

## 双通道图协同过滤推荐算法

苗雨欣, 宋春花, 牛保宁, 康瑞雪

(太原理工大学 信息与计算机学院, 太原 030000)

**摘要:** 协同过滤是一种应用广泛的推荐算法, 其核心过程是学习用户和商品的向量表示。基于图卷积网络(GCN)的协同过滤算法在向量嵌入过程中加入邻居节点的关联信息, 进一步提升了算法的推荐性能。然而, 图协同过滤算法中存在过平滑现象, 且其仅采用邻接矩阵在局部结构中扩展, 没有从图的整体结构出发挖掘节点间潜在的交互模式, 使得交互信息来源单一。提出一种基于GCN的双通道协同过滤推荐算法DCCF。将向量嵌入过程划分为局部卷积通道和全局卷积通道, 以获取不同类型的连接信息。在局部卷积通道中, 直接定位邻域节点并使用单层网络结构完成计算, 优化信息的聚合方式以应对过平滑问题。在全局卷积通道中, 通过聚类的方式构造全局交互图并参与信息的聚合过程, 从而挖掘节点间的潜在联系。将局部信息与全局信息相结合, 以获得包含不同类型高阶关系的节点向量表示。在3个公开数据集上进行对比实验, 结果表明, 相较基准算法中性能表现最优的模型, DCCF在归一化折损累计增益和召回率这2个指标上最高分别提升2.8%和5.0%。

**关键词:** 推荐算法; 深度学习; 协同过滤; 图卷积网络; 向量嵌入

开放科学(资源服务)标志码(OSID):



中文引用格式: 苗雨欣, 宋春花, 牛保宁, 等. 双通道图协同过滤推荐算法[J]. 计算机工程, 2022, 48(8): 121-128.

英文引用格式: MIAO Y X, SONG C H, NIU B N, et al. Dual-channel graph collaborative filtering recommendation algorithm[J]. Computer Engineering, 2022, 48(8): 121-128.

## Dual-Channel Graph Collaborative Filtering Recommendation Algorithm

MIAO Yuxin, SONG Chunhua, NIU Baoning, KANG Ruixue

(College of Information and Computer, Taiyuan University of Technology, Taiyuan 030000, China)

**[Abstract]** Collaborative filtering is a widely used recommendation algorithm whose core process is to learn the vector representation of users and goods. The Graph Convolution Network (GCN)-based collaborative filtering algorithm adds the correlation information of neighbor nodes in the vector embedding process, which further improves the algorithm's recommendation performance. However, the graph collaborative filtering algorithm tends to over smooth and only uses the adjacency matrix to expand the local structure. Furthermore, it does not mine the potential interaction patterns between nodes from the overall graph structure, making the interaction information source single. To overcome these limitations, a GCN-based Dual-Channel Collaborative Filtering Recommendation (DCCF) algorithm is proposed in this study. In the proposed algorithm, the vector embedding process is divided into local and global convolution channels to obtain different types of connection information. In the local convolution channel, the neighborhood nodes are directly located and a single-layer network structure is used to complete the calculation. Additionally, information aggregation is optimized to address the over smoothing problem. In the global convolution channel, the global interaction graph is constructed via clustering and participates in the information aggregation process, so as to mine the potential connections between nodes. Local and global information are combined to obtain a node vector representation containing different types of high-order relationships. Comparative experiments are conducted on three public datasets and the results show that the DCCF algorithm has a maximum increase of 2.8% and 5.0% in two indicators, namely, the normalized impairment cumulative gain and recall, respectively, compared with the best-performing model in the benchmark algorithm.

**[Key words]** recommendation algorithm; deep learning; collaborative filtering; Graph Convolutional Network (GCN); vector embedding

DOI: 10.19678/j.issn.1000-3428.0062453

**基金项目:** 国家自然科学基金面上项目(62072326); 山西省重点研发计划项目(201903D421007)。

**作者简介:** 苗雨欣(1996—), 男, 硕士研究生, 主研方向为推荐系统、图卷积网络; 宋春花, 副教授、博士; 牛保宁, 教授、博士; 康瑞雪, 硕士研究生。

**收稿日期:** 2021-08-23 **修回日期:** 2021-10-14 **E-mail:** 479893700@qq.com

## 0 概述

个性化推荐算法作为解决信息过载问题的一种有效手段,一直受到学术界和工业界的关注<sup>[1]</sup>。协同过滤是推荐算法的重要分支,因其在学习用户和商品向量表示方面具有良好的表现而得到广泛应用<sup>[2]</sup>。近年来,图卷积网络(Graph Convolutional Network, GCN)<sup>[3-4]</sup>在推荐方面显示出巨大潜力,利用GCN搭建协同过滤推荐模型成为当前的研究热点<sup>[5]</sup>。

基于GCN的协同过滤模型<sup>[6-8]</sup>将用户和商品都视作交互图中同等地位的节点,在计算其向量嵌入时不仅考虑节点本身的特征,还加入邻居节点的信息,进一步提升了模型学习用户和商品向量嵌入的能力。多层GCN<sup>[9]</sup>作为最具代表性的图卷积网络,广泛应用于协同过滤模型的设计中。基于GCN网络结构设计的图协同过滤模型在聚合用户或商品节点的邻域信息时普遍采用多层网络堆叠的形式来扩展图中节点的邻域。一方面,这样的设计会使模型不可避免地受到GCN结构本身固有的过平滑问题<sup>[10-11]</sup>所带来的影响,GCN结构中的图卷积运算实际上是一种特殊的图拉普拉斯平滑<sup>[10]</sup>,节点的特征会在经过多层图卷积运算之后逐渐趋于相似,彼此间难以区分,过平滑问题造成推荐过程中的偏好特征同质化,最终导致模型性能下降;另一方面,在大规模的交互关系图上,如果采用多层网络结构,模型每次迭代都会使邻域内部的节点数量成倍增长,这种情况下并不能保证邻域内传递的高阶信息都是积极的。此外,基于GCN结构设计的图协同过滤模型采用邻接矩阵代表图上的关系结构参与用户和商品向量表达的更新过程,只在局部范围内进行扩展,而忽视了图整体结构中高阶邻居间的潜在交互关系。

本文提出一种双通道图协同过滤推荐(Dual-Channel Graph Collaborative Filtering Recommendation, DCCF)算法。该算法将用户和商品节点的向量表达更新过程划分为局部卷积通道和全局卷积通道,分别承担不同的信息聚合任务。在局部卷积通道中,DCCF对节点局部邻域的扩展方式作出改进,通过引入转移矩阵<sup>[12]</sup>,在单层网络中直接定位节点的高阶邻域范围,并通过设定转移概率阈值的方式进行信息筛选,从而代替多层网络堆叠的形式以应对过平滑问题。在全局卷积通道中,DCCF基于节点特征进行聚类,构造全局交互图,从整体结构出发对节点间的高阶交互模式进行建模,将其作为对局部信息的补充。DCCF算法通过对局部信息和全局信息的整合,获得包含不同类型高阶关系的节点向量表达,从而提升算法的推荐性能。

## 1 相关工作

在推荐算法领域,协同过滤是一个重要分支,其核心过程是以数字向量的形式代表用户和商品,通过向量的计算完成预测和推荐。传统的协同过滤模型<sup>[13-14]</sup>以用户-商品交互矩阵的行或列作为用户和商品的向量表示,基于预定义的相似度函数计算用户和商品间的相似度以完成推荐。矩阵分解<sup>[15-17]</sup>在传统算法的基础上进行改进,从交互矩阵中提取潜在语义表达,用维度统一的隐向量表示用户和商品。深度学习技术的出

现为协同过滤提供了新的发展方向,NCF(Neural Collaborative Filtering)<sup>[18]</sup>、NAIS(Neural Attentive Item Similarity)<sup>[19]</sup>等模型在向量嵌入和相似度计算的过程中引入全连接网络,进一步提升了算法的推荐性能。但是,上述方法往往只考虑用户和商品本身的特征信息,没有充分利用历史行为中已经产生的交互关系,为此,研究人员构建基于图卷积网络的协同过滤模型<sup>[6-8]</sup>以解决这一问题。

相较传统算法以及其他深度网络模型,基于图卷积网络的协同过滤模型的最大优势在于能够在向量嵌入的过程中添加来自邻居节点的协作信号,从而对图中的高阶关系进行建模。GC-MC(Graph Convolutional Matrix Completion)<sup>[20]</sup>模型将图卷积运算应用于用户-商品二部交互图,将提取到的邻居节点信息作为用户和商品节点的向量表达。WANG等<sup>[6]</sup>提出NGCF(Neural Graph Collaborative Filtering)模型,并通过实验证明了利用图卷积网络对用户和商品之间的关联性进行建模可以提升模型学习用户和商品向量嵌入的能力,同时对交互图中的高阶连接特性作出了解释。GC-MC<sup>[20]</sup>和NGCF<sup>[6]</sup>在向量嵌入过程中引入图卷积网络,增强了模型的表达能力,但过平滑问题的存在会使节点特征在向深层传递的过程中趋于同质化,导致推荐性能难以提高。CHEN等<sup>[8]</sup>对NGCF模型进行简化并引入残差连接,提升了算法的推荐性能。除了过平滑现象之外,交互图的选择也是一个值得关注的问题。JI等<sup>[21]</sup>通过重新定义节点间的交互规则,构建超图<sup>[22-23]</sup>参与节点邻域信息的聚合过程,其不再局限于以邻接矩阵为核心构建图卷积运算,为图协同过滤模型的设计提供了新思路。

在本文基于图卷积网络的双通道协同过滤推荐算法DCCF中,通过多尺度卷积的方式,利用转移矩阵结合转移概率阈值确定节点的邻域构成,然后使用单层网络完成局部信息整合,从而应对过平滑问题。在此基础上,通过聚类的方式构造全局结构上的交互图,以节点特征为基础定义节点间的交互,对潜在的高阶关系进行建模,使得模型能够综合考虑局部和全局这2种不同类型的交互信息。

## 2 DCCF算法

DCCF算法的预测流程如图1所示。模型的输入是用户与商品之间的历史交互记录,为[UserID, [ItemID1, ItemID2, ...]]的形式。首先,从历史数据中导出初始图结构,包括用户与商品之间的二部交互图和图中节点的初始向量表达;然后,为了获取更加准确的用户和商品的向量表达,在向量嵌入阶段设计2个卷积通道,分别承担不同的信息聚合任务,以对节点间的交互关系进行建模;最后,利用向量嵌入阶段得到的用户和商品的特征信息,通过向量内积预测历史数据中未被观测到的交互。

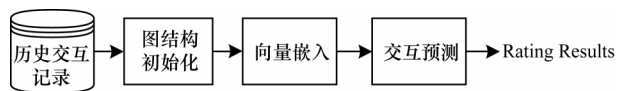


图1 DCCF算法的预测流程

Fig.1 Prediction procedure of the DCCF algorithm

基于上述流程分析,DCCF算法的模型结构如图2所示,主要由3个部分构成:1)局部卷积通道,通过状态转移矩阵直接定位节点的高阶邻域范围,使用一个单层图卷积网络完成用户和商品节点局部信息的聚合;2)全局卷积通道,以聚类的方式构建一个

全局交互图作为图整体结构中不同节点间交互关系的表达,对图整体结构中潜在的交互关系进行建模;3)交互预测,对节点的局部信息和全局信息进行整合,更新用户节点和商品节点的向量表达,通过向量内积的方式预测未知交互。

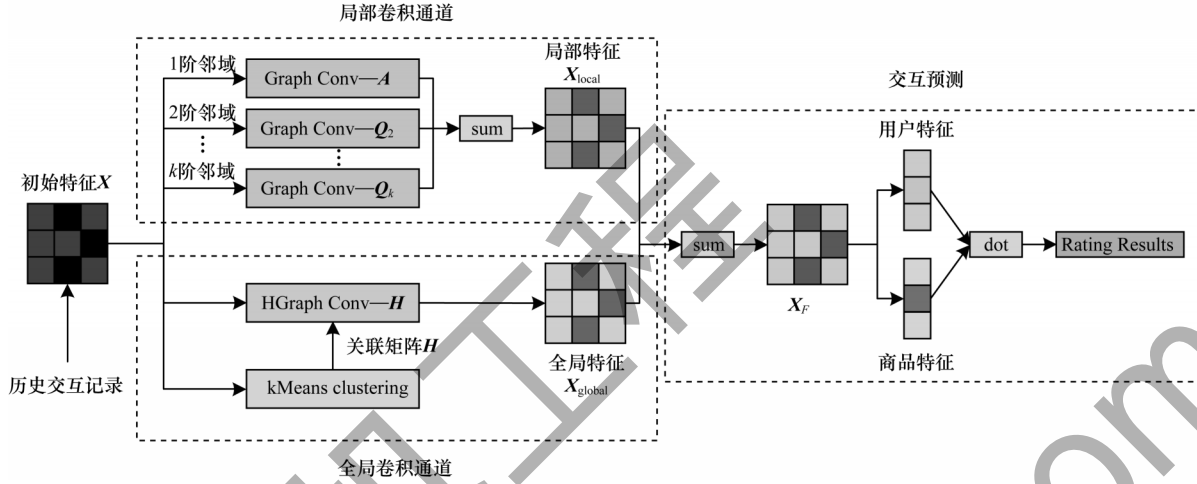


图2 DCCF模型结构

Fig.2 DCCF model structure

## 2.1 图结构初始化

由历史交互数据导出用户和商品间的二部交互图  $G(V, E)$ , 其中:  $V$  代表图中的节点集合, 由用户节点集  $V_{\text{User}}$  和商品节点集  $V_{\text{Item}}$  组成;  $E$  代表图中连接边的集合, 如果用户  $\text{User}_n$  与商品  $\text{Item}_m$  产生过交互(购买、浏览等行为), 则两者之间存在一条连接边  $e_{nm} \in E$  (由于交互图是一个无向图, 因此  $e_{nm} = e_{mn}$ )。在此基础上, 可以得到交互矩阵  $R$ , 其为一个  $N$  行  $M$  列的矩阵,  $N$  代表用户的数量,  $M$  代表商品的数量。对于矩阵中的每一个分量  $[R]_{nm}$ , 如果其对应的边存在, 即  $e_{nm} \in E$ , 那么当前位置的分量  $[R]_{nm} = 1$ , 反之  $[R]_{nm} = 0$ 。进一步地, 交互图的邻接矩阵  $A$  可以表示为:

$$A = \begin{bmatrix} \mathbf{0} & R \\ R^T & \mathbf{0} \end{bmatrix} \quad (1)$$

其中:  $\mathbf{0}$  代表零矩阵;  $T$  代表转置操作。邻接矩阵代表图中节点的交互情况, 但是, 由于在二部交互图中用户与用户之间、商品与商品之间没有直接的交互关系, 因此对应位置的值均为 0。

通过对交互矩阵  $R$  中的行和列进行降维, 分别得到用户节点和商品节点的初始向量表达, 将它们统一到一个固定的维度  $d$ , 如下:

$$X_{\text{User}} = [x_{u_1}, x_{u_2}, \dots, x_{u_n}], x_{u_i} \in \mathbb{R}^d \quad (2)$$

$$X_{\text{Item}} = [x_{i_1}, x_{i_2}, \dots, x_{i_m}], x_{i_i} \in \mathbb{R}^d \quad (3)$$

其中:  $X_{\text{User}}$ 、 $X_{\text{Item}}$  分别代表用户节点和商品节点的初始特征矩阵。

## 2.2 局部卷积通道

局部卷积通道的任务是为图中节点聚合局部邻域信息。以邻接矩阵作为图上关系结构的表达, 归一化后迭代多次是一种常见的邻域扩展方式, 但这种形式的图卷积运算实际上是一种特殊的拉普拉斯平滑<sup>[10]</sup>, 会使用户节点和商品节点的特征在向深层叠加时趋于相似。过平滑现象的存在导致偏好特征

在模型迭代的过程中逐渐趋于同质化, 影响了模型的推荐性能。为了在聚合信息的过程中尽可能地避免过平滑问题所带来的影响, DCCF 模型中引入转移矩阵<sup>[12]</sup>。如果图结构已知, 则图上的转移矩阵可以定义为:

$$P = D^{-1}A \quad (4)$$

其中:  $A$  为二部交互图的邻接矩阵;  $D$  是邻接矩阵的度矩阵(一个对角矩阵, 其对角线上的分量表示与当前顶点直接相邻的节点数量)。转移矩阵的实质是二部交互图上的随机游走, 以节点  $i$  为例, 假设有  $d(i)$  个节点与  $i$  直接相邻, 在不带权的无向图中, 从节点  $i$  出发, 走一步到达其任意一个邻居节点的概率均为  $1/d(i)$ 。一阶转移矩阵  $P$  中的每一个分量  $[P]_{ij}$  代表图中节点  $i$  走一步到达节点  $j$  的转移概率。而对于高阶转移矩阵  $P^k (k \geq 2)$ , 其中的每一个分量  $[P^k]_{ij}$  则代表图中节点  $i$  走  $k$  步到达节点  $j$  的概率。

在局部卷积通道中, 当确定图中节点的邻域范围时, 一阶邻域仍然由邻接矩阵  $A$  来确定。而对于高阶邻域(大于等于二阶), 则以对应的转移矩阵为基础, 设定转移阈值  $P_\theta$ , 以确定节点的各阶邻域范围。如图3所示, 随着用户节点  $u_1$  的邻域逐渐向外扩展, 其局部结构中的信息传递路径也逐渐增加。

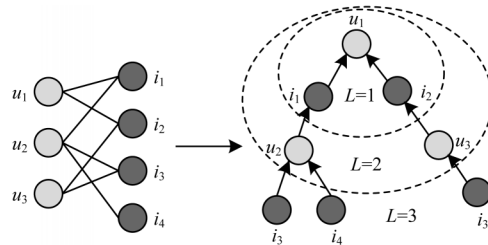


图3 用户-商品交互图和高阶连接

Fig.3 User-item interaction diagram and high order connections



从图3可以看出,当邻域扩展到三阶时,商品 $i_3$ 和用户 $u_1$ 之间有2条信息传递路径,而商品 $i_4$ 和用户 $u_1$ 之间只有1条,这表明用户 $u_1$ 可能对商品 $i_3$ 更加感兴趣。而对应到转移矩阵中,如果2个节点间存在多条信息传递路径,随着邻域的扩展,转移路径不断叠加,其对应的转移概率也就越高。通过设定转移阈值 $P_\theta$ ,在各阶邻域内挑选出具有较高转移概率的邻居节点,从而起到信息筛选的作用。因此,在计算得到各阶转移矩阵之后,如果转移矩阵中的分量 $[P^k]_{ij} > P_\theta$ ,则认为在对应的 $k$ 阶邻域内节点 $i$ 与节点 $j$ 之间存在一条连接边。至此,可以定义高阶邻域内节点之间的交互关系矩阵 $Q_k (k \geq 2)$ :在同阶邻域内,如果转移矩阵对应位置的分量 $[P^k]_{ij} > P_\theta$ ,则有 $[Q_k]_{ij} = 1$ ,反之 $[Q_k]_{ij} = 0$ 。通过上述方式,得到每一个节点不同阶的邻域构成。在每一阶邻域内,以用户节点 $u$ 为例,其对应的邻域信息聚合方式为:

$$\mathbf{x}_{N_{(u),k}} = \frac{1}{\sqrt{|N_{(u),k}|}} \sum_{i \in N_{(u),k}} \frac{1}{\sqrt{|N_{(i),k}|}} \mathbf{x}_i \quad (5)$$

其中: $k$ 代表邻域阶数; $N_{(u),k}$ 代表节点的 $k$ 阶邻域; $|N_{(u),k}|$ 代表节点在当前邻域内邻居节点的数目。然后对各阶邻域信息进行整合得到节点的局部邻域特征:

$$\mathbf{x}_{N_{(u)}} = \frac{1}{k} (\mathbf{x}_{N_{(u),1}} + \mathbf{x}_{N_{(u),2}} + \dots + \mathbf{x}_{N_{(u),k}}) \quad (6)$$

则局部卷积通道的输出为:

$$\mathbf{x}_u^{\text{local}} = \mathbf{x}_u + \mathbf{x}_{N_{(u)}} \quad (7)$$

其中: $\mathbf{x}_u$ 是用户节点 $u$ 自身的特征; $\mathbf{x}_{N_{(u)}}$ 是局部邻域特征。其他用户以及商品的特征更新过程也都遵循上述步骤,同时,为了方便数据的批量处理,本文给出局部卷积通道中计算过程的矩阵形式:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{\text{User}} \\ \mathbf{X}_{\text{Item}} \end{bmatrix} \quad (8)$$

$$\mathbf{X}_N = \frac{1}{k} (\mathbf{L}\mathbf{X} + \mathbf{L}_2\mathbf{X} + \dots + \mathbf{L}_k\mathbf{X}) \quad (9)$$

$$\mathbf{X}_{\text{local}} = \mathbf{X} + \mathbf{X}_N \quad (10)$$

其中: $\mathbf{X}$ 为图中节点的初始特征矩阵; $\mathbf{X}_N$ 为局部邻域特征矩阵; $\mathbf{X}_{\text{local}}$ 为图中节点的局部特征矩阵; $\mathbf{L}$ 、 $\mathbf{L}_k$ 分别代表邻接矩阵 $\mathbf{A}$ 和高阶关系矩阵 $\mathbf{Q}_k$ 的拉普拉斯矩阵。 $\mathbf{L}$ 、 $\mathbf{L}_k$ 计算如下:

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \quad (11)$$

$$\mathbf{L}_k = \mathbf{D}_k^{-\frac{1}{2}} \mathbf{Q}_k \mathbf{D}_k^{-\frac{1}{2}} \quad (12)$$

其中: $\mathbf{D}$ 、 $\mathbf{D}_k$ 为相应的度矩阵。 $\mathbf{L}$ 和 $\mathbf{L}_k$ 对应位置的分量计算如下:

$$[\mathbf{L}]_{ui} = \frac{1}{\sqrt{|N_{(u),1}|} \sqrt{|N_{(i),1}|}} \quad (13)$$

$$[\mathbf{L}_k]_{ui} = \frac{1}{\sqrt{|N_{(u),k}|} \sqrt{|N_{(i),k}|}} \quad (14)$$

### 2.3 全局卷积通道

在全局卷积通道中,以节点特征为基础,通过聚类的方式构造一个全局交互图 $G_{\text{global}}$ 来代表所有节

点在整体结构中的交互关系,并利用由 $G_{\text{global}}$ 导出的关联矩阵 $\mathbf{H}$ 构建图卷积运算,以完成全局卷积通道中的信息聚合任务,挖掘节点间潜在的交互关系。

在全局交互图 $G_{\text{global}}$ 中,产生交互关系的2个节点既可能属于同一局部结构,也可能在空间距离上相隔较远,体现了节点的全局属性,而不再局限于历史数据中已经被观测到的成对的二部图连接方式。同时,聚合来自特征表现相似的邻居节点的协作信号,能够加强高阶信息的积极性。为了在构造交互图的过程中体现节点的结构性,在构图之前为所有节点添加其邻域信息,即将节点本身的特征与一阶邻域特征进行拼接,如下:

$$\mathbf{X}_c = \text{concat}(\mathbf{X}, \mathbf{L}\mathbf{X}) \quad (15)$$

然后以携带一阶邻域信息的节点特征 $\mathbf{X}_c$ 为基础,通过kMeans聚类算法对所有节点进行再分类,如果2个节点被划分到同一类别中,则在2个节点间添加一条连接边,由此得到全局结构上的交互图 $G_{\text{global}}$ 。上述过程具体描述如算法1所示。

#### 算法1 全局交互图 $G_{\text{global}}$ 构造算法

输入 节点特征 $\mathbf{X}_c$

输出 全局交互图 $G_{\text{global}}$

1. 以节点特征 $\mathbf{X}_c$ 为基础,利用kMeans聚类算法将所有用户节点和商品节点进行再分类,得到各个类别的聚类中心 $\mathbf{C}.\text{centers}$

2. for each node in  $\mathbf{X}_c$ :

3. 以当前节点的特征向量 $\mathbf{X}_c[\text{node}]$ 为基础,计算当前节点与各个聚类中心点 $\mathbf{C}.\text{centers}$ 之间的欧氏距离 $D$

4. 当前节点与哪一类聚类中心的距离最近,则当前节点属于该类

5. 在当前节点的全局邻域 $G_{\text{global}}[\text{node}]$ 内添加相应类别的邻居节点

6. end for

7. return  $G_{\text{global}}$

在完成上述算法流程后,为了保证全局邻域内传递的高阶信息的积极性,需要对全局交互图 $G_{\text{global}}$ 中所有节点的邻域进行二次筛选,为每一个节点在其所属类别内部选择固定数量的特征表现最为相近的邻居节点。同样以用户节点 $u$ 为例,其在全局邻域内的信息聚合方式为:

$$\mathbf{x}_{N_{(u),\text{global}}} = \frac{1}{\sqrt{|N_{(u),\text{global}}|}} \sum_{i \in N_{(u),\text{global}}} \frac{1}{\sqrt{|N_{(i),\text{global}}|}} \mathbf{x}_i \quad (16)$$

其中: $\mathbf{x}_{N_{(u),\text{global}}}$ 是节点 $u$ 的全局邻域特征; $N_{(u),\text{global}}$ 代表节点的全局邻域。将节点的全局邻域特征与原始特征相加得到全局卷积通道的输出 $\mathbf{x}_u^{\text{global}}$ :

$$\mathbf{x}_u^{\text{global}} = \mathbf{x}_u + \mathbf{x}_{N_{(u),\text{global}}} \quad (17)$$

这里同样给出全局卷积通道计算过程的矩阵形式,由交互图 $G_{\text{global}}$ 导出关联矩阵 $\mathbf{H}$ (代表节点在全局结构中的交互情况),全局卷积通道的特征更新过程如下:

$$\mathbf{X}_{\text{global}} = \mathbf{X} + \mathbf{L}_H \mathbf{X} \quad (18)$$

$$\mathbf{L}_H = \mathbf{D}_H^{-\frac{1}{2}} \mathbf{H} \mathbf{D}_H^{-\frac{1}{2}} \quad (19)$$

其中: $\mathbf{X}_{\text{global}}$ 为图中节点的全局特征矩阵; $\mathbf{L}_H$ 为关联矩阵 $\mathbf{H}$ 的拉普拉斯矩阵; $\mathbf{D}_H$ 为相应的度矩阵,对应位置分量与式(16)中相同。

## 2.4 交互预测

在完成局部卷积通道和全局卷积通道的特征更新过程后,分别得到节点的局部特征和全局特征,将这2个部分的信息相加得到节点的最终向量表达:

$$\mathbf{x}_u^F = \mathbf{x}_u^{\text{local}} + \mathbf{x}_u^{\text{global}} \quad (20)$$

矩阵形式为:

$$\mathbf{X}_F = \mathbf{X}^{\text{local}} + \mathbf{X}^{\text{global}} \quad (21)$$

此时,用户  $u$  对商品  $i$  的评分可以通过向量表达  $\mathbf{x}_u^F$ 、 $\mathbf{x}_i^F$  之间的内积来进行预测:

$$\hat{y}_{\text{DCCF}}(u, i) = (\mathbf{x}_u^F)^T \mathbf{x}_i^F \quad (22)$$

## 2.5 模型优化

本文模型中的各项参数使用成对BPR损失函数进行优化,通过最大化正样本和负样本之间差距的方式来完成训练:

$$\text{loss}_{\text{BPR}} = \sum_{(u, i, j) \in O} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \quad (23)$$

其中:  $\sigma(\cdot)$  代表 Sigmoid 激活函数;  $O = \{(u, i, j) | (u, i) \in O^+, (u, j) \in O^-\}$  代表训练数据,既包含正样本  $O^+$  (即真实发生过的交互)也包含负样本  $O^-$  (即虚构的交互)。正、负2种样本按照1:1的比例设置数量,在每一个训练周期内,对于训练数据集中的每一个用户,正样本是所有与该用户产生过交互的商品集合,负样本则从没有与该用户产生过交互的商品集合中随机挑选,数量与正样本相同。使用Adam优化器来优化模型并更新参数。

## 3 实验结果与分析

### 3.1 实验设置

#### 3.1.1 数据集

为了验证DCCF模型的性能表现,本文选择3个公开的常用于推荐算法评价的数据集进行实验。对于每个数据集,选取其中80%的样本作为训练集,剩下20%作为测试集,数据集信息如表1所示。

表1 数据集信息

Table 1 Datasets information

数据集	用户数	商品数	交互数	稀疏度
MovieLens-100K	943	1 682	100 000	0.063 05
Gowalla	29 858	40 981	1 027 370	0.000 84
Yelp2018	31 668	38 048	1 561 406	0.001 30

#### 3.1.2 对比模型

本文实验选择以下基于不同网络架构的模型进行对比:

1)BPR-MF<sup>[16]</sup>:利用BPR损失函数进行优化的矩阵分解模型,以固定维度的向量嵌入代表商品和用户。

2)DMF<sup>[24]</sup>:通过多层全连接网络从交互矩阵中为用户和商品学习维度统一的向量表达。

3)GC-MC<sup>[20]</sup>:将用户与商品之间的历史交互组织为二部图的形式,使用单层图卷积网络生成用户和商品的向量表达。

4)NGCF<sup>[6]</sup>:基于GCN设计的协同过滤模型,在向量嵌入阶段使用多层图卷积网络获取信息。

5)LR-GCCF<sup>[8]</sup>:在NGCF的基础上进行简化,去

掉激活函数和线性变换并引入残差连接。

#### 3.1.3 评价指标

为了对本文模型在topK预测和偏好排序任务中的表现进行评价,选取归一化折损累计增益(NDCG@K)和召回率(Recall@K)这2个被广泛应用于推荐算法评估的指标。

NDCG@K对模型返回的候选物品列表进行评价,计算如下:

$$\text{DCG}@K = \sum_{i=1}^K \frac{r(i)}{\text{lb}(i+1)} \quad (24)$$

$$\text{NDCG}@K = \frac{\text{DCG}@K}{\text{IDCG}@K} \quad (25)$$

其中:DCG@K是考虑排序顺序因素后推荐列表内商品的得分总和;IDCG@K为归一化参数,是DCG@K能达到的最大值。NDCG@K的计算规则为:候选物品与当前用户的相关性越高则得分越高,高相关性候选物品在列表中的位置越靠前则得分越高。

召回率Recall@K代表真实产生的交互中模型预测正确的比例,计算如下:

$$\text{Recall}@K = \frac{T_p}{T_p + F_n} \quad (26)$$

其中: $T_p$ 是将正样本预测为正样本的数量; $F_n$ 是将正样本预测为负样本的数量。

本文设置K=10,利用测试集中所有用户的平均指标结果代表模型的训练结果。

#### 3.1.4 实验环境及参数设置

实验硬件配置为Intel® Xeon® Bronze 3104处理器,RTX2080ti显卡,512 GB内存,操作系统是Windows10。模型基于深度学习框架PyTorch1.4.0,通过Python语言实现。在对比实验中,采用统一的64维向量来表达用户和商品。对于所有模型,使用BPR-Loss优化各项参数,训练批次大小固定为1 024。其他参数方面,局部卷积通道中的转移概率阈值 $P_o$ 设定为0.5,全局卷积通道中3个数据集上的聚类数目分别为[20,100,100],学习率在[0.000 1,0.000 5,0.001,0.005]中选择。对于较为重要的超参数,局部卷积通道中在最高邻域阶数分别为[1,2,3,4]时进行实验,全局卷积通道中邻域节点数目在[10,20,30,40,50]中选择。

#### 3.1.5 模型实现

DCCF模型按照第2节中提供的矩阵计算形式实现,模型中涉及的邻接矩阵、各阶转移矩阵以及全局交互图关联矩阵均使用稀疏矩阵的形式存放,使用scipy.sparse计算包实现,而不是完整的数据矩阵,以减少内存和计算开销。在模型训练过程中,PyTorch框架也支持将稀疏矩阵转化为稀疏张量后进行迭代计算。同时,DCCF模型在更新特征的过程中并没有使用激活函数和线性变换,训练参数量相比NGCF模型大幅降低,每一个epoch训练时间低于NGCF的一半。在本文设计的全局卷积通道中,采用聚类的方式构建节点的全局邻域,其中,聚类算法的空间复杂度开销为 $O(d(N+M+K))$ , $d$ 为向量嵌入的维度, $N+M$ 为节点总数, $K$ 为聚类数目。聚类所需的空

度计算的空间复杂度开销  $O((N+M)^2)$ ,特别是在大规模数据集下,维护完整的节点关系矩阵需要大量内存和显存资源。

3.2 结果分析

3.2.1 对比实验

表2和表3是在3个数据集上分别以 Recall@10 和 NDCG@10 作为评价指标进行对比实验的结果,其中,DCCF 模型的结果加粗标注,“—”代表基准算法中表现最好的模型,“提升”一项则表示 DCCF 模型相对于基准算法中表现最好的模型的性能提升。从中可以看出,本文 DCCF 模型在所有数据集上均能取得最佳性能表现,相较基准算法中表现最好的模型,DCCF 在 Recall@10 和 NDCG@10 两个指标上最高提升 2.8% 和 5.0%,性能提升验证了 DCCF 对于推荐任务的有效性。GC-MC 的性能表现整体上优于 MF 和 DMF,说明在向量嵌入的过程中添加来自邻居节点的协作信号可以提升模型的推荐能力。NGCF 的性能表现始终优于 GC-MC,原因是 GC-MC 只考虑节点的一阶邻域,而 NGCF 通过多层网络堆叠对局部结构中的可达路径进行建模和表达。LR-GCCF 在 NGCF 的基础上进行简化,去掉激活函数和线性变换,并在每一层增加额外的残差连接,改善了模型表现。DCCF 在不同数据集上的表现始终优于 LR-GCCF,表明 DCCF 通过构建不同的图卷积运算引入多种类型的高阶关系信息,能进一步提升算法的推荐性能。

表2 模型的 Recall@10 对比结果

Table 2 Recall@10 comparison results of models

模型	Recall@10		
	MovieLens-100K	Gowalla	Yelp2018
BPR-MF	0.223 1	0.098 0	0.027 3
DMF	0.204 1	0.069 1	0.023 7
GC-MC	0.206 5	0.098 6	0.027 5
NGCF	0.227 2	0.101 3	0.030 2
LR-GCCF	<u>0.230 3</u>	<u>0.113 5</u>	<u>0.032 0</u>
DCCF	<b>0.234 4</b>	<b>0.116 7</b>	<b>0.032 3</b>
提升	1.8%	2.8%	0.9%

表3 模型的 NDCG@10 对比结果

Table 3 NDCG@10 comparison results of models

模型	NDCG@10		
	MovieLens-100K	Gowalla	Yelp2018
BPR-MF	0.681 7	0.258 6	0.102 1
DMF	0.649 4	0.189 5	0.091 8
GC-MC	0.683 4	0.265 4	0.103 7
NGCF	0.687 2	0.271 4	0.112 4
LR-GCCF	<u>0.689 8</u>	<u>0.284 0</u>	<u>0.115 7</u>
DCCF	<b>0.694 5</b>	<b>0.298 3</b>	<b>0.119 6</b>
提升	0.7%	5.0%	3.4%

3.2.2 消融实验

GC-MC<sup>[20]</sup>和 NGCF<sup>[6]</sup>在其模型设计中保留了线性变换矩阵和非线性激活函数这2种结构,但对于

协同过滤任务而言,这2种结构并不一定有效,甚至会对算法性能造成负面影响<sup>[7-8]</sup>。为了保证 DCCF 模型的推荐性能,对于上述2种结构的不同组合进行对比实验,结果如表4、表5所示,其中:n代表非线性激活函数,这里使用 LeakyReLU;w代表线性变换。

表4 不同结构下的 Recall@10 对比结果

Table 4 Recall@10 comparison results under different structures

模型	Recall@10		
	MovieLens-100K	Gowalla	Yelp2018
DCCF+w	0.229 7	0.112 6	0.029 8
DCCF+n	0.230 3	0.114 7	0.030 7
DCCF+w+n	0.228 9	0.113 9	0.030 2
DCCF	<b>0.234 4</b>	<b>0.116 7</b>	<b>0.032 3</b>

表5 不同结构下的 NDCG@10 对比结果

Table 5 NDCG@10 comparison results under different structures

模型	NDCG@10		
	MovieLens-100K	Gowalla	Yelp2018
DCCF+w	0.690 9	0.283 3	0.112 9
DCCF+n	0.692 5	0.292 5	0.113 2
DCCF+w+n	0.687 5	0.290 3	0.111 7
DCCF	<b>0.694 5</b>	<b>0.298 3</b>	<b>0.119 6</b>

从表4、表5可以看出:在 MovieLens-100K 数据集上,同时添加激活函数和线性变换结构后,模型性能表现最差;在 Gowalla 数据集上,只添加线性变换结构后,模型性能表现最差;相较其他结构组合,去掉线性变换矩阵和激活函数后,DCCF 的性能表现在所有数据集上达到最佳。

为了验证本文 DCCF 模型的局部卷积通道是否能够有效聚合信息,以及全局卷积通道对模型性能的影响,将去掉全局卷积通道后的模型 DCCF\_Local 与 NGCF、DCCF 进行对比,结果如表6、表7所示。

表6 Recall@10 消融实验结果

Table 6 Recall@10 ablation experiment results

模型	Recall@10		
	MovieLens-100K	Gowalla	Yelp2018
NGCF	0.227 2	0.101 3	0.030 2
DCCF_Local	0.229 6	0.107 9	0.030 4
DCCF	<b>0.233 3</b>	<b>0.116 7</b>	<b>0.032 3</b>

表7 NDCG@10 消融实验结果

Table 7 NDCG@10 ablation experiment results

模型	NDCG@10		
	MovieLens-100K	Gowalla	Yelp2018
NGCF	0.687 2	0.271 4	0.112 4
DCCF_Local	0.691 0	0.285 2	0.113 1
DCCF	<b>0.694 5</b>	<b>0.298 3</b>	<b>0.119 6</b>



从表6、表7可以看出:只保留局部卷积通道的模型DCCF\_Local在整体性能表现上优于NGCF,表明本文在局部邻域信息聚合方式上的改进具有有效性;在局部卷积通道的基础上添加全局信息后,模型性能表现再一次得到提升,说明在向量嵌入过程中融合局部特征和全局特征这2种来源不同的高阶信息,可以提升模型的表达能力。

3.2.3 超参数灵敏度分析

为了使DCCF的实验性能取得最佳,针对局部

通道中的最高邻域阶数以及全局通道中的邻居节点数这2个重要的超参数,在不同取值下进行参数灵敏度实验。首先,设定全局卷积通道中邻居节点数为20,调整局部卷积通道中的最高邻域阶数,在3个数据集上进行对比实验,结果如表8所示,从整体实验结果来看,模型的性能表现随着邻域阶数的增加而逐渐提升,除MovieLens-100K数据集外,DCCF在最高邻域阶数为3时取得最佳性能。

表8 不同邻域阶数下的实验结果

Table 8 Experimental results under different neighborhood orders

邻域阶数	Recall@10			NDCG@10		
	MovieLens-100K	Gowalla	Yelp2018	MovieLens-100K	Gowalla	Yelp2018
1	0.229 4	0.115 6	0.032 0	0.679 1	0.297 3	0.118 4
2	0.230 0	0.116 1	0.032 0	0.684 3	0.297 6	0.118 8
3	<b>0.232 2</b>	<b>0.116 3</b>	<b>0.032 2</b>	0.688 6	<b>0.297 9</b>	<b>0.119 6</b>
4	0.231 3	0.116 0	0.031 9	<b>0.694 2</b>	0.297 5	0.119 0

根据上述结果,设置局部卷积通道的最高邻域阶数为3,调整全局卷积通道中的全局邻域节点数进行实验,结果如图4所示。从图4可以看出:在3个数据集上,对于评价指标Recall@10,DCCF在邻居节点数分别为40、30、30时表现最佳;对于评价指标NDCG@10,

DCCF在邻居节点数分别为40、30、20时表现最佳。实验结果表明,全局邻域内邻居节点数取值为30左右时模型性能表现较好,这也说明全局邻域内传递的高阶信息并非越多越好,过度的邻域扩展可能会聚合一些不必要的邻居噪声,从而导致模型性能下降。

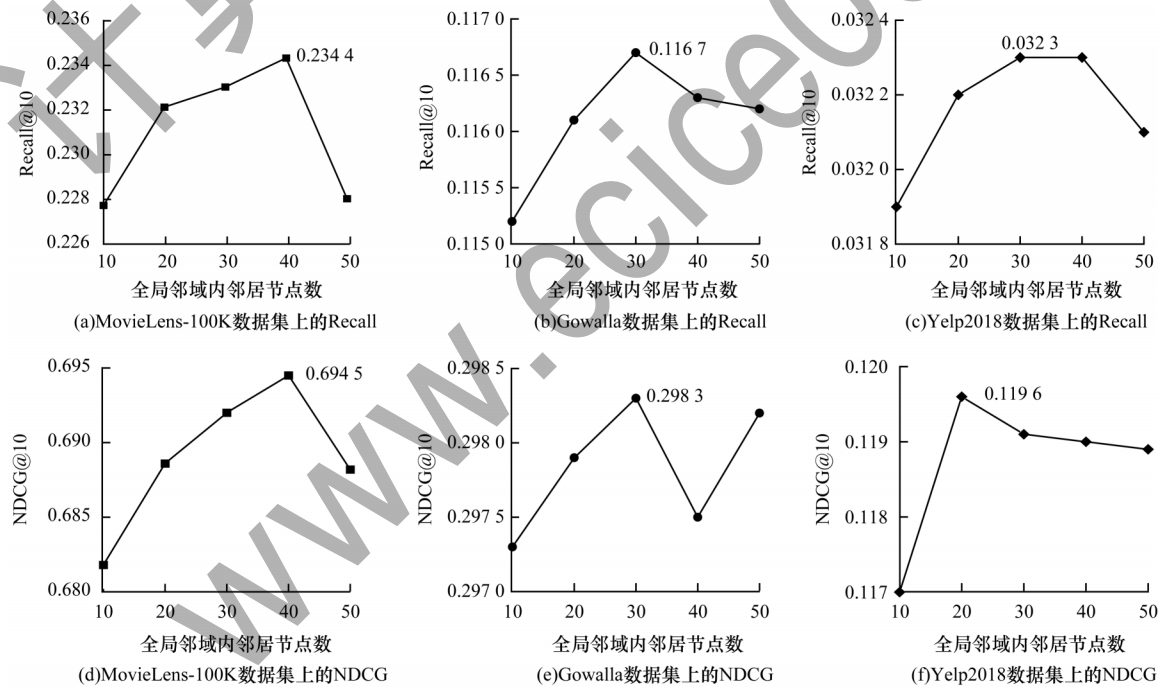


图4 不同邻居节点数下的模型性能表现

Fig.4 Performance of the model under different number of neighbor nodes

4 结束语

本文将图卷积网络与协同过滤推荐算法相结合,提出一种双通道协同过滤推荐算法DCCF,该算法利用2个不同的卷积通道分别聚合局部和全局关联信息。在局部卷积通道中,DCCF结合转移矩阵

以及转移阈值来确定用户节点和商品节点的局部邻域构成,使用单层网络完成信息聚合任务。在全局卷积通道中,DCCF以聚类的方式构造全局交互图,并以此为基础构建图卷积运算,从而聚合特征表现相似的节点信息。将上述局部信息和全局信息进行整合,作为用户节点和商品节点的最终向量表示形

式。实验结果表明,DCCF算法的推荐性能优于BPR-MF、DMF等基准算法。

下一步将从注意力机制和超图这2个角度对DCCF推荐算法进行改进:引入注意力机制,设计相应的关联强度计算方法<sup>[25-26]</sup>,在确定节点的邻域范围后有区别地聚合信息,从而提升推荐性能;在用户和商品过往交互的基础上,通过设计不同的自定义规则来构建超图,针对不同的交互关系进行细粒度建模,从而提升模型的可解释性。

### 参考文献

- [1] 黄立威,江碧涛,吕守业,等. 基于深度学习的推荐系统研究综述[J]. 计算机学报,2018,41(7):1619-1647.  
HUANG L W,JIANG B T,LÜ S Y,et al. Survey on deep learning based recommender systems[J]. Chinese Journal of Computers,2018,41(7):1619-1647. (in Chinese)
- [2] 刘军,杨军,宋姗姗. 基于用户购买意愿力的协同过滤推荐算法[J]. 吉林大学学报(理学版),2021,59(6):1432-1438.  
LIU J,YANG J,SONG S S. Collaborative filtering recommendation algorithm based on purchasing intention of users[J]. Journal of Jilin University (Science Edition),2021,59(6):1432-1438. (in Chinese)
- [3] 徐冰冰,岑科廷,黄俊杰,等. 图卷积神经网络综述[J]. 计算机学报,2020,43(5):755-780.  
XU B B,CEN K T,HUANG J J,et al. A survey on graph convolutional neural network [J]. Chinese Journal of Computers,2020,43(5):755-780. (in Chinese)
- [4] ZHOU J,CUI G Q,HU S D,et al. Graph neural networks: a review of methods and applications[J]. AI Open,2020,1:57-81.
- [5] WU S W,ZHANG W T,SUN F,et al. Graph neural networks in recommender systems: a survey [EB/OL]. [2021-07-01]. <https://arxiv.org/abs/2011.02260>.
- [6] 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.
- [7] 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.
- [8] CHEN L,WU L,HONG R C,et al. Revisiting graph based collaborative filtering: a linear residual graph convolutional network approach[J]. Proceedings of the AAAI Conference on Artificial Intelligence,2020,34(1):27-34.
- [9] KIPF T N,WELLING M. Semi-supervised classification with graph convolutional networks[EB/OL]. [2021-07-01]. <https://arxiv.org/abs/1609.02907>.
- [10] LI Q M,HAN Z C,WU X M. Deeper insights into graph convolutional networks for semi-supervised learning [EB/OL]. [2021-07-01]. <https://arxiv.org/abs/1801.07606>.
- [11] HUANG W B,RONG Y,XU T Y,et al. Tackling over-smoothing for general graph convolutional networks[EB/OL]. [2021-07-01]. <https://arxiv.org/abs/2008.09864>.
- [12] HECHTLINGER Y,CHAKRAVARTI P,QIN J. A generalization of convolutional neural networks to graph-structured data[EB/OL]. [2021-07-01]. <https://arxiv.org/abs/1704.08165>.
- [13] HERLOCKER J L,KONSTAN J A,BORCHERS A,et al. An algorithmic framework for performing collaborative filtering[J]. ACM SIGIR Forum,2017,51(2):227-234.
- [14] 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. New York,USA: ACM Press,2001:285-295.
- [15] KOREN Y,BELL R,VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer,2009,42(8):30-37.
- [16] RENDLE S,FREUDENTHALER C,GANTNER Z,et al. BPR: Bayesian personalized ranking from implicit feedback [EB/OL]. [2021-07-01]. <https://arxiv.org/abs/1205.2618>.
- [17] 贾俊杰,刘鹏涛,陈旺虎. 融合社交信息的矩阵分解改进推荐算法[J]. 计算机工程,2021,47(9):97-105.  
JIA J J,LIU P T,CHEN W H. Improved matrix factorization algorithm using social information for recommendation[J]. Computer Engineering,2021,47(9):97-105. (in Chinese)
- [18] HE X N,LIAO L Z,ZHANG H W,et al. Neural collaborative filtering [C]//Proceedings of the 26th International Conference on World Wide Web. New York,USA: ACM Press,2017:173-182.
- [19] HE X N,HE Z K,SONG J K,et al. NAIS: neural attentive item similarity model for recommendation [J]. IEEE Transactions on Knowledge and Data Engineering,2018,30(12):2354-2366.
- [20] BERG R,KIPF T N,WELLING M. Graph convolutional matrix completion [C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York,USA: ACM Press,2018:1-9.
- [21] JI S Y,FENG Y F,JI R R,et al. Dual channel hypergraph collaborative filtering [C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York,USA: ACM Press,2020:2020-2029.
- [22] FENG Y F,YOU H X,ZHANG Z Z,et al. Hypergraph neural networks[J]. Proceedings of the AAAI Conference on Artificial Intelligence,2019,33:3558-3565.
- [23] JIANG J,WEI Y,FENG Y,et al. Dynamic hypergraph neural networks[EB/OL]. [2021-07-01]. <https://www.ijcai.org/Proceedings/2019/0366.pdf>.
- [24] XUE H J,DAI X,ZHANG J,et al. Deep matrix factorization models for recommender systems[EB/OL]. [2021-07-01]. <https://www.ijcai.org/Proceedings/2017/0447.pdf>.
- [25] WANG X,WANG R J,SHI C,et al. Multi-component graph convolutional collaborative filtering[J]. Proceedings of the AAAI Conference on Artificial Intelligence,2020,34(4):6267-6274.
- [26] 康雁,李涛,李浩,等. 融合知识图谱与协同过滤的推荐模型[J]. 计算机工程,2020,46(12):73-79,87.  
KANG Y,LI T,LI H,et al. Recommendation model fusing with knowledge graph and collaborative filtering [J]. Computer Engineering,2020,46(12):73-79,87. (in Chinese)

编辑 吴云芳