervised auxiliary learning with meta-paths for heterogeneous graphs [J]. Advances in Neural Information Processing Systems, 2020, 33: 10294-10305.
- [20] WANG X, LIU N, HAN H, et al. Self-supervised heterogeneous graph neural network with co-contrastive learning [C] // Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2021: 1726-1736.
- [21] WU J C, WANG X, FENG F L, et al. Self-supervised graph learning for recommendation[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press, 2021: 726-735.
- [22] YU J, YIN H, XIA X, et al. Are graph augmentations necessary? Simple graph contrastive learning for recommendation[C]//Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press, 2022: 1294-1303.
- [23] YU J L, XIA X, CHEN T, et al. XSimGCL: towards extremely simple graph contrastive learning for recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 36(2): 1-14.
- [24] JIANG Y Q, HUANG C, HUANG L H. Adaptive graph contrastive learning for recommendation[C]//Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2023: 4252-4261.
- [25] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB/OL]. [2023-10-11]. <https://arxiv.org/abs/1710.10903v3>.
- [26] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback [EB/OL]. [2023-10-11]. <https://arxiv.org/abs/1205.2618v1>.
- [27] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [28] WANG C Y, YU Y Q, MA W Z, et al. Towards representation alignment and uniformity in collaborative filtering [C] // Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2022: 1816-1825.

编辑 金胡考