



基于网络表示学习与深度学习的推荐算法研究

刘峰¹, 王宝亮¹, 潘文采²

(1. 天津大学 信息与网络中心, 天津 300072; 2. 天津大学 电气自动化与信息工程学院, 天津 300072)

摘要: 针对传统基于协同过滤的推荐算法信息提取能力有限的问题, 提出基于网络表示学习的卷积协同过滤推荐算法。将二分网络分成物品与用户同质网络, 在各自的两类网络上使用 GraphSAGE 模型得到融合网络空间信息和用户与物品属性信息的矩阵。在此基础上, 利用外积运算丰富用户和物品特征向量各维度的相关表示, 通过卷积神经网络训练物品和用户的交互信息得到算法模型。实验结果验证了该算法的有效性, 且相比 ConvNCF 算法, 其在 Movielens 数据集上 HR@5 和 NDCG@5 分别提升了 1.89 和 2.19 个百分点, 在 Last.fm 数据集上 HR@5 和 NDCG@5 分别提升了 1.09 和 2.32 个百分点。

关键词: 推荐算法; 网络表示学习; 深度学习; 卷积神经网络; 协同过滤

开放科学(资源服务)标志码(OSID):



中文引用格式: 刘峰, 王宝亮, 潘文采. 基于网络表示学习与深度学习的推荐算法研究[J]. 计算机工程, 2021, 47(8): 54-61.
英文引用格式: LIU F, WANG B L, PAN W C. Research on recommendation algorithm based on network representation learning and deep learning[J]. Computer Engineering, 2021, 47(8): 54-61.

Research on Recommendation Algorithm Based on Network Representation Learning and Deep Learning

LIU Feng¹, WANG Baoliang¹, PAN Wencai²

(1. Information and Network Center, Tianjin University, Tianjin 300072, China;

2. School of Electrical Automation and Information Engineering, Tianjin University, Tianjin 300072, China)

[Abstract] The traditional recommendation algorithms based on collaborative filtering information are faced with the bottleneck of the information extraction capability. To address the problem, a recommendation algorithm based on convolution collaborative filtering for network representation learning is proposed. The algorithm decomposes the bipartite network into a homogenous network of users and items, and then uses the GraphSAGE algorithm for the two homogenous networks to obtain feature vectors. On this basis, the outer product operation is performed to enrich the relevant representation of each dimension of the user and item feature vectors. Finally, the convolutional neural network is used to train the interaction information between the item and the user to obtain the algorithm model. The experimental results show that the algorithm is effective, compared with the ConvNCF algorithm, the proposed algorithm improves HR@5 by 1.89 percentage point and NDCG@5 by 2.19 percentage point on the Movielens data set. The improvement of HR@5 is 1.09 percentage point and that of NDCG@5 is 2.32 percentage point on the Last.fm data set.

[Key words] recommendation algorithm; Network Representation Learning (NRL); Deep Learning (DL); Convolutional Neural Network (CNN); collaborative filtering

DOI: 10.19678/j.issn.1000-3428.0058631

0 概述

随着互联网技术的高速发展, 人类社会进入了信息爆炸的时代。为缓解信息过载问题, 对推荐系统的研究迫在眉睫。推荐系统主要根据用户行为推

荐用户偏好的内容^[1], 在电子商务^[2]、新闻^[3]、短视频^[4]等领域发挥着重要作用。

用户-物品推荐是推荐系统比较常见的运用, 其对应于图模型中的二部图。大多数网络表示学习方法不能直接应用在二部图模型中研究同质网络问题。

基金项目: 赛尔网络下一代互联网技术创新项目(NGII20160206)。

作者简介: 刘峰(1963—), 男, 副研究员, 主研方向为信号与信息处理; 王宝亮, 高级工程师; 潘文采, 硕士研究生。

收稿日期: 2020-06-15 **修回日期:** 2020-08-08 **E-mail:** 1959190293@qq.com

网络表示学习法主要构建适合处理节点类型较多的网络知识图谱进而解决异质网络问题, 而二部图模型缺乏针对性。文献[5]提出基于图模型的推荐算法可以通过随机游走理论解释, 但该算法时间复杂度较高且不能体现用户节点与物品节点的差异性。

将网络表示学习法运用到推荐算法中具有较好的效果^[6]。传统网络表示学习法是根据一定的假设构造矩阵, 保留了网络结构信息的矩阵, 进一步得到节点的低维特征向量, 如 LLE^[7]、IsoMap^[8]、Laplacian Eigenmap^[9]等方法。以上算法虽然在小网络上具有较好的效果, 但时间复杂度较高, 在大规模网络上应用困难。近年来, 人们提出用于学习图的卷积神经网络结构。

推荐算法需考虑用户与物品大量交互和物品自身属性的丰富性问题。基于领域的推荐算法相似度表达简单; 基于矩阵分解的推荐算法相似度表达没有考虑到用户与物品的多维特征; 基于协同过滤的推荐算法不能充分提取信息。因此, 利用多维特征提高推荐算法的表示能力已成为需要研究的问题。

本文提出基于网络表示学习的卷积协同过滤推荐算法。将二分网络分成用户与物品两个同质网络, 并在各自网络上利用 GraphSAGE 模型进行训练得到两类节点的嵌入表示。在此基础上, 采用外积运算得到两类节点的关系矩阵, 最终通过卷积神经网络捕捉特征中每一维的交互关系完成推荐任务。

1 本文提出的推荐算法

1.1 GraphSAGE 模型

通过使用 GraphSAGE^[10] 模型完成网络表示学习的任务。GraphSAGE 模型用于监督式学习和非监督式学习, 还可以选择是否使用节点属性进行训练。该方法适于解决外部信息多样的推荐问题, 对于将图加入到新节点时不用重新训练整个模型, 提高算法的泛用性。

GraphSAGE 模型是针对同质图问题进行构建的, 通过节点的属性信息和网络结构信息生成节点的嵌入表示。嵌入表示是每个节点学习各自的聚合函数, 通过该函数聚合节点的邻域信息。该算法的前向传播分为采样、聚合、预测 3 个步骤^[10]。

在聚合和预测时, 每阶邻居节点通过使用不同的函数聚合邻居节点特征, 将目标节点特征表示和邻居节点聚合属性连接后通过非线性变换得到目标节点的更新嵌入表示, 所有节点逐层进行迭代。该算法提出

均值聚合函数 (Mean Aggregator)、LSTM Aggregator、池化聚合函数 (Pooling Aggregator)。均值聚合函数对采样邻居节点特征向量的每个维度求均值, 作为目标节点的特征向量; LSTM Aggregator^[11] 具有较强的数据表达能力, 但对数据顺序敏感。池化聚合函数 (Pooling Aggregator) 通过对目标节点的邻居节点做非线性变换并进行池化操作。GraphSAGE 前向传播算法伪代码如算法 1 所示。

算法 1 GraphSAGE 前向传播算法

输入 图 $G(V, E)$, 节点特征向量 $\{x_v, \forall v \in V\}$, 非线性函数 σ , 节点连接权重矩阵 $W^k, \forall k \in \{1, 2, \dots, K\}$, 聚合深度 K , 聚合函数 AGGREGATE_k, $\forall k \in \{1, 2, \dots, K\}$, 邻居函数 N

输出 节点嵌入表示 z_v

```

1.  $h_v^0 \leftarrow x_v, \forall v \in V$ 
2. for  $k = 1, 2, \dots, K$  do
3. for  $v \in V$  do
4.  $h_{N(v)}^k \leftarrow \text{AGGREGATE}_k(\{h_u^{k-1}, \forall u \in N(v)\})$ 
5.  $h_v^k \leftarrow \sigma(W^k \cdot \text{CONCAT}(h_v^{k-1}, h_{N(v)}^k))$ 
6. end for
7.  $h_v^k \leftarrow h_v^k / \|h_v^k\|_2, \forall v \in V$ 
8. end for
9.  $z_v \leftarrow h_v^K, \forall v \in V$ 

```

其中, $\{h_u^{k-1}, \forall u \in N(v)\}$ 表示在 $k-1$ 层中节点 v 的邻居节点特征, $h_{N(v)}^k$ 表示在第 k 层, 节点 v 的邻居节点特征聚合表示, h_v^k 表示在第 k 层节点 v 的特征。

对于反向传播部分, 采用非监督学习方式。参考 SkipGram 模型, 采用基于图的损失函数使相邻节点有更相似的特征表达, 损失函数如式 (1) 所示:

$$L_G(z_u) = -\exp\left(\sigma\left(z_u^\top z_v\right)\right) - Q \cdot E_{v_n \sim P_n(v)} \exp\left(\sigma\left(-z_u^\top z_{v_n}\right)\right) \quad (1)$$

其中, z_u 为节点 u 通过 GraphSAGE 生成的嵌入表示, 节点 v 为节点 u 在 k 层采样内得到的邻居, σ 为 sigmoid 函数, P_n 为负采样的概率分布, Q 为负样本数目。

1.2 算法的设计与实现

1.2.1 算法的整体设计

对于用户与物品二分网络的问题, 本文算法将二部图分解成物品与物品的同质网络 and 用户与用户的同质网络, 利用 GraphSAGE 模型将用户的网络特征结构和属性特征融合, 得到具有相同维度的嵌入表达。对用户与物品的特征向量进行外积运算, 即通过矩阵表示用户与物品特征每个维度之间的关系。最终通过卷积神经网络提取物品与用户的潜在关系。本文算法流程如图 1 所示。

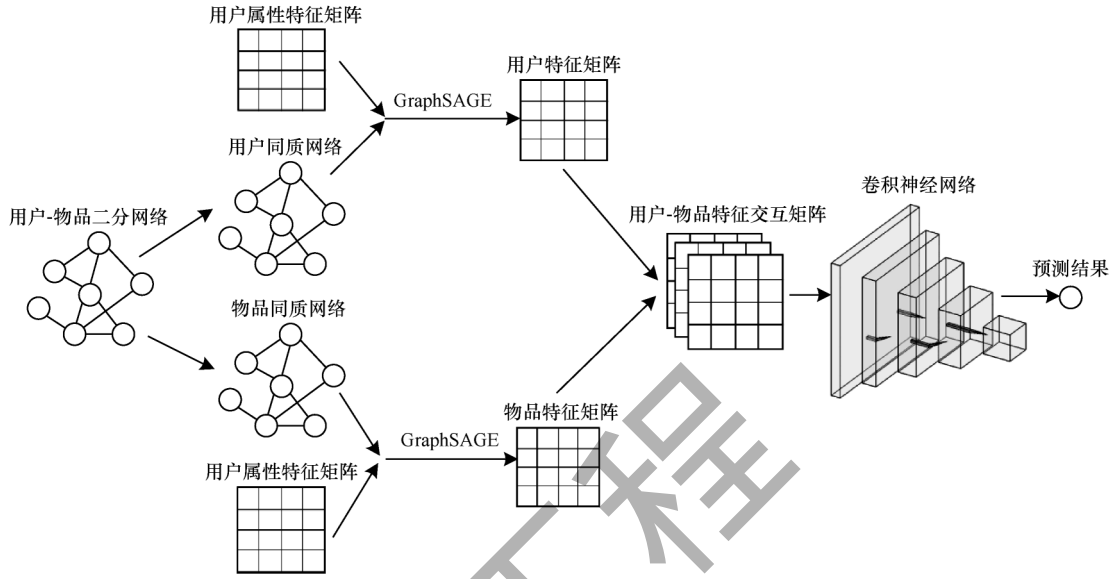


图1 本文推荐算法流程

Fig.1 Procedure of the proposed recommendation algorithm

在用户与物品推荐场景下,通常物品和用户的属性信息和物品与用户的交互信息是已知的。在这种环境下,通过图的结构来表示它们之间的关系。用户和物品节点是图的节点,物品与用户的交互关系是图的连边。通过这些映射就形成了物品与用户的二部图模型。假设推荐场景中包含 m 个用户、 n 个物品,用户节点集合用 $U = \{u_1, u_2, \dots, u_m\}$ 表示,物品节点集合用 $V = \{v_1, v_2, \dots, v_n\}$ 表示。上述推荐问题对应的用户-物品二部图表示为 $G = (U, V, E, W)$, E 为图 G 中所有边的集合, e_{ij} 表示节点 u_i 与 v_j 的连边; W 为图 G 中用户与物品的交互权重矩阵, w_{ij} 为图 G 中 e_{ij} 对应的权重。

同质网络问题包括用户网络和物品网络两部分。以物品网络为例,不同的物品有着相似的适用群体,如乒乓球和球拍有很强的关联性。对于用户网络,如果两个用户都是运动爱好者,则两者有着相似的爱好。因此,同质网络也有着深刻的联系。

将用户-物品二部图分解为两个同质图。对于图 G ,定义用户节点的一阶相似度 w_{ij}^U 为:

$$w_{ij}^U = \sum_{k \in V} w_{ik} w_{jk} \quad (2)$$

定义物品节点的一阶相似度 w_{ij}^V 为:

$$w_{ij}^V = \sum_{k \in U} w_{ki} w_{kj} \quad (3)$$

得到 $|V| \times |V|$ 维的物品相似度矩阵 $W^V = [w_{ij}^V]$ 和 $|U| \times |U|$ 维的用户相似度矩阵 $W^U = [w_{ij}^U]$,根据 W^V 和 W^U 构建用户同质图 G^U 和物品同质图 G^V 。在使用 W^U 、 W^V 构建用户同质图与物品同质图之前,根据 W^U 、 W^V 中权重分布情况适当去掉权重过低的边,避免噪声干扰影响后续计算结果。

构建用户及物品属性特征需考虑用户及物品的属性信息类型。对于结构化数据,将其中的非离散数据离散化,进行 one-hot 编码得到特征编码;对于非结构化数据,若为本文信息常使用 TF-IDF^[12]或 LDA 算法^[13]进行结构化处理,若为音频、视频、图像等信息则使用相对应的深度学习方法转化为结构化数据。通过 GraphSAGE 模型得到的用户属性特征矩阵 A^U 、物品属性特征矩阵 A^V 转换为用户特征矩阵 M^U 与 M^V ,表示如下:

$$M^U = F_{\text{GraphSAGE}}(G^U, A^U) \quad (4)$$

$$M^V = F_{\text{GraphSAGE}}(G^V, A^V) \quad (5)$$

假设得到的嵌入表示维度都为 t ,则可以表示用户特征矩阵 $M^U \in \mathbb{R}^{m \times t}$ 和物品特征矩阵 $M^V \in \mathbb{R}^{n \times t}$ 。

利用外积运算得到用户与物品特征交互矩阵,对于用户 u 与物品 i , m_u^U 表示用户 u 的特征向量, m_i^V 表示物品 i 的特征向量,则 $M_{u,i}$ 的计算公式如下:

$$M_{u,i} = m_u^U \otimes m_i^V = \begin{bmatrix} m_{u1}^U m_{i1}^V & m_{u1}^U m_{i2}^V & \dots & m_{u1}^U m_{it}^V \\ m_{u2}^U m_{i1}^V & m_{u2}^U m_{i2}^V & \dots & m_{u2}^U m_{it}^V \\ \vdots & \vdots & \ddots & \vdots \\ m_{ut}^U m_{i1}^V & m_{ut}^U m_{i2}^V & \dots & m_{ut}^U m_{it}^V \end{bmatrix} \quad (6)$$

在协同过滤中,通常使用矩阵分解表示物品与用户的关系并对物品与用户的关系做内积,只使用 $M_{u,i}$ 中对角线上的信息。因多层感知机(Multilayer Perceptron, MLP)算法在理论上可以拟合任何函数关系,但是需要大量数据进行训练。在用户与物品推荐系统中,每个用户的行为信息是有限的,所以本算法未采用直接拼接用户、物品特征后通过 MLP 进行学习的方案。利用有限数据训练的深层网络会降低其性能,且很难保证 MLP 收敛到真实模型。同时在实验部分,直接拼接用户、物品特征不经过卷积神

经网络训练, 直接通过 MLP 进行对比实验, 进一步说明加入卷积神经网络增强了算法的效果。

本算法使用外积对物品与用户的信息交互进行建模, 并利用卷积神经网络进行训练, 减少了训练所需数据量的同时也减少了模型中需要训练的参数。

1.2.2 卷积神经网络结构设计

卷积神经网络 (Convolutional Neural Network, CNN) 模型中的参数定义比传统模型更复杂, 其参数设计的一般规律可总结为以下 4 个方面:

1) 卷积层一般使用较小的卷积核, 卷积核越大, 输出的特征图越小, 难以提取数据的特征, 而且卷积核越小, 相应的参数也越小。

2) 卷积步长一般设置较小, 便于更好地提取特征。

3) 池化层常使用 2×2 的池化窗口。池化层的作用是对输入数据进行空间降维, 当池化操作过大时, 数据信息易丢失, 最终导致网络性能下降。

4) 全连接层的层数一般不宜超过 3 层, 全连接层数越多, 训练难度越大, 越容易造成过拟合和梯度消散。

本模型的 CNN 部分如图 2 所示。采用常见的卷积网络模型进行设计, 为避免丢失过多的结构信息, 没有使用卷积层和池化层将矩阵数据压成一维。模型由 3 层全连接层和 6 层卷积层组成, 卷积核大小均为 3×3 , 步长为 1×1 。为了使输入输出的特征图大小不变, 将进行填充操作。每 2 层卷积层后加入池化层进行最大值池化, 池化核大小为 2×2 , 步长为 2×2 。对特征图进行下采样, 并在之后 2 个卷积层中将通道数翻倍, 以此类推, 在最后的卷积层添加 Flatten 层将数据压平并连接 MLP 网络, 逐渐将输出维度缩小到一维。整个流程用式 (7) 表示, 其中 \hat{y}_{ui} 为最终神经网络的输出。

$$\hat{y}_{ui} = \sigma(w_n^T (\sigma(w_{n-1}^T (\cdots \sigma(w_1^T \text{CNN}(M_{u,i}) + b_1) \cdots) + b_{n-1})) + b_n) \quad (7)$$

其中, σ 为非线性函数, 除最后一层全连接层为 sigmoid 激活函数外, 其余所有卷积层与全连接层均使用 ReLu 激活函数。总体的模型参数如表 1 所示。

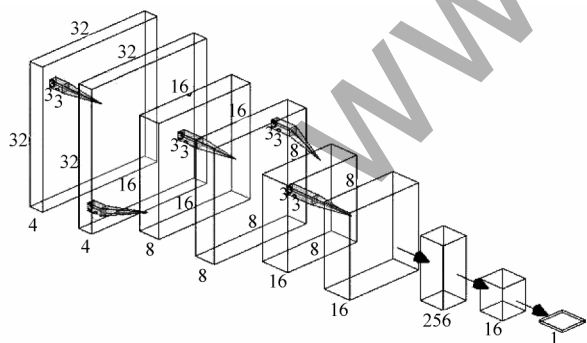


图2 卷积神经网络示意图

Fig.2 Schematic diagram of convolutional neural network

表1 卷积神经网络部分模型参数

Table 1 Convolutional neural network partial model parameter

网络层	结构设置	输出尺寸
卷积层 1	3×3 卷积核, 步长 1, 尺寸相同填充	$32 \times 32 \times 4$
卷积层 2	3×3 卷积核, 步长 1, 尺寸相同填充	$32 \times 32 \times 4$
池化层 1	2×2 池化核, 步长 2, 无填充	$16 \times 16 \times 4$
卷积层 3	3×3 卷积核, 步长 1, 尺寸相同填充	$16 \times 16 \times 8$
卷积层 4	3×3 卷积核, 步长 1, 尺寸相同填充	$16 \times 16 \times 8$
池化层 2	2×2 池化核, 步长 2, 无填充	$8 \times 8 \times 8$
卷积层 5	3×3 卷积核, 步长 1, 尺寸相同填充	$8 \times 8 \times 16$
卷积层 6	3×3 卷积核, 步长 1, 尺寸相同填充	$8 \times 8 \times 16$
全连接层 1	—	256
全连接层 2	—	16
全连接层 3	—	1

1.2.3 损失函数设计

模型损失函数主要是平方损失函数, 定义该函数的前提是观察的结果服从高斯分布。在实际问题中, 用户与物品的交互信息不一定服从高斯分布。本算法采用二分类的思想, 将用户与物品的关系采用“0”和“1”表示, 0 表示不相关, 1 表示相关。 \hat{y}_{ui} 表示卷积网络的输出, 代表预测物品 i 与用户 u 相关的可能性。为了使 \hat{y}_{ui} 具有概率性的含义, 将其取值范围限制在 $[0, 1]$, 因此在神经网络的最后一层激活函数选用 sigmoid 函数, 使用的损失函数如式 (8) 所示:

$$L = - \sum_{(u,i) \in Y \cup Y^-} y_{ui} \exp \hat{y}_{ui} + (1 - y_{ui}) \exp (1 - \hat{y}_{ui}) \quad (8)$$

其中, Y 为正采样集合, Y^- 为负采样集合, y_{ui} 表示用户 u 与物品 i 的联系, \hat{y}_{ui} 表示卷积网络最后的输出。

最终模型训练时采用 Adam 优化算法^[14], 模型在 MLP 部分添加 Dropout 层以解决训练过拟合的问题^[15], 增强模型泛化能力。

2 实验与结果分析

2.1 实验数据集

Movielens-100k 数据集包含用户提供的电影评分数据。该数据集包括 10 万组用户电影评分信息, 评分为 1~5 的整数, 以及用户的性别、年龄等类别标签信息。

Last.fm-2k 数据集包括近 10 万组用户对歌手的收听信息, 用户对某歌手的关系权值是用户对该歌手所有作品的播放次数, 还有 1 万组用户间好友关系, 以及近 20 万组用户对歌手添加的标签信息。Movielens 和 Last.fm 的基本统计信息如表 2 所示。

表2 实验数据集信息统计

Table 2 Experimental data set information statistics

数据集	用户总量	商品总量	交互数	稀疏性/%
Movielens	943	1 682	100 000	93.7
Last.fm	1 892	17 632	92 834	99.7

2.2 对比算法

为验证本文算法的有效性,实验部分选取了具有代表性的 ItemKNN^[16]、BPR^[17]、MF^[18]、NeuMF^[19]、ConvNCF^[20]算法作为对比算法。

2.3 实验设置

实验在 ubuntu-16.04 环境下进行,使用 python-3.6.5 开发,使用的库包主要为 numpy-1.14.3、networkx-2.3、keras-2.0.5、sklearn-0.20.0。

模型训练时需要提供正负样本进行学习,本文对 Movielens 与 Last.fm 数据集进行相同的处理。在排除实验测试集所需正样本后,对考虑每一位用户,假设其包含的正样本数为 n ,设定负采样系数为 ns 。对于该用户将从其没有发生过交互的物品集中随机抽取 $ns \times n$ 个物品作为负样本,保证对于每个用户其正负样本比例都是一样的,以下实验数据中如没有特殊说明 ns 的取值均为 4。将全部样本的 90% 作为训练集,10% 作为验证集。

2.4 实验指标

为验证算法在推荐问题下的性能,选用 HR@k 与 NDCG@k 作为评价指标,以综合衡量算法结果在无序评价指标和有序评价指标下的性能。

1) 召回率

设 $R(u)$ 是测试集上为用户 u 产生的推荐列表, $T(u)$ 是用户发生过交互行为的物品列表。召回率定义为推荐列表中用户最终发生交互的物品在测试集中的占比,计算公式如下:

$$R_c = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (9)$$

R_c 值越大,表示推荐算法的召回率越高。由于 R_c 的值与推荐列表的长度密切相关,因此常写作 $R_c@k$ 以直接表明条件设置,表示为 HR@k。

2) 归一化折损累计增益

归一化折损累计增益(Normalized Discounted Cumulative Gain, NDGG)对推荐结果在列表中的排名增加了惩罚,计算公式如下:

$$D_k = \sum_{i=1}^k \frac{2^{\text{rel}_i} - 1}{\text{lb}(i+1)} \quad (10)$$

其中, rel_i 表示位置 i 推荐结果的相关性, k 表示推荐列表的长度。考虑到评价指标需要衡量推荐算法对不同用户的推荐效果,因此提出 NDCG 评价指标,对用户 u 的 NDCG@k 的计算公式如下:

$$N_u@k = \frac{D_u@k}{I_u} \quad (11)$$

其中, $I@K$ 为推荐算法为某一用户返回的最佳推荐结果列表,最终 N@k 的计算公式为:

$$N@k = \frac{\sum_{u \in U} N_u@k}{|U|} \quad (12)$$

NDCG 的取值范围是 $[0, 1]$, 且越接近 1, 推荐效果越好。

2.5 结果分析

2.5.1 性能分析

以下实验数据均为 5 次独立实验结果的平均值。本文与对比算法在 Movielens 数据集上的结果如表 3 所示,在 Last.fm 数据集上的结果如表 4 所示。根据实验数据可以看出,本文提出的算法较 ItemKNN 算法的性能有显著提升,而基于神经网络的 NeuMF 算法在 Movielens 与 Last.fm 上的表现均比传统方法要好。ConvNCF 算法与本算法都优于其他对比算法,说明利用卷积神经网络进行协同过滤是有效的。与 ConvNCF 算法相比,本文算法在 Movielens 数据集上将 HR@5 提升了 1.89 个百分点、NDCG@5 提升了 2.19 个百分点,在 Last.fm 数据集上将 HR@5 提升了 1.09 个百分点, NDCG@5 提升了 3.32 个百分点。因此,本算法在用户-物品二分网络的推荐问题上可以提升性能。

表 3 对比算法在 Movielens 数据集上的实验结果

Table 3 Experiment results of comparison algorithm on Movielens datasets

算法	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
ItemKNN	0.552 8	0.683 2	0.815 5	0.387 9	0.461 8	0.499 0
BPR	0.634 9	0.768 7	0.868 2	0.433 8	0.506 5	0.549 2
MF	0.628 0	0.761 6	0.864 1	0.431 6	0.502 4	0.545 1
NeuMF	0.638 5	0.775 3	0.876 4	0.446 5	0.514 5	0.557 7
ConvNCF	0.651 7	0.802 4	0.892 1	0.469 8	0.531 5	0.564 9
本文算法	0.664 0	0.816 8	0.901 5	0.480 1	0.546 9	0.573 8

表4 对比算法在 Last.fm 数据集上的实验结果

Table 4 Experiment results of comparison algorithm on Last.fm datasets

算法	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
ItemKNN	0.565 9	0.612 7	0.655 3	0.453 5	0.524 6	0.560 2
BPR	0.683 0	0.741 5	0.777 6	0.538 6	0.572 5	0.585 4
MF	0.673 4	0.738 3	0.776 5	0.532 8	0.563 7	0.583 9
NeuMF	0.689 7	0.746 1	0.784 2	0.541 3	0.582 0	0.600 3
ConvNCF	0.703 4	0.758 3	0.796 5	0.552 8	0.593 7	0.613 9
本文算法	0.711 1	0.770 4	0.810 9	0.565 6	0.613 1	0.634 7

2.5.2 聚合器选择对算法性能的影响分析

在上述2个数据集上,Mean、LSTM、Pooling这3种聚合器的表现对比如图3所示,其中NG是NDCG的缩写。Pooling聚合器的性能最好,Mean聚合器的性能最差,LSTM聚合器的性能与Pooling较为接近。由于LSTM聚合器构成邻居节点序列时具有不确定性,因此LSTM聚合器性能偶尔优于Pooling聚合器性能。LSTM训练学习时间较长,因此,使用Pooling聚合器的效果最佳。

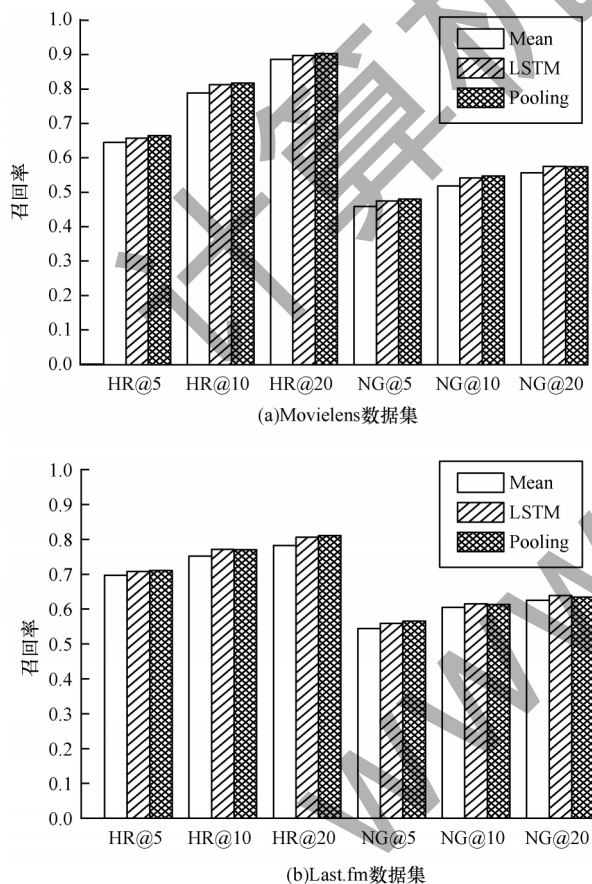


图3 3种聚合器性能对比

Fig.3 Performance comparison of three aggregators

2.5.3 算法收敛性分析

式(7)是本文推荐算法的基本模型,由于是二分类问题,因此可用“0”和“1”表示是否为推荐的物品。选

择sigmoid函数将结果限制在0和1之间得到具体的损失函数,如式(8)通过最小化损失函数训练模型参数。

通过实验验证本模型的收敛性。在训练学习过程中,采用早停法避免过拟合现象的发生。在Movielens数据集上训练500次迭代,实验结果如图4所示。HR@10在训练学习时比NDCG@10更稳定。随着训练次数的增加,两者均表现出稳定的状态,因此本算法的收敛性能良好。

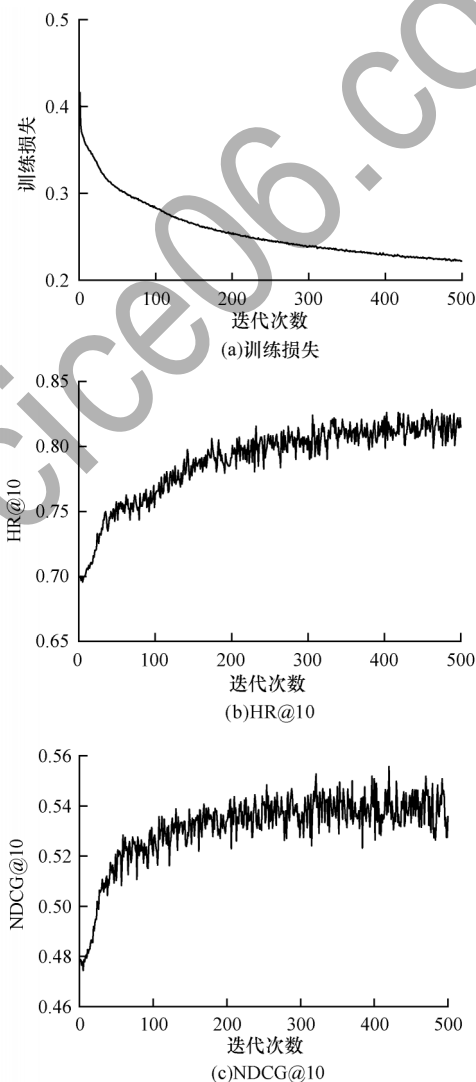


图4 训练损失、HR@10与NDCG@10的训练结果

Fig.4 Training loss, HR@10 and NDCG@10 training results

2.5.4 参数敏感性分析

参数敏感性是衡量算法的指标。本算法主要分析卷积层数对推荐效果的影响。将物品与用户的交互矩阵压缩成一维向量,采用MLP替代卷积神经网络,作为卷积神经网络的消融实验。调整卷积神经网络的层数使输出的特征图尺寸依次为 16×16 、 8×8 、 4×4 、 2×2 、 1×1 ,将其压成一维。根据维度大小设计合适的MLP并完成后续训练。将上述5组实验依次称为Conv1~Conv5,在上述2个数据集上完成的实验结果如图5所示。在协同过滤学习中加入卷积神经网络比单独使用MLP的效果更好。随着卷积层数的增加和特征图尺寸的缩小,HR@10与NDCG@10都存在极值点,呈现先上升后下降的趋势。因此,适当调整网络层数与最终保留特征图的尺寸能获得最有效的信息。

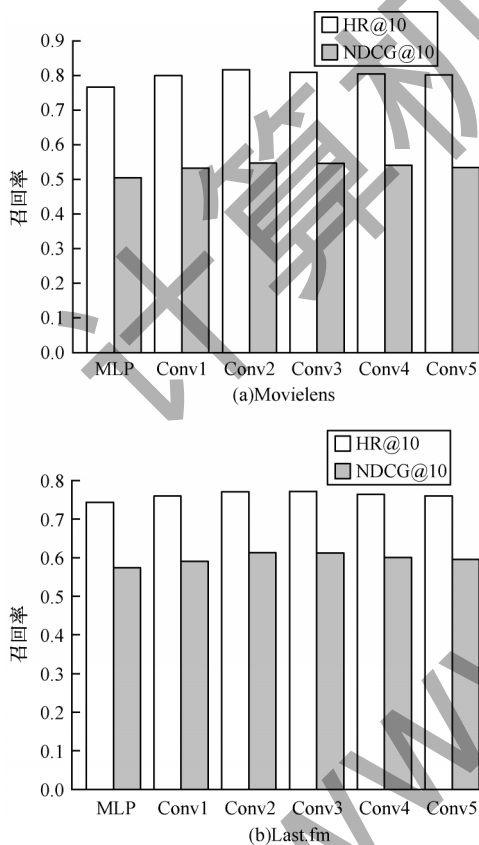


图5 卷积层数的敏感性分析

Fig.5 Sensitivity analysis of convolution layers

本文推荐问题是一种二分类问题,其负采样较灵活,因此需考虑负采样比例对模型性能的影响。设置负采样系数 ns 为2、4、6、8、10。在2个数据集上进行实验实验结果如图6所示。 ns 为4~6较为合理。

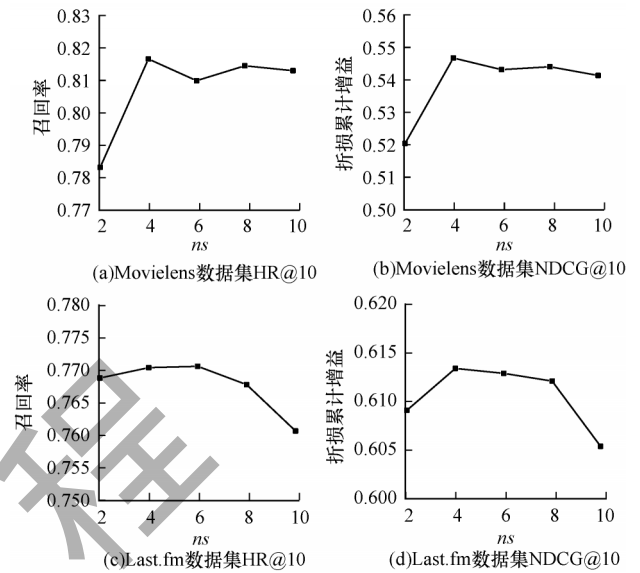


图6 训练集负采样个数的敏感性分析

Fig.6 Sensitivity analysis of the number of negative samples in the training set

用户、物品的节点嵌入维度也会影响模型的效果。本实验将嵌入维度调整为8、16、32、64。实验结果如表5所示,整体上各项评价指标随着维度的增加而增加。增大嵌入维度可以更好地保留有效信息,但当维度过大时,训练参数会急剧增加,从而导致训练时间大大增加。在实际情况下,需要结合需求选择合适的维度。

表5 用户、物品节点嵌入维度的敏感性分析

Table 5 Sensitivity analysis of embedded dimensions of user and item nodes

数据集	评价指标	维度为8	维度为16	维度为32	维度为64
Movielens	HR@10	0.763 1	0.796 2	0.816 8	0.834 4
	NDCG@10	0.502 8	0.531 0	0.546 9	0.558 1
Last.fm	HR@10	0.714 5	0.755 6	0.770 4	0.786 3
	NDCG@10	0.549 2	0.598 4	0.613 1	0.627 0

3 结束语

本文提出基于网络表示学习与深度学习的推荐算法。将二部图模型运用于网络表示学习法,将用户与物品二分网络分解为两个同质网络。通过外积运算表示用户与物品特征每一维的关系矩阵,使用卷积神经网络捕捉特征中每一维的高阶交互关系。与其他算法相比,在数据集上测试的推荐算法召回率和折损率都有相应的提升并具有良好的收敛性。下一步将节点的属性信息考虑进网络表示学习法,并进行对比实验研究。

参考文献

- [1] TU C, LIU H, LIU Z, et al. CANE: context-aware network embedding for relation modeling[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. San Diego, USA: Association for Computational Linguistics, 2017, (1): 1722-1731.
- [2] HAN Q Y. The study of personalized recommendation algorithm in e-commerce system[J]. International Journal of Education and Economics, 2020, 3(2): 1-6.
- [3] HU L M, LI C, SHI C, et al. Graph neural news recommendation with long-term and short-term interest modeling[EB/OL]. [2020-07-05]. <https://arxiv.org/abs/1910.14025?context=stat.ML>.
- [4] ABUL-FOTTOUH D, SONG M Y, GRUZD A. Examining algorithmic biases in YouTube's recommendations of vaccine videos[J]. International Journal of Medical Informatics, 2020, 140: 104385.
- [5] HAVELIWALA T H. Topic-sensitive pagerank: a context-sensitive ranking algorithm for Web search[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 784-796.
- [6] GROVER A, LESKOVEC J. Node2vec: scalable feature learning for networks[C]//Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 855-864.
- [7] SUN X F, GUO J, DING X, et al. A general framework for content-enhanced network representation learning[EB/OL]. [2020-07-03]. <https://arxiv.org/abs/1610.02906>.
- [8] TENENBAUM J B, DE SILVA V, LANGFORD J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000, 290(5500): 2319-2323.
- [9] BELKIN M, NIYOGI P. Laplacian eigenmaps and spectral techniques for embedding and clustering[C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2002: 585-591.
- [10] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2017: 1024-1034.
- [11] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [12] TAS O, KIYANI F. A survey automatic text summarization[C]//Proceedings of the 2nd World Conference on Technology, Innovation and Entrepreneurship. Istanbul, Turkey: Press Academia Procedia, 2007, 5(1): 205-213.
- [13] BLEI D M, NG A Y, JORDAN M I. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3(1): 993-1022.
- [14] KINGMA D P, BA J. Adam: a method for stochastic optimization[EB/OL]. [2020-07-06]. <https://arxiv.org/pdf/1412.6980.pdf>.
- [15] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2012: 1097-1105.
- [16] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on World Wide Web. Berlin, Germany: Springer, 2001: 285-295.
- [17] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback[C]//Proceedings of the 25th International Conference on Uncertainty in Artificial Intelligence. Washington D. C., USA: IEEE Press, 2009: 452-461.
- [18] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.
- [19] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering[C]//Proceedings of the International Conference on World Wide Web. Berlin, Germany: Springer, 2017: 173-182.
- [20] HE X N, DU X Y, WANG X, et al. Outer product-based neural collaborative filtering[EB/OL]. [2020-07-04]. <https://arxiv.org/pdf/1808.03912.pdf>.

编辑 薛晋栋