



# 计算机科学

COMPUTER SCIENCE

## 一种基于层次超图注意力神经网络的服务推荐算法

杨东昇, 王桂玲, 郑鑫

引用本文

杨东昇, 王桂玲, 郑鑫. 一种基于层次超图注意力神经网络的服务推荐算法[J]. 计算机科学, 2024, 51(11): 103-111.

YANG Dongsheng, WANG Guiling, ZHENG Xin. Hierarchical Hypergraph-based Attention Neural Network for Service Recommendation [J]. Computer Science, 2024, 51(11): 103-111.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

### 基于注意力机制和深层特征优化的混凝土路面裂缝检测

Crack Detection of Concrete Pavement Based on Attention Mechanism and Deep Feature Optimization  
计算机科学, 2024, 51(11): 198-204. <https://doi.org/10.11896/jsjx.240100082>

### 面向自动驾驶的高精度实时语义分割算法架构

High-precision Real-time Semantic Segmentation Algorithm Architecture for Autonomous Driving  
计算机科学, 2024, 51(11): 174-181. <https://doi.org/10.11896/jsjx.231000009>

### 基于特征插值的深度图对比聚类算法

Feature Interpolation Based Deep Graph Contrastive Clustering Algorithm  
计算机科学, 2024, 51(11): 157-165. <https://doi.org/10.11896/jsjx.231000209>

### 用于谣言检测的图卷积时空注意力融合与图重构方法

Graph Convolution Spatio-Temporal Attention Fusion and Graph Reconstruction Method for Rumor Detection  
计算机科学, 2024, 51(11): 54-64. <https://doi.org/10.11896/jsjx.240300189>

### 融合Inception与SE-Attention的加密流量移动业务识别

Identification of Mobile Service Type of Encrypted Traffic Based on Fusion of Inception and SE-Attention  
计算机科学, 2024, 51(10): 399-407. <https://doi.org/10.11896/jsjx.230900103>

# 一种基于层次超图注意力神经网络的服务推荐算法

杨东昇 王桂玲 郑鑫

北方工业大学信息学院 北京 100144

北方工业大学大规模流数据集成与分析技术北京市重点实验室 北京 100144

(ydstheng1@163.com)

**摘要** 随着 Internet 和 Web 上各种服务和 API 数量的迅速增加,开发人员要快速准确地找到满足其需求的 API 变得越来越具有挑战性,因此亟需一个高效的推荐系统。目前,将图神经网络应用于服务推荐领域取得了巨大成功,但大多数方法仍然局限于简单的交互,忽略了 mashup 和 API 调用之间的内在关系;为了解决这个问题,提出了一种基于层次超图注意力的服务推荐方法(H-HGSR)来进行 API 推荐。首先定义了 8 种类型的超边,并探究了对应类型超边的超图邻接矩阵生成方法,然后提出了节点级和超边级的注意力机制。节点级注意力机制用于聚合特定类型超图邻接矩阵下的不同邻居的重要信息,以捕获 mashup 和 API 之间的高阶关系;超边级注意力机制用于对从不同类型超图邻接矩阵生成的节点嵌入进行加权组合。通过学习节点级和超边级注意力的重要性,可以获得更准确的嵌入表示。最后使用一个多层感知器神经网络(MLP)进行服务推荐。在 Programmable Web 真实数据集上进行了大量实验,结果表明,所提 H-HGSR 框架优于目前最先进的服务推荐方法。

**关键词:** 服务推荐;超图;图神经网络;注意力机制

**中图分类号** TP311

## Hierarchical Hypergraph-based Attention Neural Network for Service Recommendation

YANG Dongsheng, WANG Guiling and ZHENG Xin

School of Information, North China University of Technology, Beijing 100144, China

Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, North China University of Technology, Beijing 100144, China

**Abstract** With the rapid growth of various services and APIs on the Internet and the Web, it has become increasingly challenging for developers to quickly and accurately find APIs that meet their needs, thus requiring an efficient recommendation system. Currently, the application of graph neural networks in service recommendation has achieved great success, but many such methods are still limited to simple interactions and ignore the intrinsic relationships between mashups and API calls. To address this issue, this paper proposes a hierarchical hypergraph-based attention neural network for service recommendation method(H-HGSR) for API recommendation. First, eight types of hyperedges are defined, and the corresponding hypergraph adjacency matrix generation methods are explored. Then, node-level and hyperedge-level attention mechanisms are proposed. The node-level attention mechanism is used to aggregate important information from different neighbors under specific types of hypergraph adjacency matrices to capture high-order relationships between mashups and APIs. The hyperedge-level attention mechanism is used to weight the combination of node embeddings generated from different types of hypergraph adjacency matrices. By learning the importance of node-level and hyperedge-level attention, more accurate embedding representations can be obtained. Finally, a multi-layer perceptron neural network(MLP) is used for service recommendation. Extensive experiments are conducted on the Programmable Web real dataset, and the overall comparison results show that the proposed H-HGSR framework outperforms the state-of-the-art service recommendation methods.

**Keywords** Service recommendation, Hypergraphs, Graph neural networks, Attention mechanism

## 1 引言

当前,API 在 Web 和移动应用中的开发已经越来越多,开发人员通过组合不同的 API 来开发新应用程序,如 mash-

up。 mashup 是将多个 API 集成到一个接口中的 Web 应用程序的代表。然而,随着 API 数量的快速增长,开发人员要从众多候选 API 中选择合适的 API 变得越来越困难。为了解决这个问题,API 推荐引起了学术界的广泛关注,并且已经有

到稿日期:2023-11-01 返修日期:2024-03-27

基金项目:国家自然科学基金重点项目(61832004);国家自然科学基金国际(地区)合作与交流项目(62061136006)

This work was supported by the Key Project of the National Natural Science Foundation of China(61832004) and International(Regional) and Cooperation and Exchange Project of National Natural Science Foundation of China(62061136006).

通信作者:王桂玲(wanguiling@ncut.edu.cn)

许多用于 API 推荐的方法被提出。目前,协同过滤(CF)<sup>[1]</sup>是 Web 服务推荐系统中常用的方法之一。它通过分析 mashup 和 API 的历史调用信息,找到与目标 mashup 具有相似调用行为的其他 mashup,然后根据这些相似 mashup 的调用,推荐目标 mashup 可能感兴趣的 API。

在各种协同过滤技术中,矩阵分解(MF)<sup>[2]</sup>是应用于 Web API 推荐系统中的最流行的方法之一。该方法通过将 mashup 和 API 投影到共享的潜在空间(或称隐空间)中,并使用潜在特征向量(或称隐向量)来表示它们。然而,这种方法无法捕捉到 mashup 和 API 交互数据的复杂结构。当调用矩阵非常稀疏时,学习复杂的 mashup-API 关系变得更加困难。

目前,深度学习方法已被成功应用于推荐系统中,并展示了学习复杂表示的强大能力。在建模 mashup 和 API 之间的交互方面,一些工作采用多层感知机代替传统的矩阵分解方法<sup>[3]</sup>。另外,也有工作将神经网络与协同过滤相结合<sup>[4]</sup>,以缓解传统推荐算法中的数据稀疏性等问题,从而提高推荐系统的性能。然而,许多这样的方法使用 mashup-API 直接调用数据,忽略了调用之间的内在关系。因此,需要开发新的方法来解决这一问题,以更好地表示 mashup 和 API 之间的交互关系,提高推荐的准确性。

超图将边的概念推广到连接两个及两个以上的节点<sup>[5]</sup>,为捕获 mashup 和 API 之间复杂的高阶关系提供了一种自然的方法。然而,在服务推荐方面,还没有工作将不同形状的超图与层次注意力网络结合在一起进行研究。因此,本文提出了一种基于层次超图注意力的服务推荐方法——Hierarchical Hypergraph-based Attention Neural Network for Service Recommendation(H-HGSR)来填补这一空白。该方法定义多个类别的超边,具体化不同类型的高阶关系,然后设计了一个层次化的注意力网络来学习节点嵌入。该节点嵌入包含节点级注意力网络中各种节点邻居重要性的聚合信息,以及超边级注意力网络中各种超图的重要性信息。基于学习到的两个层次的注意力值,H-HGSR 可以以分层的方式获得邻居和多个超图的最佳组合,全面捕捉 mashup 和 API 之间复杂的高阶依赖关系,细化 mashup 和 API 的表示。这种嵌入表示包含丰富的高阶关系信息。实验结果表明,本文方法在推荐性能上优于现有的推荐方法。

本文的主要贡献如下:

- 1)提出了一种层次超图注意力网络,该网络同时考虑了节点级和超边级的注意力。通过这种层次注意力网络,可以同时学习节点和超图的重要性,从而获得更准确的嵌入表示。
- 2)提出了一种基于层次超图注意力神经网络的服务推荐算法,该方法可以自动学习邻居节点和不同超图中的重要信息,生成嵌入表示,并使用一个多层感知器神经网络进行服务推荐。
- 3)在 Programmable Web 数据集上进行了广泛的实验,实验结果验证了本文方法的有效性。

本文第 2 章介绍了相关工作;第 3 章给出了问题陈述和文中使用的术语的定义;第 4 章介绍了本文提出的模型;第 5 章描述了实验设置,并对实验结果进行了比较和分析;最后

总结全文,并提出了未来的研究方向。

## 2 相关工作

### 2.1 基于协同过滤的服务推荐系统

基于矩阵分解的协同过滤方法<sup>[6]</sup>因其高效性和准确性而受到广泛欢迎,它通过将 mashup-API 交互矩阵分解为两个低维矩阵的乘积,从而得到 mashup 和 API 的潜在特征向量表示。文献[7]提出了一种具有隐式相关正则化的概率矩阵分解方法。文献[8]将用户的显式和隐式偏好合并到正则化矩阵分解模型中,用于 API 推荐。此外,文献[9]提出了具有隐式相关正则化的概率矩阵分解方法。然而,由于传统的协同过滤算法没有考虑到 mashup 和 API 之间的高阶交互信息,因此在面对数据稀疏性较大的数据集时的推荐效果不理想,并且容易受到噪声特征的影响。

### 2.2 基于深度神经网络的服务推荐系统

神经网络已被广泛应用于服务推荐,缓解了数据稀疏性并提升了推荐性能。例如,文献[10]构建了一种新的神经网络架构,将其深度学习与因子分解机结合,学习特征并进行推荐。VAE-CF<sup>[11]</sup>扩展了基于协同过滤的变分自编码器,其考虑了通过多项式生成的直接邻域向量,但 VAE-CF 仅限于捕获二阶连接信号。GraphSAGE<sup>[12]</sup>利用局部结构信息和邻域中节点的特征,产生了更好的结果。文献[13]利用 API 关系来构建 API 图,并使用图卷积神经网络模型来进行 API 推荐。NeuMF<sup>[14]</sup>利用多层感知器来学习 mashup 和 API 的潜在表示向量,并通过合并它们的预测向量来统一 MF 和 MLP,从而与传统的 MF 模型相结合。NNCF<sup>[15]</sup>是一种基于邻域的神经协同过滤模型,其将 mashup 和 API 的邻域信息集成到神经协同过滤方法中,以此进行推荐。NGCF<sup>[16]</sup>证明了推荐系统中 mashup 和 API 之间高阶连接的重要性,通过多层迭代产生 mashup 和 API 的特征信息,然后按照矩阵分解的协同过滤框架进行推荐。然而,很多基于神经网络的方法只能应用于简单的图结构,并且在聚集高阶邻居信息时存在过平滑问题,这可能导致节点表示不可区分,从而降低了推荐性能。

### 2.3 超图在推荐系统中的应用

超图是对普通图的一种推广,其每一条超边可以表示不同数量的对象之间的相关性。受超图在复杂高阶依赖关系建模中的成功应用的启发<sup>[17]</sup>,超图已被广泛用于解决各种问题。随着深度学习的发展,一些研究将图神经网络和超图结合起来以增强表示学习。HGNN<sup>[18]</sup>是第一个设计超图卷积运算的工作,它从谱的角度处理表示学习中的复杂数据相关性。文献[19]将超图注意力引入超图卷积网络以提高其容量。也有一些研究人员将超图神经网络应用于推荐系统。文献[20]提出了一种名为 GDE 的模型,该模型在超图上传播节点嵌入,并在不堆叠层的情况下保留重要的图特征。HyperRec<sup>[21]</sup>利用超图神经网络的优势来建模下一个项目推荐的短期用户偏好。DHCF<sup>[22]</sup>利用超图神经网络来建模用户和项目之间的高阶相关性。文献[23]提出了一种自监督框架,使用超图增强后的跨视图对比学习架构来分别捕获本地信息和全局信息。

### 3 定义和问题陈述

表 1 列出了本文中使用的关键符号和说明。

表 1 符号和概念

Table 1 Symbols and their concepts

符号	说明
$R_i$	超边类型
$A_i$	超图邻接矩阵
$\Phi_i^A$	节点 $i$ 在 $A_i$ 中的邻居节点集合
$q_m^A, q_a^A$	初始输入特征
$e_m^A, e_a^A$	投影节点特征
$p_i^A$	捕获节点级注意力层后节点 $i$ 的特征
$p_i'$	通过池化 $p_i^A$ 后节点 $i$ 的特征
$z_i$	通过连接 $p_i'$ 后节点 $i$ 的特征
$\varepsilon_i$	最终输出特征
$h_{i,j}^A$	节点 $i$ 邻居 $j$ 在 $A_i$ 中的注意力系数
$\alpha_{i,j}^A$	节点 $i$ 邻居 $j$ 在 $A_i$ 中的注意力分数
$k^{R_i}$	来自 $R_i$ 的特征的注意力系数
$\beta^{R_i}$	来自 $R_i$ 的特征的注意力分数

本文设计了 8 种超边类型  $R_i$ 。类型 1:当两个 API 被同一个 mashup 调用时,这两个 API 之间存在某种相似性,因此它们之间形成一条超边。类型 2:当两个 mashup 调用同一个 API 时,我们认为这两个 mashup 之间存在某种相似性,因此它们之间形成一条超边。超图将边的概念推广到连接两个及两个以上的节点,本文最多探索了连接 3 个节点的超边,所提方法可以拓展到更多节点的连接,这将是未来的研究方向。类型 3:当 3 个 API 被同一个 mashup 调用时,我们认为这 3 个 API 之间的相似性对推荐有更积极的影响,因此它们之间形成一条超边。类型 4:当 3 个 mashup 调用同一个 API 时,它们之间形成一条超边。接下来的 4 种类型都包含两个 mashup 和两个 API,将这两种成对关系综合在一起,发现更复杂的隐含关系。因此,通过这种成对关系形成的超边可以更好地体现潜在的相似性。类型 5 和类型 6 分别表示两个 mashup 调用两个 API,形成两个 mashup 之间的超边和两个 API 之间的超边。类型 7 和类型 8 分别表示一个 API 被两个 mashup 同时调用,而另一个 API 只被这两个 mashup 中的一个调用,形成两个 mashup 之间的超边和两个 API 之间的超边。不同超边类型如图 1 所示,其中圆圈表示 mashup,方块表示 API。

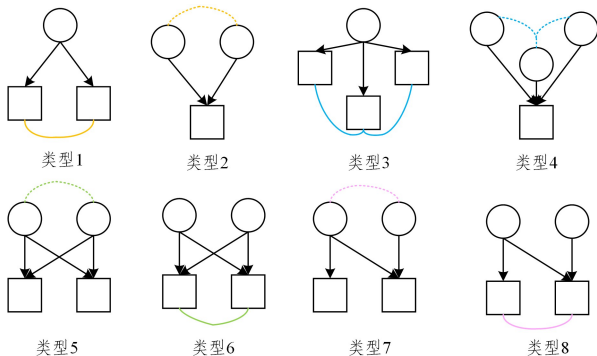


图 1 8 种超边类型

Fig.1 Eight types of hyperedge

**定义 1(超图邻接矩阵)** 对于给定的一种类型超边  $R_i$ , 生成一个该超边类型的超图邻接矩阵  $A_i$ , 当中  $A_i$  位置  $(i, j)$  处的值  $k > 0$  时,意味着对于节点  $i$  和节点  $j$ , 存在  $k$  个该

类型超边的实例。

问题陈述:对于给定的一个目标 mashup  $m$ , 本文的目标是生成  $K$  个推荐的 API, 其中推荐结果尽可能类似于  $m$  已经调用过的 API。

### 4 模型介绍

#### 4.1 超图邻接矩阵生成方法

超图将边的概念推广到连接两个及两个以上节点,它能够更好地表示多个顶点之间的关系,这为捕获 mashup 和 API 之间复杂的高阶关系提供了一种自然的方法,从而能够提升推荐系统的性能。图 2 展示了邻接矩阵和超图邻接矩阵之间的区别。在邻接矩阵中,只考虑了 mashup 和 API 之间的直接调用。但 mashup 1 和 mashup 2 同时调用了 API 3, 说明 mashup 1 和 mashup 2 具有某种相似性,这种相似性会对推荐的结果产生影响。但在邻接矩阵中  $(1, 2)$  和  $(2, 1)$  的位置都设置为 0, 这显然是不合理的。本文通过设计不同类型的超边来捕获 mashup 和 API 之间的这种相似性,因此超图邻接矩阵体现了图中的高阶连通性。根据不同的超边类型和定义 1, 找到对应超边类型  $R_i$  的所有超图实例来生成超图邻接矩阵  $A_i$ 。下面以超边类型 7 的超图邻接矩阵生成算法为例进行说明。

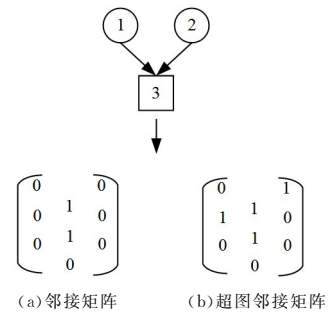


图 2 两种邻接矩阵的区别

Fig.2 Difference between two kinds of adjacency matrix

算法 1 描述了用于超边类型  $R_i$  的超图邻接矩阵生成过程。第一步是通过相同的 API 节点将所有边分成组,形成空矩阵  $M$ 。外部循环中的每步按如下步骤进行 ( $f$  和  $g$  表示任何两个 API 节点, API 集合用  $V$  表示): 首先,找到  $f$  的邻居和  $g$  的邻居之间的交集  $S$  和对称差  $D$ , 用  $n_S$  和  $n_D$  分别表示  $S$  和  $D$  中的节点数;如果在  $S$  中存在至少一个节点并且在  $D$  中存在一个节点,则进行下一步。

有 3 种类型的计数更新: 1) 对于相互的邻居  $x$ , 将  $x, f$  之间和  $x, g$  之间的计数加上  $n_D$ ,  $n_D$  是从对称差中选择任何一个节点的可能性的数量, 即  $C_{n_D}^1$ 。 2) 对于任何  $x \in S$  和  $b \in D$ ,  $x$  和  $b$  之间计数加 1,  $b$  和  $x$  之间计数加 1, 因为它们和  $f, g$  只能形成一个超图实例。 3) 对于每个节点  $b \in D$ , 即  $b$  是  $f$  或  $g$  的邻居, 如果  $b$  是  $f$  的邻居, 则  $b$  和  $f$  之间的计数增加  $n_S$ ; 如果  $b$  是  $g$  的邻居, 则  $b$  和  $g$  之间的计数增加  $n_S$ 。  $n_S$  是从相互邻居中选择任何一个节点的可能性的数量, 即  $C_{n_S}^1$ 。 计数完成时, 将矩阵输出。算法的时间复杂度为:  $O(C_{|V|}^2 \cdot (|S| + |S| \cdot |D| + |D|)) = O(|V|^2 \cdot |S| \cdot |D|) \leq O(|V|^2 \cdot \left(\frac{|S| + |D|}{2}\right)^2) \leq O(|V|^2 \cdot \left(\frac{|E|}{|V|}\right)^2) = O(|E|^2)$ 。其中  $V$

表示 API 的集合,  $S$  表示邻居之间的交集,  $D$  表示邻居之间的对称差,  $E$  表示边的集合。

#### 算法 1 类型 7 的超图邻接矩阵生成算法

输入: 边  $E$ ; 节点数  $N$ ; 邻居函数  $\Gamma$ ; 相交函数 INTERSECT; 对称差函数 SYMDIFF

输出: 超图邻接矩阵

1. 创建  $\mathbf{M} \leftarrow N \times N$  矩阵并填充为 0
2. for  $\forall f, g \in \text{API 节点 } V$  do
3.  $S \leftarrow \text{INTERSECT}(\Gamma(f), \Gamma(g))$
4.  $D \leftarrow \text{SYMDIFF}(\Gamma(f), \Gamma(g))$
5.  $n_S \leftarrow |S|, n_D \leftarrow |D|$
6. if  $n_S > 0$  and  $n_D > 0$  then
7. for  $\forall x \in S$  do
8. add  $n_D$  to  $(x, f)$  in  $\mathbf{M}$
9. add  $n_D$  to  $(x, g)$  in  $\mathbf{M}$
10. for  $\forall b \in D$  do
11. add 1 to  $(x, b)$
12. add 1 to  $(b, x)$
13. end
14. end
15. for  $\forall b \in D$  do
16. if  $b \in \Gamma(f)$  then
17. add  $n_S$  to  $(b, f)$  in  $\mathbf{M}$
18. if  $b \in \Gamma(g)$  then
19. add  $n_S$  to  $(b, g)$  in  $\mathbf{M}$
20. end
21. end
22. end

#### 4.2 框架组成

本文提出的基于层次超图注意力的服务推荐方法 (H-HGSR) 的整体框架由 4 个部分组成: 1) 初始化输入层, 用于对初始输入节点的特征进行增强; 2) 节点级注意力层, 使节点能够选择性地捕获来自不同邻居的信息; 3) 超边级注意力层, 捕捉每个类型的超边对于最终聚合的节点嵌入的不同重要性; 4) 预测层, 使用内积来预测偏好得分。图 3 给出了 H-HGSR 的主要模块。下文将详细介绍每个组件。

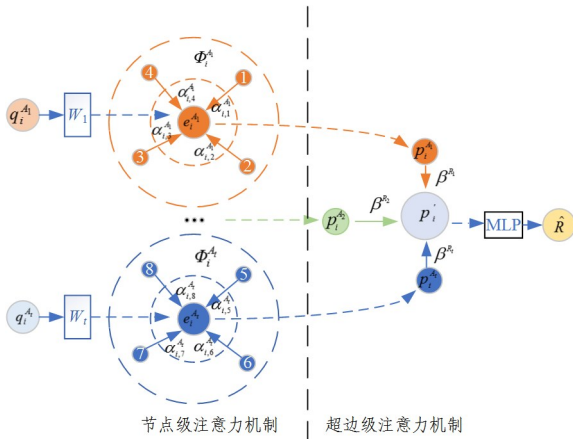


图 3 H-HGSR 的主要模块

Fig. 3 Main module of H-HGSR

#### 4.3 初始化输入层

初始化时, 对于给定类型的超图邻接矩阵  $\mathbf{A}_t$ , 将  $\mathbf{Q}_t =$

$\{q_{m_1}^A, q_{m_2}^A, \dots, q_{m_{|U|}}^A, q_{a_1}^A, q_{a_2}^A, \dots, q_{a_{|V|}}^A\}$  作为输入集合,  $q \in R^D$ ,  $q$  是每个 mashup 节点或 API 节点的初始特征,  $D$  是输入特征的维数。然后将输入集通过变换权重为  $\mathbf{W} \in R^{D \times D'}$  的线性变换层产生具有新的节点特征的集合  $\mathbf{E}_t$ ,  $\mathbf{E}_t = \{e_{m_1}^A, e_{m_2}^A, \dots, e_{m_{|U|}}^A, e_{a_1}^A, e_{a_2}^A, \dots, e_{a_{|V|}}^A\}$ ,  $e \in R^{D'}$ , 其中  $D'$  是投影特征的维数。此外, 为了在下一小节中分别使用每个  $\mathbf{A}_t$  中的注意力机制, 本文通过权重矩阵  $\mathbf{W}_t \in R^{D \times D'}$  来变换特征, 这意味着每种类型的超图与唯一的参数矩阵相关联。

$$e_t^A = q_t^A \mathbf{W}_t \quad (1)$$

#### 4.4 节点级注意力层

受 GAT 的启发, 对每个节点应用自注意力网络  $\text{att}_{\text{node}}$ 。由于不同的超图类型代表不同的高阶结构, 不同高阶结构会对输出结果产生不同影响, 因此本文在每个  $\mathbf{A}_t$  中单独应用  $\text{att}_{\text{node}}$ , 使用深度神经网络  $\text{att}_{\text{node}}: R^D \times R^{D'} \rightarrow R$  来计算注意力系数。

$$h_{i,j}^A = \text{att}_{\text{node}}(e_i^A, e_j^A) \quad (2)$$

注意力系数  $h_{i,j}^A$  表示节点  $j$  的特征对节点  $i$  在  $\mathbf{A}_t$  中的重要性大小。为了简化计算, 对 mashup 和 API 使用相同的计算方式。本文使用 softmax 对注意力系数进行归一化, 最终得到注意力分数  $\alpha_{i,j}^A$ , 其中  $\Phi_i^A$  是  $R_t$  类型的超图中节点  $i$  的邻居节点。

$$\alpha_{i,j}^A = \text{softmax}(h_{i,j}^A) = \frac{\exp(h_{i,j}^A)}{\sum_{x \in \Phi_i^A} \exp(h_{i,x}^A)} \quad (3)$$

实验中, 对  $\text{att}_{\text{node}}$  采用由权重向量  $\mathbf{a} \in R^{2D'}$  参数化的单层前馈神经网络和 LeakyReLU 激活函数:

$$h_{i,j}^A = \text{LeakyReLU}([\mathbf{q}_i^A \mathbf{W}_t \parallel \mathbf{q}_j^A \mathbf{W}_t](\mathbf{a}^A)^T) \quad (4)$$

$\mathbf{a}^A \in R^{2D'}$  是用于将初始特征投影到所需维度中的权重向量,  $T$  是转置符号,  $\parallel$  是级联操作。

最后, 聚合节点  $i$  在  $R_t$  类型超边的超图邻接矩阵的邻居信息。

$$\mathbf{p}_i^A = \sum_{j \in \Phi_i^A} \alpha_{i,j}^A \cdot e_j^A \quad (5)$$

在完成上述步骤之后, 即可获得高阶结构  $\mathbf{A}_t$  的节点特征  $\mathbf{p}_i^A$ , 并由此来获得每个高阶结构  $\mathbf{A}_t \in \mathbf{A}$  的不同节点特征表示。最后, 本文将  $\mathbf{p}_i^A$  堆叠在一起作为矩阵的行, 以形成该模块的最终输出  $\mathbf{P}^A$ , 输出矩阵的维度为  $(6861, 32)$ 。

#### 4.5 超边级注意力层

本文提出一个超边级的注意力机制, 自动选择多个不同超图邻接矩阵中的重要信息, 并对这些重要信息进行融合。取  $l$  ( $1 \leq l \leq t$ ) 个不同的  $\mathbf{P}^A$  作为输入的集合  $L$ , 本文使用  $\text{att}_{\text{edge}}: R^{(U+V) \times D'} \rightarrow R$  来计算超图的注意力系数  $k^{R_t}$ , 其中  $U$  是 mashup 的集合,  $V$  是 API 的集合,  $|U+V|$  是 mashup 和 API 的节点总数。

$$k^{R_t} = \text{att}_{\text{edge}}(\mathbf{P}^A) \quad (6)$$

其中,  $k^{R_t}$  表示  $R_t$  类型超边的重要性大小。在本文的实验中, 详细的计算过程如下:

$$\mathbf{s} = \mathbf{P}^A \mathbf{W}_s \quad (7)$$

$$k^{R_t} = \mathbf{s}^T \quad (8)$$

这个步骤是在计算嵌入  $\mathbf{s}$  时考虑所有的节点嵌入,  $\mathbf{W}_s \in R^{D'}$  是一个全局上下文权重, 捕获节点嵌入矩阵  $\mathbf{P}^A$  的信息并

降低其维数; $\mathbf{v}$ 是权重向量, $\mathbf{v} \in R^{U+V}$ 。

与上一节相同,本文使用 softmax 对  $k^{R_i}$  进行归一化,最终得到超边级注意层的注意力分数  $\beta^{R_i}$ 。

$$\beta^{R_i} = \text{softmax}(k^{R_i}) = \frac{\exp(k^{R_i})}{\sum_{A_j \in L} \exp(k^{R_j})} \quad (9)$$

然后,将每个超图邻接矩阵中的节点信息乘以  $\beta^{R_i}$  并聚合。

$$p_i' = \sum_{A_j \in L} \beta^{R_i} \cdot p_i^{A_j} \quad (10)$$

对于整个矩阵  $\mathbf{P}^{A_i}$ ,聚合运算表示如下:

$$\mathbf{P}' = \sum_{A_j \in L} \beta^{R_i} \cdot \mathbf{P}^{A_j} \quad (11)$$

单一的注意机制可能无法捕捉到图中的所有信息,因此使用多头注意机制来捕捉图中每个子空间的信息。具体地,使用  $G$  独立注意机制来获得节点  $i$  的特征  $p_i'^g$ ,然后将其特征级联成  $\mathbf{z}_i$ ,如式(12)所示:

$$\mathbf{z}_i = \parallel_{g=1}^G (p_i'^g) \quad (12)$$

最后,添加了第二个节点级注意力层,用于最终的信息融合。将  $\mathbf{z}_i$  作为节点  $i$  的特征来计算注意力分数,计算过程与式(1)一式(5)相同,得到最终的输出特征  $\mathbf{e}_i$ 。其中将  $\mathbf{A}_i$  的类型选取为图的邻接矩阵,因为图的低阶结构包含更多节点之间的直接调用信息,可直接反映它们的依赖关系。

#### 4.6 预测层

本节旨在通过多层感知器神经网络和非线性变换来优化  $\mathbf{e}_i$  以产生  $\mathbf{c}_i$ 。采用三层 MLP 来优化特征,在 MLP 的每一层,使用权重  $\mathbf{W}_r \in R^{D_r \times D_r}$  来更新输入特征并使用 LeakyReLU 进行非线性变换。 $D_r$  是前一层输出的维度, $D_o$  是本文需要定义的 MLP 输出的维度。

$$\mathbf{c}_i = \text{LeakyReLU}(\mathbf{e}_i \mathbf{W}_r) \quad (13)$$

最后使用  $\mathbf{c}_m$  和  $\mathbf{c}_a$  的点积来预测它们的相关性  $\hat{R}_{m,a}$ ,其中  $\hat{R}_{m,a} \in [0,1]$  的值反映了  $m \in U$  调用  $a \in V$  的可能性。

$$\hat{R}_{m,a} = \mathbf{c}_m \odot \mathbf{c}_a \quad (14)$$

#### 4.7 模型学习优化

模型学习的目标是 minimized 在推荐系统中经常使用的二进制交叉熵损失。它检查了 mashup-API 调用之间的真实值和预测的差异。真实值  $R_{m,a}$  表示  $m$  是否与  $a$  交互的二进制化的 1 或 0。将  $R_{m,a}$  的值置为 1 表示 mashup 和 API 交互,否则为 0。 $N$  是用来计算损失的所有节点的数量。预测得分  $\hat{R}_{m,a}$  则表示  $m$  与  $a$  相关的可能性有多大。损失函数被定义为:

$$\text{Loss} = -\frac{1}{N} \sum_{m \in U, a \in V} R_{m,a} \log \hat{R}_{m,a} + (1 - R_{m,a}) \log(1 - \hat{R}_{m,a}) \quad (15)$$

## 5 实验

本文在 Programmable Web 数据集上进行了大量实验,旨在回答以下研究问题。

RQ1:本文提出的 H-HGSR 方法是否优于最先进的协同过滤方法?

RQ2:不同类型的超边如何影响模型的性能?

RQ3:不同的超参数对实验结果有什么影响?

### 5.1 实验设置

#### 5.1.1 数据集

本文使用 Programmable Web 数据集来证明所提模型的有效性。该数据集包含 17829 个 API 和 6340 个 mashup 以及它们的历史调用。因为数据集没有 mashup 和 API 之间的评级,所以采用它们的调用数据作为评级。例如,mashup m1 调用 API a6,它们的调用数据是 1,使用这个值作为 mashup-API 对之间的调用。之后,对数据集进行重新处理:去除空白 API 和 mashup,得到 mashup 5691 个,API 1170 个。在测试集和训练集中,数据集被分成 4 部分,每个实验中使用其中 3 部分进行训练,另一部分用于测试,最终结果是 4 个实验的平均值。通过这种方式,使得测试集包含在开始时划分的所有测试集和训练集。

#### 5.1.2 评价指标

本文使用命中率(HR)和归一化折损累计增益(NDCG)来评估 API 推荐的性能。

$$HR@K = \frac{\text{number of hits}@K}{N_m} \quad (16)$$

$$DCG_m@K = \sum_{i=1}^K \frac{2^{\hat{R}_{m,i}} - 1}{\log_2(i+1)} \quad (17)$$

$$NDCG_m@K = \frac{DCG_m@K}{K} \quad (18)$$

$$NDCG@K = \frac{\sum_{m \in (\text{test set} \cap M)} NDCG_m@K}{|\text{test set} \cap M|} \times HR@K \quad (19)$$

式(16)中, $N_m$  是测试集中的 mashup 数量。使用 NDCG 来衡量模型生成的预测列表中的 API 排名的有效性。通过式(18)计算 NDCG,其中在  $NDCG_m@K$  中使用  $K$ ,是因为模型生成的结果列表被排序。 $\text{test set} \cap M$  表示测试集中的 mashup, $|\text{test set} \cap M|$  是这样的 mashup 数量。将最终结果乘以  $HR@K$ ,以反映本文模型提供的增益和精度。

#### 5.1.3 超参数

训练设置将所有数据集按 6:2:2 的比例划分为训练集、验证集和测试集使用 Pytorch 来实现 H-HGSR。超参数的最佳学习率为 0.002,收敛期为 100,隐藏层大小 32,正则化系数为  $5 \times 10^{-4}$ ,多头注意力机制的数量为 12,丢弃率为 0.6。这些参数的设置是通过实验获得的。

#### 5.1.4 实验环境

实验在 GPUSHARE<sup>1)</sup>上运行,该 GPU 云计算平台上使用 NVIDIA A100 GPU 和 64 GB 内存。源代码基于 Pytorch 1.12.0 和 cuda 11.3 实现。

#### 5.1.5 基准模型

将本文提出的 H-HGSR 与一些最先进的方法进行比较:

1)AMF<sup>[24]</sup>:注意矩阵分解模型,使用神经注意网络来学习不同特征的重要性。

2)NGCF<sup>[16]</sup>:基于 GCN 的协同过滤方法,利用项目和用户的高阶连接性信号传播嵌入。

3)HACF<sup>[25]</sup>:用于服务推荐的高阶数据增强协同过滤方法。

<sup>1)</sup> <https://www.example.com>

4) GAT-CF<sup>[26]</sup>: 基于图注意力网络的协同过滤模型。

5) MISR<sup>[27]</sup>: 基于深度学习的服务推荐框架。

6) MGSr<sup>[28]</sup>: 一种通过 motif 邻接矩阵的注意机制来捕获高阶结构的 Web 服务推荐系统。

7) H-MGSr<sup>[29]</sup>: 基于 motif 邻接矩阵的分层注意机制的 Web 服务推荐系统。

## 5.2 性能比较 (RQ1)

对于所有基线和 H-HGSr 变体, 运行模型直到损失值收敛, 并获得 HR 和 NDCG 的最佳结果。本文将  $K$  的取值范围设置为 2~10 并计算 HR 和 NDCG。H-HGSr123578 是在 H-HGSr 变体中获得最佳结果的版本, 后文将其简称为 H-HGSr\*。表 2 和表 3 列出了基线和 H-HGSr\* 的性能结果。

表 2 HR 在不同  $K$  值时的比较

Table 2 Comparison of HR at different  $K$  values

模型	K 值									
	2	3	4	5	6	7	8	9	10	
AMF	0.4021	0.4152	0.4182	0.4262	0.4352	0.4521	0.4921	0.5201	0.5583	
NGCF	0.4372	0.4552	0.5037	0.5537	0.6012	0.6252	0.6317	0.6677	0.6882	
MISR	0.6494	0.6760	0.7023	0.7211	0.7286	0.7392	0.7392	0.7407	0.7441	
HACF	0.6645	0.7190	0.7367	0.7483	0.7547	0.7603	0.7604	0.7606	0.7616	
GAT-CF	0.5967	0.6951	0.7561	0.7805	0.8171	0.8293	0.8293	0.8293	0.8293	
MGSr	0.5976	0.6707	0.7561	0.7805	0.8171	0.8293	0.8415	0.8415	0.8415	
H-MGSr3-7	0.6902	0.7624	0.7904	0.8174	0.8382	0.8540	<b>0.8638</b>	0.8721	0.8770	
H-HGSr*	<b>0.7007</b>	<b>0.7889</b>	<b>0.8097</b>	<b>0.8283</b>	<b>0.8561</b>	<b>0.8561</b>	0.8631	<b>0.8770</b>	<b>0.8840</b>	

表 3 NDCG 在不同  $K$  值时的比较

Table 3 Comparison of NDCG at different  $K$  values

模型	K 值									
	2	3	4	5	6	7	8	9	10	
AMF	0.3321	0.3498	0.3694	0.3972	0.3999	0.4093	0.4109	0.4271	0.4306	
NGCF	0.3521	0.3841	0.4393	0.4647	0.4678	0.4580	0.4603	0.4613	0.4647	
MISR	0.5209	0.5050	0.5343	0.5301	0.5351	0.5433	0.5569	0.5568	0.5583	
HACF	0.5626	0.5666	0.5627	0.5657	0.5711	0.5716	0.5718	0.5725	0.5726	
GAT-CF	0.5255	0.5718	0.6020	0.6108	0.6239	0.6272	0.6272	0.6197	0.6197	
MGSr	0.5255	0.5604	0.6009	0.6095	0.6229	0.6263	0.6319	0.6319	0.6264	
H-MGSr3-7	0.6454	0.6760	0.6840	0.6877	0.6912	0.6933	0.6920	0.6810	0.6890	
H-HGSr*	<b>0.6570</b>	<b>0.6944</b>	<b>0.6979</b>	<b>0.6971</b>	<b>0.7046</b>	<b>0.7006</b>	<b>0.6984</b>	<b>0.6990</b>	<b>0.6989</b>	

总体来看, 本文提出的模型优于现有的工作。AMF, NGCF 和 MISR 不考虑高阶连通性, 基于潜在因素的方法无法有效利用 mashup 和 API 之间的深层交互。HACF, GAT-CF, MGSr, H-MGSr 和 H-HGSr\* 附加 mashup-API 高阶关系, 比前三种模型取得了更好的结果, 表明了从相互作用数据捕获高阶特征的益处。本文的 H-HGSr\* 优于所有基线, 因为 H-HGSr\* 不仅考虑了高阶连通性, 还学习到了不同类型超边的重要性, 并且本文的超图邻接矩阵的设计能够更好地表示 mashup 和 API 之间的深层交互。

我们观察到一个有趣的现象: 所有模型的性能可以分为 3 个不同的组。表现最差组由 AMF 和 NGCF 组成, 该组都没有考虑图的高阶连通性。中间组中的大多数方法考虑了高阶连接性, 这表明了将高阶信息附加到预测模型中的有效性。该组中表现最好的是 MGSr, 它手动选择模型训练中的 Motif 类型。高性能组包括 H-MGSr 和 H-HGSr\*, 它们都利用了层次性的注意力机制来捕获不同类型信息的重要性, 更好地获得节点的嵌入表示。H-HGSr\* 的性能优于 H-MGSr, 主要归因于超图邻接矩阵能对高阶结构进行更为准确的表示, 避免信息干扰, 从而带来性能的提升。

### 5.3 不同类型的超边对模型表现的影响 (RQ2)

将超边信息引入已经被证明能改进推荐结果。本节将研究每种类型的超边如何助力这一创新的成功。

图 4 展示了在 H-HGSr 框架上应用不同类型超边时 HR 和 NDCG 得分的细节。本文比较了不同的超边组合的结果, 结果表明, 含有类型 4 超边变体的 HR 值较小。对于

NDCG 也是如此。本文还发现使用的超边类型越多, 性能越好。总体而言, H-HGSr123578 与其他方法相比性能最佳。

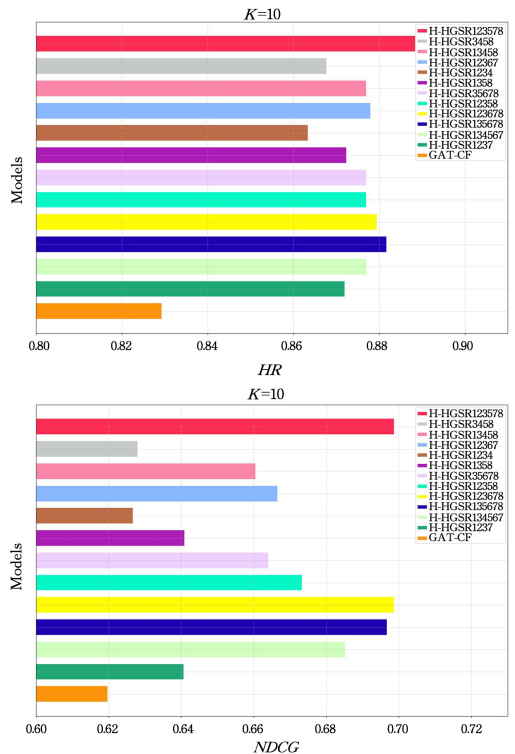


图 4 H-HGSr 变体的 HR 和 NDCG 对比

Fig. 4 Comparison of HR and NDCG of H-HGSr variants

由于含有类型 4 超边变体的性能较差, 因此本文对含有

类型4超边变体进行改进,设计了类型9超边,如图5所示。当4个mashup调用同一个API时,它们之间形成一条超边,本文通过类型9超边来探索更多节点的连接,以捕获更复杂的高阶关系。将含有类型4超边变体中的类型4超边换成类型9超边进行实验,实验结果如图6所示,换成类型9超边变体后的模型性能全部优于对应的类型4超边变体,表明捕获更复杂的高阶关系更有助于性能的提升。

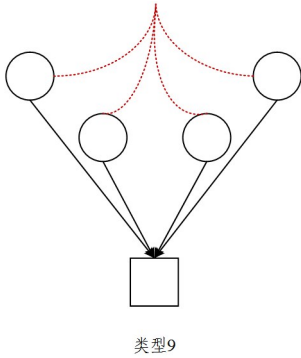


图5 类型9超边  
Fig. 5 Type nine hyperedge

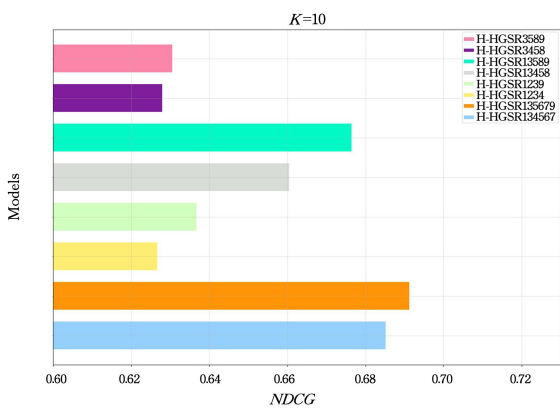
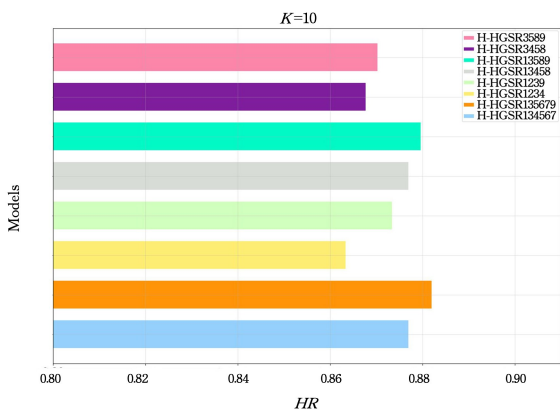


图6 含类型9超边变体对比

Fig. 6 Comparison of variants including type nine hyperedge

### 5.4 超参数分析(RQ3)

本节研究了H-HGSR的超参数对推荐性能的影响。固定其他参数,只改变实验中的超参数。

#### 5.4.1 多头注意力机制的数量

为了评估多头注意力机制的数量对推荐性能的影响,本文将头数从8增加到14,步长为2,结果如图7所示。可以观察到,推荐性能先显著提高,并且当头数为12时达到最佳值,

当头数从12增加到14时,性能下降。这一结果表明,增加多头注意力机制的数量并不总是有益的,当头数过多时,模型需要计算更多的注意力分数,这可能导致计算负担增大,训练时间延长,甚至可能导致过拟合问题。另外,当聚合高阶邻居时,过多的头数可能会导致过度平滑问题,使得模型无法有效地捕捉到数据中的细微特征和差异,从而影响推荐性能。相反,当头数过少时,模型可能无法充分利用多头注意力机制的优势,无法捕捉到数据中的复杂关系和特征,导致推荐性能下降。

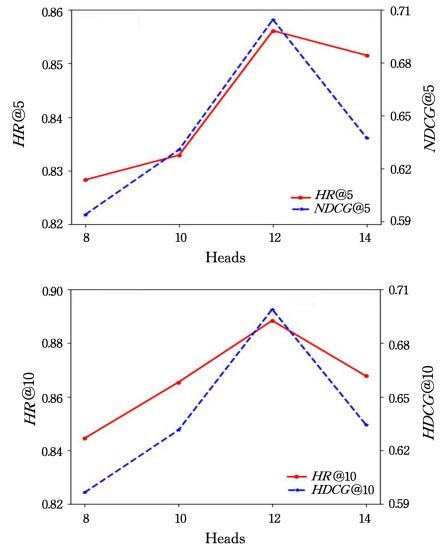


图7 多头注意力机制的数量影响

Fig. 7 Quantitative effects of multi-head-attention

#### 5.4.2 丢弃率

Drop out是提高测试集泛化能力的常用技术。为评估Drop out对推荐性能的影响,将Drop out从0.2增加到0.8,步长为0.2,结果如图8所示。

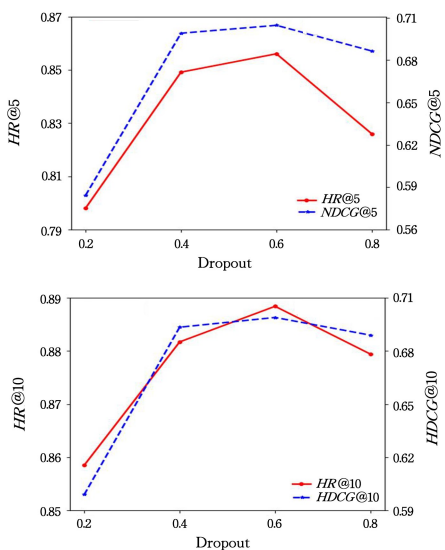


图8 丢弃率的影响

Fig. 8 Effect of drop out rate

较大的丢弃率更有助于提升推荐性能,因为Drop out参数设置过小,模型在训练过程中丢弃的神经元较少,这可能导致模型过拟合训练数据,在测试集上的泛化能力下降。过低

的丢弃率也可能导致模型对噪声和不确定性的适应能力不足,使模型在面对新数据时表现不佳。当 Drop out=0.6 时, H-HGSR 具有最好的推荐性能。当 Drop out 从 0.6 增加到 0.8 时,推荐性能迅速下降,这是由于模型丢弃了较多的神经元,无法充分学习数据的特征和关系,模型的表达能力下降,从而影响推荐性能。

#### 5.4.3 正则化系数

正则化的目的是防止模型过拟合。为了评估  $\lambda$  对推荐性能的影响,本文探索  $\lambda$  的值介于  $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$  之间。结果如图 9 所示,可以观察到,性能首先随着  $\lambda$  的增加而提升。当  $\lambda = 5 \times 10^{-4}$  时,可以实现最佳性能。然而,当  $\lambda$  设置为  $1 \times 10^{-3}$  时,性能显著下降,表明太强的正则化会对模型训练产生负面影响。这是因为正则化是通过惩罚模型复杂度来防止过拟合,当正则化系数过大时,惩罚项的影响就会过于强烈,导致模型更加趋向于简单的形式,无法很好地拟合训练数据。

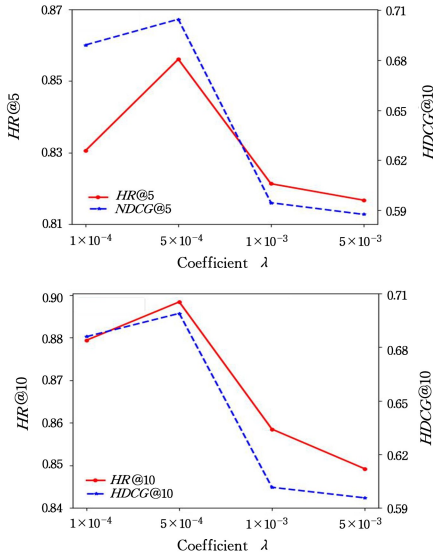


图 9 正则化系数的影响

Fig. 9 Effect of regularization coefficient

## 5.5 Movielens 数据集上的研究

本文使用 Movielens-1M 数据集进行进一步研究,该数据集被许多协同过滤模型广泛使用。本文使用 Movielens-1M 中 100 万个评级的版本,其中用户对电影有明确的评级。对数据集进行重新处理,过滤掉空白的用户和电影,得到数据集的统计数据,如表 4 所列。

表 4 Movielens-1M 数据集的统计数据

Table 4 Statistics of Movielens-1M dataset

Dataset	Users	Movies	Ratings
Movielens-1M	6 040	3 706	$1 \times 10^6$

在 Movielens-1M 数据集上,本文使用 HR 和 NDCG 来比较所有基线的性能。总体比较结果如图 10 所示。H-HGSR\* 取得了最好的性能,这表明层次超图注意力机制有助于提升推荐性能。所有考虑高阶连通性的方法,包括 HACF, GAT-CF, MGSR, H-MGSR 和 H-HGSR\* 都比其他方法获得了更好的 HR 和 NDCG,这表明了协同过滤中考虑高阶关系的正确性。

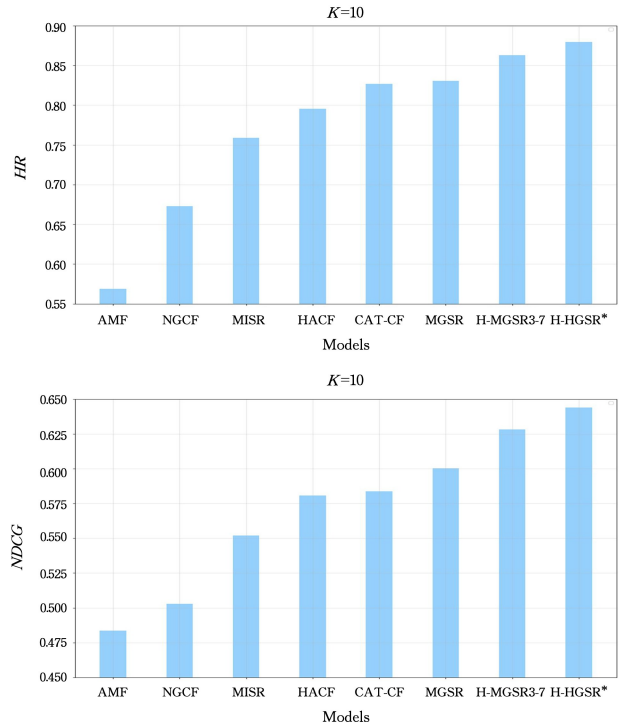


图 10 Movielens-1M 数据集上的性能对比

Fig. 10 Performance comparison on Movielens-1M dataset

**结束语** 在过去的十年中,Web 服务迅速发展,使得开发人员找到合适的 API 变得越来越具有挑战性。近年来,基于图神经网络的模型在服务推荐领域取得了巨大成功。然而,许多这样的方法使用 mashup-API 的直接调用数据,忽略了调用之间的内在关系。因此,本文提出了一种基于层次超图注意力的服务推荐方法,利用不同类型的超边信息和分层自注意力机制来进行 API 推荐。对于特定类型的超图邻接矩阵,使用节点级注意力有选择地捕获来自不同邻居的信息,以区分节点的各种邻居的重要性。通过超边级注意力,自动选择不同类型的超图邻接矩阵中包含的各种复杂信息,从而有助于节点的嵌入表示。最后,使用多层感知机来优化节点嵌入并生成推荐结果。本文提出的超图邻接矩阵设计合理,能够更准确地表示高阶结构,避免了信息的干扰,并带来了性能的提升。在公开数据集上进行了大量实验,结果表明所提方法在各项指标上均优于现有的基准方法。在未来的研究中,可以进一步探索连接更多节点的超边,以捕获更复杂的高阶关系。此外,计划将本文方法应用于更丰富的数据集,以提高方法的泛化能力。

## 参考文献

- [1] ZHENG N, LI Q, LIAO S, et al. Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr[J]. Journal of Information Science, 2010, 36(6): 733-750.
- [2] WANG R, CHENG H K, JIANG Y, et al. A novel matrix factorization model for recommendation with LOD-based semantic similarity measure[J]. Expert Systems with Applications, 2019, 123: 70-81.
- [3] LIU Z, GUO S, WANG L, et al. A multi-objective service composition recommendation method for individualized customer:

- hybrid MPA-GSO-DNN model[J]. *Computers & Industrial Engineering*, 2019, 128: 122-134.
- [4] LIU S, ZHENG Y. Long-tail session-based recommendation [C]// *Proceedings of the 14th ACM Conference on Recommender Systems*. 2020: 509-514.
- [5] BRETTO A. *Hypergraph theory: An introduction* [M]. Cham: Springer, 2013.
- [6] YUE W, WANG Z, ZHANG J, et al. An overview of recommendation techniques and their applications in healthcare[J]. *IEEE/CAA Journal of Automatica Sinica*, 2021, 8(4): 701-717.
- [7] YAO L, WANG X, SHENG Q, et al. Service recommendation for Mashup composition with implicit correlation regularization [C]// *2015 IEEE International Conference on Web Services*. 2015.
- [8] FLETCHER K. Regularizing matrix factorization with implicit user preference embeddings for web API recommendation[C]// *2019 IEEE International Conference on Services Computing (SCC)*. IEEE, 2019: 1-8.
- [9] YAO L, WANG X, SHENG Q Z, et al. Service recommendation for mashup composition with implicit correlation regularization [C]// *2015 IEEE International Conference on Web Services*. IEEE, 2015: 217-224.
- [10] GUO H, TANG R, YE Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[J]. *arXiv*: 1703.04247, 2017.
- [11] LIANG D, KRISHNAN R G, HOFFMAN M D, et al. Variational autoencoders for collaborative filtering[C]// *Proceedings of the 2018 World Wide Web Conference*. 2018: 689-698.
- [12] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[J]. *arXiv*: 1706.02216, 2017.
- [13] ZHANG Y, YANG H, KUANG L. A web API recommendation method with composition relationship based on GCN[C]// *2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom)*. IEEE, 2020: 601-608.
- [14] HE X, LIAO L, ZHANG H, et al. Neural collaborative filtering [C]// *Proceedings of the 26th International Conference on World Wide Web*. 2017: 173-182.
- [15] BAI T, WEN J R, ZHANG J, et al. A neural collaborative filtering model with interaction-based neighborhood [C]// *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. 2017: 1979-1982.
- [16] WANG X, HE X, WANG M, et al. Neural graph collaborative filtering [C]// *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019: 165-174.
- [17] HE L, CHEN H, WANG D, et al. Click-through rate prediction with multi-modal hypergraphs [C]// *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021: 690-699.
- [18] FENG Y, YOU H, ZHANG Z, et al. Hypergraph neural networks [C]// *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019: 3558-3565.
- [19] BAI S, ZHANG F, TORR P H S. Hypergraph convolution and hypergraph attention [J]. *Pattern Recognition*, 2021, 110: 107637.
- [20] PENG S, SUGIYAMA K, MINE T. Less is more: reweighting important spectral graph features for recommendation [C]// *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022: 1273-1282.
- [21] WANG J, DING K, HONG L, et al. Next-item recommendation with sequential hypergraphs [C]// *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020: 1101-1110.
- [22] JI S, FENG Y, JI R, et al. Dual channel hypergraph collaborative filtering [C]// *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020: 2020-2029.
- [23] XIA L, HUANG C, XU Y, et al. Hypergraph contrastive collaborative filtering [C]// *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2022: 70-79.
- [24] NGUYEN M, YU J, NGUYEN T, et al. Attentional matrix factorization with context and co-invocation for service recommendation [J]. *Expert Systems with Applications*, 2021, 186: 115698.
- [25] NGUYEN M, YU J, NGUYEN T, et al. High-order autoencoder with data augmentation for collaborative filtering [J]. *Knowledge-Based Systems*, 2022, 240: 107773.
- [26] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks [J]. *arXiv*: 1710.10903, 2017.
- [27] MA Y, GENG X, WANG J. A deep neural network with multiplex interactions for cold-start service recommendation [J]. *IEEE Transactions on Engineering Management*, 2020, 68(1): 105-119.
- [28] WANG G, YU J, NGUYEN M, et al. Motif-based graph attentional neural network for web service recommendation [J]. *Knowledge-Based Systems*, 2023, 269: 110512.
- [29] ZHENG X, WANG G, ZHANG J, et al. H-MGSR: A Hierarchical Motif-based Graph Attention Neural Network for Service Recommendation [C]// *2023 IEEE International Conference on Web Services (ICWS)*. IEEE, 2023: 553-562.



**YANG Dongsheng**, born in 1995, post-graduate. His main research interests include recommendation system and deep learning.



**WANG Guiling**, born in 1978, Ph. D., professor, is a professional member of CCF (No. 17649M). Her main research interests include data integration, services computing and large-scale streaming.