

融合语义信息与注意力的图神经网络推荐算法^①



闫 阳, 王雷全, 李家瑞

(中国石油大学(华东) 计算机科学与技术学院, 青岛 266580)

通信作者: 闫 阳, E-mail: yanyang@upc.edu.cn

摘 要: 基于图神经网络的推荐算法通过从图中获取知识生成节点的特征表示, 提高了推荐结果的可解释性. 然而, 随着推荐系统原始数据规模的不断扩大, 大量包含语义信息的文本数据没有得到有效利用. 同时图神经网络在融合图中邻居信息时没有区分关键节点, 使得模型难以学习到高质量的实体特征, 进而导致推荐质量下降. 本文将图神经网络与语义模型相结合, 提出一种融合语义信息与注意力的图神经网络推荐算法. 该算法基于 SpanBERT 语义模型处理实体相关的文本信息, 生成包含语义信息的特征嵌入, 并将注意力机制引入到基于用户社交关系以及用户-项目交互的影响传播融合过程中, 从而实现用户和项目两类实体特征的有效更新. 在公开数据集上的对比实验结果表明, 本文所提出的方法较现有基准方法在各项指标上均有所提升.

关键词: 图神经网络; 推荐模型; 语义信息; 注意力机制; 特征提取

引用格式: 闫阳, 王雷全, 李家瑞. 融合语义信息与注意力的图神经网络推荐算法. 计算机系统应用, 2023, 32(4):214-222. <http://www.c-s-a.org.cn/1003-3254/9016.html>

Graph Neural Network Recommendation Algorithm Fused with Semantic Information and Attention

YAN Yang, WANG Lei-Quan, LI Jia-Rui

(College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China)

Abstract: The recommendation algorithm based on the graph neural network generates the feature representation of nodes by obtaining knowledge from graphs, which improves the interpretability of recommendation results. Nevertheless, as the original data scale of the recommendation system continually expands, a large amount of text data containing semantic information has not been effectively used. Additionally, the graph neural network does not distinguish key nodes when fusing the neighbor information in the graph, making it difficult for the model to learn high-quality entity features, which in turn leads to a decrease in the quality of recommendation. This study combines the graph neural network with a semantic model and proposes a recommendation algorithm based on the graph neural network which integrates semantic information and attention. This algorithm processes entity-related text information based on the SpanBERT semantic model and generates feature embeddings containing semantic information. It also introduces the attention mechanism into the process of influence propagation and fusion based on user social relations and user-item interactions to effectively update user and item entity features. The comparative experimental results on public datasets show that the proposed method is better than the existing benchmark methods in all indicators.

Key words: graph neural network; recommendation model; semantic information; attention mechanism; feature extraction

① 基金项目: 中央高校基本科研业务费专项 (19CX05003A-11)

收稿时间: 2022-09-05; 修改时间: 2022-09-30; 采用时间: 2022-10-08; csa 在线出版时间: 2022-12-06

CNKI 网络首发时间: 2022-12-07

随着大数据时代下数字化技术的不断发展,各种互联网应用的出现在为用户提供有效服务的同时,也带来了数据规模的爆炸式增长。面对如此海量的数据,如何从中获取用户感兴趣的内容变得愈发重要。为了解决这一问题,推荐系统应运而生。推荐系统能够根据用户喜好和项目属性来推测用户对项目的偏好,从而决定是否将项目推荐给目标用户。通常,推荐系统通过特定的推荐算法处理用户与项目之间的交互,推荐算法的好坏决定了推荐系统的业务性能。传统协同过滤推荐算法^[1]的主要思路是:寻找与目标用户喜好相似的其他用户,将这些用户感兴趣的内容推荐给目标用户,具体可以分为基于用户的推荐和基于项目的推荐两类方法。但协同过滤方法只考虑了用户对项目的交互,没有关注其属性特征。基于内容的推荐算法^[2]能够根据用户和项目的诸多信息,比如历史评价、收藏文档等,提取用户和项目的属性特征,然后根据这些特征进行相似推荐。基于图结构的推荐算法先将用户和项目之间的交互建模成一个二分图,然后分析二分图的结构以给出合理推荐。混合推荐算法则是融合多个单一的推荐算法,使其相互取长补短,以解决各自的问题。但是由于传统推荐算法具有数据稀疏和冷启动问题^[3],所以在复杂场景下的推荐性能不高。

近年来,深度学习技术不断发展,越来越多的研究者将深度学习模型应用到推荐领域中,并取得了显著效果。He等^[4]在建模用户和项目之间的交互时,利用多层感知机(multi-layer perceptron, MLP)代替传统的矩阵分解(matrix factorization, MF)方法,在两个公共数据集上取得了更好的推荐性能。Zheng等^[5]提出了一个基于CNN构建的推荐模型DeepCoNN,该模型由两个在最后一层耦合的并行CNN组成,两个CNN能够从评论文本中联合抽取丰富的语义信息,从而分别学习用户和项目的特征,但是比较依赖评论文本数据的质量。Zhang等^[6]考虑到图像中可能包含文本没有的额外信息,使用包含CNN和RNN的混合模型进行推文主题推荐。该模型通过VGGNet提取图片的特征信息,通过LSTM(long short-term memory)网络提取文本内容的特征信息,然后使用注意力机制对两者进行融合。从Twitter上收集的数据的实验结果证明,这种多模型的组合在相同推荐任务中的准确率相对于单模型有较大提升。由于深度学习能够凭借其强大的表征学习能力提取目标的深层特征,并挖掘用户和项目之间的非

线性关系,因此对推荐效果的提升具有很大帮助^[7]。

在使用推荐系统的场景中,大多数信息具有不规则的图结构,比如电子商务、社交发现等。图神经网络(graph neural network, GNN)作为一种处理非欧式空间数据的深度学习模型,能够通过图中节点之间的消息传递来捕捉数据之间的潜在联系,因此目前许多研究开始将图神经网络应用于推荐问题中^[8-9]。例如, Li等^[10]面向大规模电子商务领域,提出了一种分层二部图神经网络模型HiGNN,模型首先将多个GNN模块叠加,然后交替使用确定性聚类算法以捕捉分层模块中用户和项目的信息,进而在更大范围内高效地捕捉用户的潜在偏好。通过在真实的电子商务数据集上的实验结果表明,HiGNN相对于主流的推荐方法取得了显著改进。葛尧等^[11]提出了一种基于图卷积网络的推荐算法GCN4RS,使用两组图卷积网络来分别提取异质和同质交互信息,然后统一两类交互信息以提升推荐系统性能。Fan等^[12]将GNN用于社交推荐中,分别考虑用户和项目的两类数据的交互和观点,并利用注意力机制聚合节点的邻居信息,但其使用随机初始化的方式获取用户和项目的初始嵌入,没有考虑隐含的语义信息。

本文考虑到用户和项目本身具有一定的属性信息,比如评论内容、商品描述等,而这些文本内容中包含了大量的语义信息。同时图中不同邻居对目标节点的影响不同,在通过影响传播学习目标节点的嵌入表示过程中具有不同大小的贡献,通过注意力机制计算邻居节点的注意力权重,再按照权重聚合相关邻居节点的特征信息,可以使得学习到的目标节点的嵌入表示更加准确合理。因此,本文构建了一种融合语义信息和注意力机制的图神经网络推荐算法,结合文本中隐含的语义信息和网络中交互关系的影响,学习用户和项目的嵌入表示,从而提高推荐结果的准确性。

本文主要贡献如下:(1)基于项目、用户的社交评论等文本信息,使用语义模型SpanBERT获得节点的语义嵌入表示,通过多特征联合表示方法融合多维度的特征信息。(2)通过用户社交以及用户-项目交互两项关系结构,利用图神经网络聚合高阶拓扑结构中的影响信息,并使用注意力机制增强目标节点的特征表示能力,提高推荐模型的质量。(3)通过在公共数据集上与多个经典模型进行实验对比,验证本文提出推荐算法的有效性。

后文的章节安排如下:第1节介绍用户-项目推荐

的问题定义以及相关知识的简介;第2节介绍本文提出模型的主要结构以及具体实现细节;第3节介绍数据集、实验环境以及实验结果与分析;第4节为总结与展望。

1 相关知识

本节主要介绍模型的问题定义,以及与模型架构相关的图神经网络、注意力机制、SpanBERT 的相关知识。

1.1 问题定义

在推荐系统中通常包括用户和项目两种实体集合,分别被表示为 $U = \{u_1, u_2, \dots, u_M\}$ 和 $V = \{v_1, v_2, \dots, v_N\}$, 其中 $|U| = M$, $|V| = N$, 分别代表用户和项目集中实体的数量。用户与项目的交互展示了用户的偏好,使用用户-项目交互矩阵 $R = \{r_{uv} | u \in U, v \in V\}$ 表征用户的隐式反馈。其中, $R \in \mathbb{R}^{M \times N}$, 当 $r_{uv} = 1$ 时, 用户 u 和项目 v 之间存在交互, 当 $r_{uv} = 0$ 时, 用户 u 和项目 v 之间不存在交互。用户与用户之间的交互展示了用户的社交关系, 可以使用用户-用户交互矩阵 $S = \{s_{ab} | a \in U, b \in U\}$ 表示。其中, $S \in \mathbb{R}^{M \times M}$, 当用户 a 和用户 b 之间存在关系时, $s_{ab} = 1$, 否则 $s_{ab} = 0$ 。当使用无向网络表示用户社交时, 用户 a 和用户 b 存在关系可表示为 $s_{ab} = s_{ba} = 1$ 。另外, 每个用户和项目实体在现实场景中都具有相关的实体属性, 比如用户简介、项目描述、用户评价等。分别使用 $X \in \mathbb{R}^{M \times D_x}$ 和 $Y \in \mathbb{R}^{N \times D_y}$ 表示用户和项目的属性特征矩阵, 则用户 a 的属性特征表示为 X 的第 a 行向量 X_a , 项目 i 的属性特征表示为 Y 的第 i 行向量 Y_i 。

给定用户-用户交互矩阵 S 、用户-项目交互矩阵 R , 以及用户属性特征矩阵 X 、项目属性特征矩阵 Y , 本文模型的最终目标是预测一个函数 $\hat{R} = f(S, R, X, Y; \theta)$ 。其中, $\hat{R} \in \mathbb{R}^{M \times N}$ 表示用户与项目交互的概率, θ 表示模型的训练参数。

1.2 图神经网络

图神经网络 (graph neural network, GNN) 提出的目的是对图域中的数据进行有效处理^[13], 即在图上对节点进行特征提取和嵌入表示。其主要思路是: 首先利用自身特征或相关节点等方法获取图中节点的描述, 然后通过输入邻域信息不断更新节点的状态, 得到具有包含邻居节点信息和图形拓扑结构的状态嵌入, 最终对这些状态嵌入进行整合映射, 输出所需结果如分类标签等。图神经网络的状态更新与结果输出过程为:

$$h_n = f(x_n, x_{CO[n]}, h_{NE[n]}, x_{NE[n]}) \quad (1)$$

$$o_n = g(h_n, x_n) \quad (2)$$

其中, h_n 代表节点 n 的当前状态; x_n 代表节点 n 的特征; $x_{CO[n]}$ 表示以节点 n 为顶点的所有边的特征; $h_{NE[n]}$ 表示节点 n 更新前的状态; $x_{NE[n]}$ 表示节点 n 的邻居节点的特征; $f(\cdot)$ 为局部转换函数, 在所有节点之间共享; $g(\cdot)$ 为局部输出函数, 描述了输出的产生方式; o_n 表示网络的输出结果。

1.3 注意力机制

注意力机制可以使模型在输入的大量信息中聚焦对当前任务更为关键的部分, 降低对于非关键信息的关注度, 从而获取更多与目标相关的细节信息, 而忽视其他的无用信息^[14]。

对于一组输入向量 $X = [x_1, x_2, \dots, x_n]$, 和查询向量 q , 在计算注意力值时, 首先通过打分函数计算 q 与每个输入 x_i 的相关性, 得到一个相关度分数, 然后使用 *Softmax* 函数对这些分数进行归一化, 归一化后的结果则是查询向量 q 在各个输入 x_i 上的注意力分布 $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]$, 其中的每一项都与原始输入向量 $X = [x_1, x_2, \dots, x_n]$ 一一对应, α_i 的计算公式如下:

$$\alpha_i = \text{Softmax}(s(x_i, q)) = \frac{\exp(s(x_i, q))}{\sum_{j=1}^n \exp(s(x_j, q))} \quad (3)$$

其中, $s(\cdot)$ 指相关度计算函数。

之后, 根据得到的注意力分布可以从输入数据中有选择性地提取有效信息, 即根据注意力分布对输入向量进行加权求和, 得到最终的注意力值, 其计算公式为:

$$\text{Attention}(X, q) = \sum_{i=1}^n \alpha_i x_i \quad (4)$$

在神经网络中, 不同的邻居节点对目标节点产生的影响不同, 应用注意力机制不仅能够加速模型的训练过程, 而且能够较好地捕捉局部信息, 即学习邻居节点对于目标节点的贡献度。注意力机制的示意图如图 1 所示。

设目标节点 i 的特征向量为 h_i , 其邻居节点 j 的特征向量为 h_j , 则邻居节点 j 对目标节点 i 的贡献度计算公式如下:

$$a_{ij} = \frac{\exp(\text{LeakyReLU}(\alpha^T [Wh_i \oplus Wh_j]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\alpha^T [Wh_i \oplus Wh_k]))} \quad (5)$$

其中, a_{ij} 为目标节点的邻居节点集合; \oplus 表示向量的拼接操作; $LeakyReLU$ 为激活函数; $W \in R^{F' \times F}$ 为参数矩阵, 用于将 F 维的特征向量线性变换为 F' 维; $\alpha \in R^{2F'}$ 是输入层到隐藏层之间的权值向量; a_{ij} 为最终经归一化操作后的注意力权重。

将目标节点的所有邻居节点特征按贡献度进行加权求和, 并经过非线性变换后, 得到目标节点更新后的特征如下:

$$h_i' = \sigma \left(\sum_{j \in N_i} a_{ij} W h_j \right) \quad (6)$$

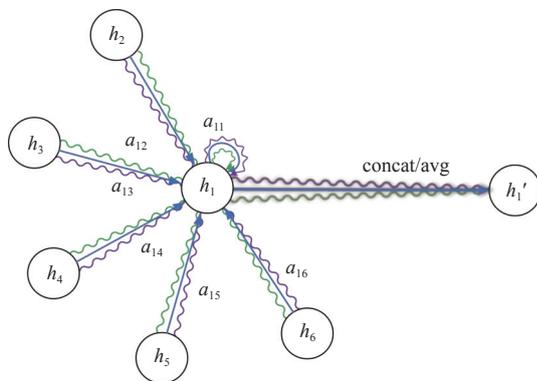


图1 注意力机制示意图

1.4 SpanBERT 模型

SpanBERT 是一种预训练语言模型, 主要基于 BERT 模型进行优化^[15]. 不同于其他方法通过增加训练的数据和加大模型的尺寸来提升性能, SpanBERT 主要通过优化预训练模型的训练目标来提升效果. 考虑到许多 NLP 任务中都涉及推理两个或多个文本区间之间的关系, 因此 SpanBERT 被设计用来更好地表示和预测区间, 其在 BERT 模型的基础上进行的改动如下。

(1) 与 BERT 对于随机词符 (token) 进行遮蔽不同, SpanBERT 会对连续的随机区间 (span) 进行遮蔽。

(2) SpanBERT 通过训练文本区间的边界表示, 对整个被遮蔽的区间进行预测, 而非其中的单个词符. 具体来说, 模型引入了 SPO (span boundary objective) 训练方式, 以促进将区间层面的信息存储在其边界的词符表示上, 从而在微调阶段获得更好的效果。

(3) BERT 中包含下一句预测 (next sentence prediction, NSP) 的预训练任务, 此任务输入两个文本序列, 目的是预测二者是否具有连接关系. 但实验发现,

这种训练方法几乎总是比只用单个文本序列训练更差. 因此, SpanBERT 去除了 NSP 任务, 仅仅使用了单个片段序列进行训练和预测。

模型训练过程如图 2 所示. 通过实验对比, SpanBERT 在许多任务上达到了比 BERT 更好的结果, 并在有关文本区间选择的任务上取得了非常大的提升, 包括问答匹配和共指消解等。

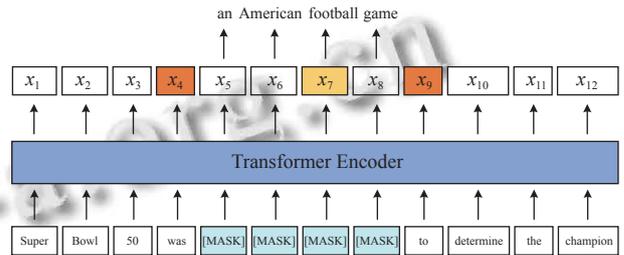


图2 SpanBERT 模型训练示意图

2 算法模型

通常物品名称或描述中包含大量语义信息, 而 SpanBERT 模型可以通过训练, 得到包含这些语义信息的词嵌入. 用户和项目的历史交互能够构成一种图结构, 图神经网络能够利用图结构聚合邻居节点的信息, 从而进行节点间的影响传播. 而注意力机制可以获取到不同节点对当前节点的影响大小, 在聚合时分配不同的权重, 使影响更加符合真实情况. 因此, 本文提出一种融合语义信息和注意力机制的图神经网络推荐算法, 算法流程如图 3 所示。

该算法主要包含 3 个部分: 语义嵌入生成、融合注意力机制的图神经网络模型以及最终的双线性预测部分. 其中, 语义嵌入生成模块基于实体的文本属性信息生成其初始的语义特征向量; 初始特征融合模块初始化实体的自由特征向量, 并分别对用户和项目的两类特征向量进行融合, 生成统一的向量表示; 用户特征传播更新和项目特征传播更新模块利用图神经网络的思想, 基于网络实体之间的关系结构, 结合注意力机制对目标节点进行层级的特征传播与更新; 最后的推荐评分预测模块通过 DistMult 函数计算实体之间的相似度, 得到用户-项目实体对的推荐评分。

2.1 语义嵌入生成

模型中语义嵌入生成模块的主要任务是根据实体的文本属性信息, 通过语义模型生成各个实体的语义

特征向量. 根据推荐领域中数据的不同特点, 本文为不同种类的实体设计对应的语义特征生成方法: 在推荐领域, 通常项目类实体中包含较多的文本属性数据, 比如项目说明、用户对项目的评价等, 因此可以直接使

用文本特征抽取方法提取其中的语义信息; 而用户类实体大多数不包含文本数据, 但在推荐网络中每个用户实体都有与其相关的项目实体, 因此可以通过聚合邻居项目节点的特征向量得到其特征表示.

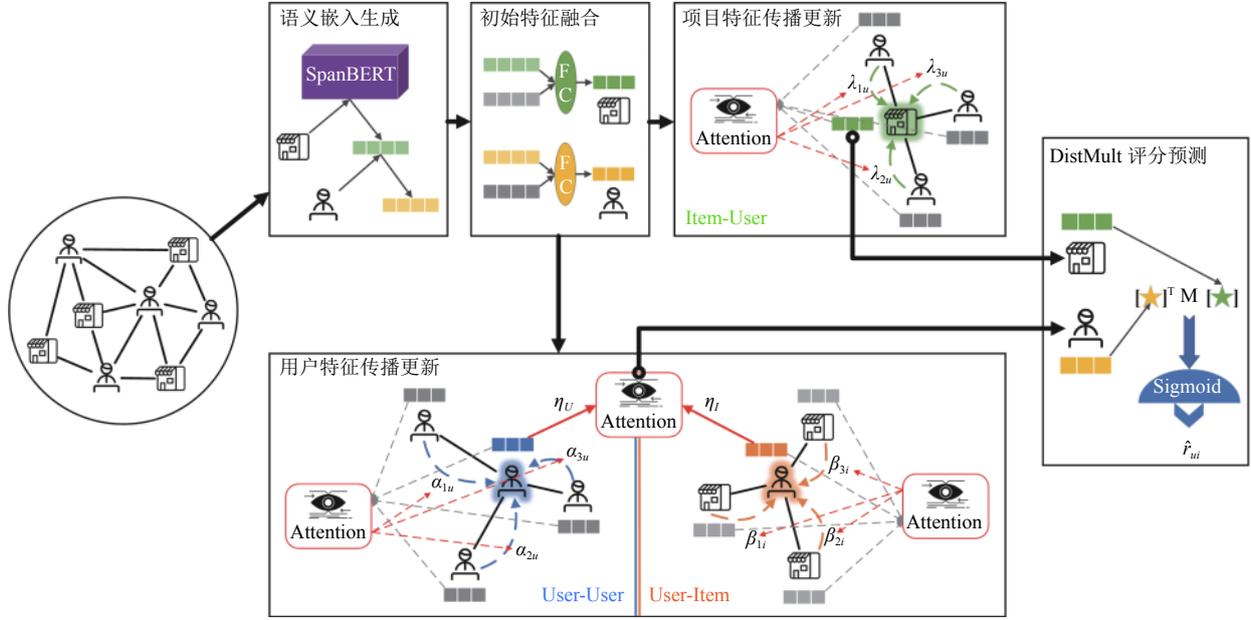


图3 模型整体框架图

基于上述思路, 本文使用 SpanBERT 模型获取项目类数据的初始特征向量. 对于项目类实体*i*, 其属性特征向量 x_i 的计算过程如式 (7) 和式 (8) 所示:

$$x_i = \sum_{a=1}^w \frac{W_i^a}{w}, a \in [1, w] \quad (7)$$

$$W_i^a = \text{SpanBERT}(i^a) \quad (8)$$

其中, i^a 指对实体*i*的描述进行分词后, 得到的第*a*个分词, 分词总数为*w*; SpanBERT(·)为使用 SpanBERT 模型获取词语词向量的函数, 这里使用 SpanBERT (large) 模型, 由 24 个 Encoder 单元组成, 每层的隐藏状态尺寸为 1024 维, 并有 16 个注意力头, 共 340M 参数; W_i^a 为分词 i^a 对应的词向量.

用户类实体的初始特征向量由邻居项目类实体融合生成, 对于用户类实体*j*, 其属性特征向量 y_j 的计算过程如式 (9) 所示:

$$y_j = \sum_{i \in NX_j} \frac{x_i}{|NX_j|} \quad (9)$$

其中, NX_j 指用户类实体*j*的项目类邻居实体集合; x_i 指项目类实体*i*的属性特征向量.

2.2 融合注意力机制的图神经网络模型

2.2.1 初始特征融合

为了捕捉用户和项目类实体的隐含特征表示, 模型首先随机初始化矩阵 $P \in \mathbb{R}^{M \times D}$ 代表项目类实体的隐含特征, 矩阵 $Q \in \mathbb{R}^{N \times D}$ 代表用户类实体的隐含特征, 其中 D 指隐含特征向量的维数, M 为用户类实体的个数, N 为项目类实体的个数. 对于项目类实体*i*和用户类实体*j*, 输入其 one-hot 编码, 经过下标运算, 能够分别输出对应的自由特征向量 p_i 和 q_j .

特征融合模块由一层全连接层构成, 输入属性特征向量以及自由特征向量, 输出包含节点初始偏好信息的融合特征向量. 对于项目类实体*i*, 特征融合的过程如式 (10) 所示:

$$v_i^0 = f(W_i^0 \times [x_i, p_i]) \quad (10)$$

其中, W_i^0 为权重矩阵; $f(\cdot)$ 是非线性变换函数.

同样对于用户类实体*j*, 其特征融合的过程如式 (11) 所示:

$$u_j^0 = g(W_U^0 \times [y_j, q_j]) \quad (11)$$

其中, W_U^0 为权重矩阵; $g(\cdot)$ 是非线性变换函数.

2.2.2 用户特征传播更新

在得到实体融合后的特征向量后, 本文将注意力机制引入用户和项目的特征更新过程中, 即在聚合与中心节点存在交互的邻居节点特征时, 通过注意力机制为邻居节点分配不同的权重, 从而体现不同邻居节点对目标节点不同的重要性. 从而模拟实体所受潜在影响的动态传播过程.

基于上述思路, 对于每个用户类实体 j , u_j^l 代表其经过第 l 层科研合作传播模块网络处理后的特征向量, 将此特征向量输入至第 $l+1$ 层, 经过传播更新过程后输出的特征向量表示为 u_j^{l+1} . 具体来说, u_j^{l+1} 经过 3 个步骤的更新获得, 首先从 j 的用户类邻居实体中聚合传播影响, 将所有的邻居影响转换为固定长度的向量 $u_{NU_j}^{l+1}$, 转换过程如式 (12) 所示:

$$u_{NU_j}^{l+1} = AGG_{UU}(u_{ju}^l) = \sum_{ju \in NU_j} \alpha_{ju}^{l+1} u_{ju}^l, \quad ju \in NU_j \quad (12)$$

其中, NU_j 代表实体 j 的所有用户类邻居实体; α_{ju}^{l+1} 是用户类邻居 ju 对用户 j 的权重系数, 具体计算公式为:

$$\alpha_{ju}^{l+1} = MLP_1(u_j^l, u_{ju}^l) \quad (13)$$

之后从 j 的项目类邻居实体中聚合传播影响, 将所有的邻居影响转换为固定长度的向量 $u_{NI_j}^{l+1}$, 转换过程如式 (14) 所示:

$$u_{NI_j}^{l+1} = AGG_{UI}(u_{ji}^l) = \sum_{ji \in NI_j} \beta_{ji}^{l+1} v_{ji}^l, \quad ji \in NI_j \quad (14)$$

其中, NI_j 代表实体 j 的所有项目类邻居实体; β_{ji}^{l+1} 是用户类邻居 ji 对用户 j 的权重系数, 具体计算公式为:

$$\beta_{ji}^{l+1} = MLP_2(u_j^l, v_{ji}^l) \quad (15)$$

最终, 将得到的用户社交网络和用户-项目交互网络中的用户隐特征 $u_{NU_j}^{l+1}$ 和 $u_{NI_j}^{l+1}$, 与第 l 层生成的实体 j 的特征向量 u_j^l 组合, 产生第 $l+1$ 层的特征向量 u_j^{l+1} , 组合过程如式 (16) 所示:

$$u_j^{l+1} = u_j^l + \eta_{jU}^{l+1} u_{NU_j}^{l+1} + \eta_{jI}^{l+1} u_{NI_j}^{l+1} \quad (16)$$

其中, η_{jU}^{l+1} 和 η_{jI}^{l+1} 分别表示用户社交网络和用户-项目交互网络对用户 j 的权重系数, 计算公式如下:

$$\eta_{jU}^{l+1} = MLP_3(u_j^l, u_{NU_j}^{l+1}) \quad (17)$$

$$\eta_{jI}^{l+1} = MLP_4(u_j^l, u_{NI_j}^{l+1}) \quad (18)$$

这里得到的用户隐含向量表示作为最终的用户向量表示.

2.2.3 项目特征传播更新

对于每个项目类实体 i , v_i^l 代表其经过第 l 层科研合作传播模块网络处理后的特征向量, 将此特征向量输入至第 $l+1$ 层, 经过传播更新过程后输出的特征向量表示为 v_i^{l+1} . 具体来说, v_i^{l+1} 经过两个步骤的更新获得, 首先从 i 的用户类邻居实体中聚合传播影响, 将所有的邻居影响转换为固定长度的向量 $v_{NU_i}^{l+1}$, 转换过程如式 (19) 所示:

$$v_{NU_i}^{l+1} = AGG_{IU}(u_{iu}^l) = \sum_{iu \in NU_i} \lambda_{iu}^{l+1} u_{iu}^l, \quad iu \in NU_i \quad (19)$$

其中, NU_i 代表实体 i 的所有用户类邻居实体; λ_{iu}^{l+1} 是用户类邻居 iu 对用户 i 的权重系数, 具体计算公式为:

$$\lambda_{iu}^{l+1} = MLP_5(v_i^l, u_{iu}^l) \quad (20)$$

最终, 将得到的项目实体 i 的隐特征 $v_{NU_i}^{l+1}$ 与第 l 层生成的实体 i 的特征向量 v_i^l 组合, 产生第 $l+1$ 层的特征向量 v_i^{l+1} , 组合过程如式 (21) 所示:

$$v_i^{l+1} = \sigma^{(l+1)}(W^{l+1} \times [v_i^l, v_{NU_i}^{l+1}]) \quad (21)$$

其中, $\sigma^{(l+1)}(\cdot)$ 是非线性变换函数.

这里得到的项目隐含向量表示作为最终的项目向量表示.

2.3 评分预测

经过 L 层的传播更新后, 能够分别得到用户 j 和项目 i 的嵌入集合 u_j^l 与 v_i^l , 其中, $l = [0, 1, \dots, L]$. 则用户 j 的最终嵌入表示为 $u_j^* = u_j^0 \oplus u_j^1 \oplus \dots \oplus u_j^L$, 项目 i 的最终嵌入表示为 $v_i^* = v_i^0 \oplus v_i^1 \oplus \dots \oplus v_i^L$, \oplus 表示拼接操作.

在获取用户和项目的嵌入表示后, 本文使用双线性投影函数 DistMult , 将一对节点的嵌入表示转化为它们之间的连接强度 $s(j, i)$ ^[16]. 之后使用 Sigmoid 函数将连接强度的范围转化到 $(0, 1)$ 区间, 得到最终的用户与项目之间的预测评分 \hat{r}_{ji} . 具体计算公式如下:

$$\hat{r}_{ji} = \text{Sigmoid}(s(j, i)) = \frac{1}{1 + e^{-(u_j^{*T} M_r v_i^*)}} \quad (22)$$

由于最终的评分与用户和项目的组合顺序无关, 所以这里的关系矩阵 M_r 选用对角矩阵, 可以减少双线性模型的参数量^[17].

2.4 模型学习

本文使用经典的交叉熵损失函数进行模型的更新,

损失函数的具体计算公式如下:

$$Loss = \sqrt{\frac{1}{C} \sum_{\substack{j \in U \\ i \in I}} (\hat{r}_{ji} - r_{ji})^2} \quad (23)$$

其中, r_{ji} 为用户 j 和项目 i 真实关系评分, U 是所有用户集合, I 是所有项目集合.

3 实验结果与分析

3.1 实验数据集

本文选择 Yelp 数据集进行对比实验与分析, Yelp 公开数据集是 Yelp 网站相关业务、商户评论和用户数据的子集, 以 JSON 数据格式组织, 获取途径为 <https://www.yelp.com/dataset>. 为了保证实验数据的有效性, 本文经过筛选删除了少于 2 条评分记录和 2 条社交链接的用户数据, 以及被评分数少于 2 次的项目数据, 最终实验数据集的组成及规模如表 1 所示.

表 1 实验数据集介绍

数据类型	数据项	统计值
用户数据	9	17 237
项目数据	14	38 342
评价交互数据	9	204 448
社交链接数据	2	143 765

3.2 实验评估指标

为了更好地训练推荐模型并评估模型效果, 本文使用 HR 和 NDCG 两种指标对推荐结果进行打分. HR 指标的计算过程如式 (24) 所示, 代表是否推荐给用户想要的结果, 关注于推荐的准确性; NDCG 指标的计算过程如式 (25) 所示, 代表推荐给用户想要的结果是否在排名较前的位置上, 关注于推荐的先后性.

$$HR = \frac{1}{N} \sum_{i=1}^N hit(i) \quad (24)$$

$$NDCG = \frac{1}{N} \sum_{i=1}^N \frac{1}{\log_2(p_i + 1)} \quad (25)$$

其中, N 是用户的总数量; $hit(\cdot)$ 代表当前用户需要的结果是否被推荐, 是置 1、否置 0; p_i 指用户 i 需要的结果在推荐结果中的位置.

3.3 实验环境与参数设置

实验硬件环境为 NVIDIA GeForce RTX 3060 显卡, Intel(R) Core(TM) i7-12700H 处理器, 32 GB 运行内存. 软件环境为 Ubuntu 20 系统, Python 3.6,

TensorFlow 1.10.0 开发环境. 模型主要参数设置如表 2 所示.

表 2 模型参数设置

参数项	参数值
语义信息向量维数	1 024
节点嵌入向量维数	128
学习率	0.001
batch size	512
Epoch	160
优化器	Adam

3.4 实验结果分析

为了分析模型的有效性, 本文将数据集按照 8:1:1 的比例随机划分为训练集、验证集和测试集, 分别进行以下实验.

3.4.1 不同层数 L 对结果的影响

在用户特征传播模块中, GNN 网络的层数 L 代表了用户邻居影响的传播深度, L 越大, 则影响传播越深. 因此应当根据问题规模选择合适的层数, 以保证节点最终特征向量的融合效果.

为了评估不同层数 L 对预测结果的影响, 本文将 GNN 的层数分别设置为 $L=1$ 、 $L=2$ 和 $L=3$ 进行实验, 同时将测试集平均分为 10 份计算 HR@10 和 NDCG@10 两项指标值, 最终结果对比如图 4 所示.

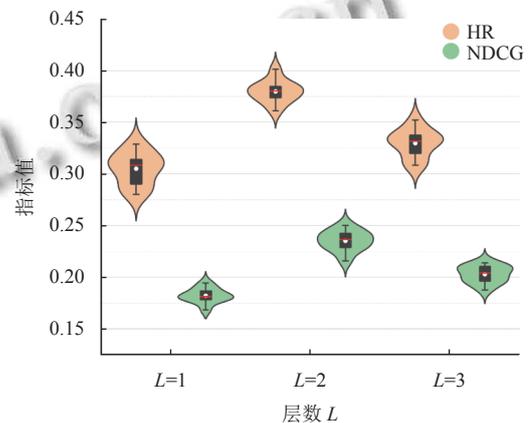


图 4 不同层数 L 下指标对比

从图 4 可以看出, 当 L 从 1 增加到 2 时, 模型取得了更好的效果, 表明更深的传播层数可以聚合更高阶的特征影响; 当 L 继续增大后, 模型的性能开始下降, 表明考虑到实体间的二阶关系影响是足够的. 因此本文在后续实验中均将模型层数设为 2, 即通过两层 GNN 来聚合邻居影响.

3.4.2 模型指标随 Epoch 变化结果

模型在 160 个 Epoch 的训练过程中, HR@10 与 NDCG@10 两个指标的变化情况如图 5 所示. 由实验结果可以看出, 当 Epoch 达到 100 次时, 模型取得最佳性能, 其中 HR@10 稳定在 0.37 左右, NDCG@10 稳定在 0.23 左右.

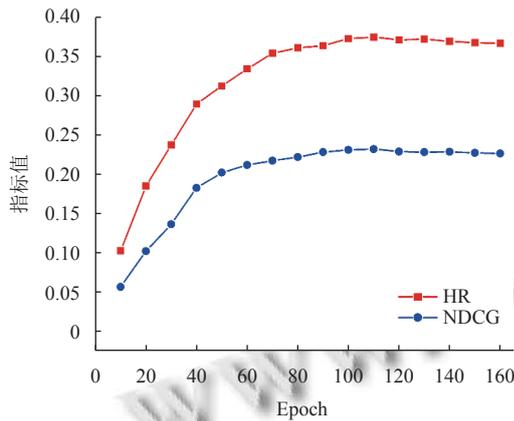


图 5 模型指标随 Epoch 变化结果

3.4.3 不同模型结果对比

为了评估本文模型的推荐性能, 选择了以下推荐模型作为基准模型, 在 Yelp 数据集上进行对比实验.

(1) SocialMF (social matrix factorization)^[18], 模型的主要思想是将信任传播合并到社交推荐的矩阵分解模型中, 使得用户表示接近其信任的用户.

(2) GraphRec (graph recommendation)^[12], 主要思想是按用户间的社交关系以及用户与商品间的交互关系, 通过 MLP 融合用户特征, 然后获取商品特征, 并训练 GNN 预测推荐评分.

(3) DiffNet (diffusion network)^[19], 模型的主要思想是根据用户社交关系和历史行为建模用户偏好, 并使用两层 GNN 来模拟高阶影响的传播.

实验选择 HR 和 NDCG 的 top- N 指标进行评价, 其中 N 分别取值 5、10、15. 最终模型对比的实验结果如图 6 所示.

由上述实验结果可以看出, 本文提出的模型在两个评价指标上均得到了提升. 相比于基线方法使用随机初始化方法生成用户与项目的节点表示, 本文模型采用的融合语义信息与注意力机制的方法具有更好的节点特征表示能力, 能够从多个维度聚合更深层次的关系影响, 验证了模型的有效性.

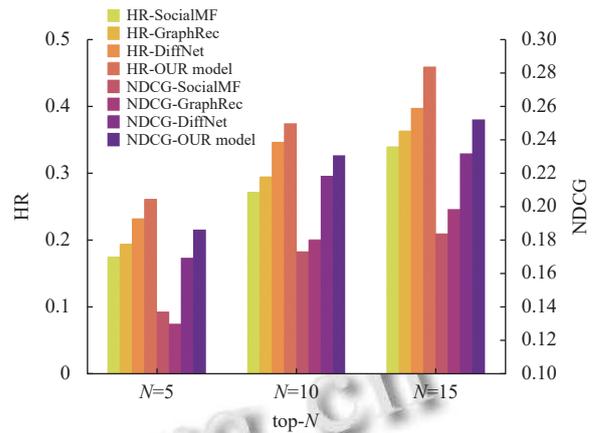


图 6 不同模型对比实验结果

4 结论与展望

针对当前基于图神经网络的推荐算法中没有合适的方法来生成节点的语义信息以及高阶影响的传播聚合等问题, 本文提出了融合语义信息与注意力机制的图神经网络推荐算法, 基于 SpanBERT 语义模型处理实体相关的文本信息, 生成包含语义信息的嵌入表示, 然后考虑用户社交关系以及用户-项目交互, 并设计带注意力机制的图神经网络模拟高阶影响的传播与融合过程, 最后在公开数据集上进行实验, 结果表明本文所提出的方法较现有基准方法在各项指标上均有所提升, 验证了推荐算法的有效性.

今后, 将进一步研究如何有效地生成项目之间的潜在关系, 并将潜在的关系影响融合至项目实体表示学习中, 增强网络节点的特征表示, 进而优化本文模型.

参考文献

- Resnick P, Iacovou N, Suchak M, et al. Grouplens: An open architecture for collaborative filtering of netnews. Proceedings of 1994 ACM Conference on Computer Supported Cooperative Work. Chapel Hill North: ACM, 1994. 175-186. [doi: 10.1145/192844.192905]
- Lops P, de Gemmis M, Semeraro G. Content-based recommender systems: State of the art and trends. Recommender Systems Handbook. Boston: Springer, 2011: 73-105. [doi: 10.1007/978-0-387-85820-3_3]
- 杨博, 赵鹏飞. 推荐算法综述. 山西大学学报(自然科学版), 2011, 34(3): 337-350. [doi: 10.13451/j.cnki.shanxi.univ(nat.sci.).2011.03.001]
- He XN, Liao LZ, Zhang HW, et al. Neural collaborative

- filtering. Proceedings of the 26th International Conference on World Wide Web. Perth: International World Wide Web Conferences Steering Committee, 2017. 173–182. [doi: [10.1145/3038912.3052569](https://doi.org/10.1145/3038912.3052569)]
- 5 Zheng L, Noroozi V, Yu PS. Joint deep modeling of users and items using reviews for recommendation. Proceedings of the 10th ACM International Conference on Web Search and Data Mining. Cambridge: ACM, 2017. 425–434. [doi: [10.1145/3018661.3018665](https://doi.org/10.1145/3018661.3018665)]
- 6 Zhang Q, Wang JW, Huang HR, *et al.* Hashtag recommendation for multimodal microblog using co-attention network. Proceedings of the 26th International Joint Conference on Artificial Intelligence. Melbourne: AAAI Press, 2017. 3420–3426.
- 7 刘华玲, 马俊, 张国祥. 基于深度学习的内容推荐算法研究综述. 计算机工程, 2021, 47(7): 1–12. [doi: [10.19678/j.issn.1000-3428.0060557](https://doi.org/10.19678/j.issn.1000-3428.0060557)]
- 8 Zhou J, Cui GQ, Hu SD, *et al.* Graph neural networks: A review of methods and applications. AI Open, 2020, 1: 57–81. [doi: [10.1016/j.aiopen.2021.01.001](https://doi.org/10.1016/j.aiopen.2021.01.001)]
- 9 Wu SW, Sun F, Zhang WT, *et al.* Graph neural networks in recommender systems: A survey. ACM Computing Surveys, 2022, 55(5): 97. [doi: [10.1145/3535101](https://doi.org/10.1145/3535101)]
- 10 Li Z, Shen X, Jiao YH, *et al.* Hierarchical bipartite graph neural networks: Towards large-scale e-commerce applications. Proceedings of 2020 IEEE 36th International Conference on Data Engineering (ICDE). Dallas: IEEE, 2020. 1677–1688. [doi: [10.1109/ICDE48307.2020.00149](https://doi.org/10.1109/ICDE48307.2020.00149)]
- 11 葛尧, 陈松灿. 面向推荐系统的图卷积网络. 软件学报, 2020, 31(4): 1101–1112. [doi: [10.13328/j.cnki.jos.005928](https://doi.org/10.13328/j.cnki.jos.005928)]
- 12 Fan WQ, Ma Y, Li Q, *et al.* Graph neural networks for social recommendation. Proceedings of the World Wide Web Conference. San Francisco: ACM, 2019. 417–426. [doi: [10.1145/3308558.3313488](https://doi.org/10.1145/3308558.3313488)]
- 13 Scarselli F, Gori M, Tsoi AC, *et al.* The graph neural network model. IEEE Transactions on Neural Networks, 2009, 20(1): 61–80. [doi: [10.1109/TNN.2008.2005605](https://doi.org/10.1109/TNN.2008.2005605)]
- 14 Vaswani A, Shazeer N, Parmar N, *et al.* Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.
- 15 Joshi M, Chen DQ, Liu YH, *et al.* SpanBERT: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 2020, 8: 64–77. [doi: [10.1162/tacl_a_00300](https://doi.org/10.1162/tacl_a_00300)]
- 16 Xue JW, Jiang N, Liang SW, *et al.* Quantifying the spatial homogeneity of urban road networks via graph neural networks. Nature Machine Intelligence, 2022, 4(3): 246–257. [doi: [10.1038/s42256-022-00462-y](https://doi.org/10.1038/s42256-022-00462-y)]
- 17 Yang BS, Yih WT, He XD, *et al.* Embedding entities and relations for learning and inference in knowledge bases. arXiv:1412.6575, 2014.
- 18 Jamali M, Ester M. A matrix factorization technique with trust propagation for recommendation in social networks. Proceedings of the 4th ACM Conference on Recommender Systems. Barcelona: ACM, 2010. 135–142. [doi: [10.1145/1864708.1864736](https://doi.org/10.1145/1864708.1864736)]
- 19 Wu L, Sun PJ, Fu YJ, *et al.* A neural influence diffusion model for social recommendation. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Paris: ACM, 2019. 235–244. [doi: [10.1145/3331184.3331214](https://doi.org/10.1145/3331184.3331214)]

(校对责编: 牛欣悦)