

# 基于知识图谱的推荐算法研究

薛艳斌,王宏生

(沈阳工业大学 信息科学与工程学院,辽宁 沈阳 110870)

**摘要:** 将知识图谱作为外部信息引入推荐系统可以有效缓解推荐系统的数据稀疏问题。文章提出一种端到端的神经网络模型,文章使用一种图注意力机制取代基于相似度或交换矩阵计算的离线元路径方法;另外,根据知识图谱中的项目的不同实体类型,文章提出一种多视图记忆注意力网络去学习更深层次的项目表征。文章在 MovieLens 数据集上进行了实验,实验结果表明,本文的模型明显优于 Top-N 推荐基线模型。

**关键词:** 推荐系统;知识图谱;图注意力机制;记忆注意力网络

**中图分类号:** TP183; **文献标识码:** A **文章编号:** 2096-9759(2023)06-0096-03

## Research on Recommendation Algorithm based on Knowledge Graph

XUE Yanbin,WANG Hongsheng

(School of information science and engineering, Shenyang University of Technology, Shen Yang 110870)

**Abstract:** Exploiting the external information called knowledge graph to recommendation has shown to effectively alleviate the data sparsity problem of recommendation system. Article proposes an end-to-end neural network model. We use a graph attention mechanism to replace the offline meta-path method based on similarity or commuting exchange matrix. In addition, according to the different entity types of items in the knowledge graph, we propose a multi-view memory attention network to learn more profound item features. Experiments on MovieLens dataset show the effectiveness of our model significantly outperform baseline model for Top-N recommendation.

**Key words:** recommendation system; knowledge graph; graph attention mechanism; memory attention network

### 1 引言

在现代生活中,推荐系统对于用户从大量的产品或服务中快速做出选择至关重要。Top-N 推荐系统的任务是为用户列出一个未来极有可能进行交互的项目的排名列表。在众多推荐算法中,协同过滤是最著名的,已经成功地用于 Top-N 推荐问题的建模。矩阵分解<sup>[1]</sup>和神经协同过滤<sup>[2]</sup>等模型就是协同过滤技术的几个实例。

协同过滤算法通过利用用户和项目之间的交互来预测用户对项目的偏好。然而,用户与一些极少被系统利用到的项目的交互很少,这通常会导致推荐系统的数据稀疏问题。为了解决这个问题,PER<sup>[3]</sup>和 FMG<sup>[4]</sup>将知识图谱视为异构信息网络,并提取基于潜在特征的元路径或元图来表示用户和项目之间沿着不同类型的关系路径或图的连通性。现有的方法虽然在一定程度上缓解了推荐系统的数据稀疏问题,但仍存在以下缺点。第一,它们不是端到端的模型,这使得它们严重依

赖于手动选择的元路径或元图,其次是对实体的离线嵌入构造、相似度或交换矩阵计算的效率都不高。第二,每个用户可能会查找项目的不同属性。例如,用户可能会根据一部电影的演员阵容或者导演来决定是否观看这部电影。第三,在第二的基础上,用户还会查找更深层次的特征。例如,《泰坦尼克号》是由凯特·温丝莱特主演的,这个演员是奥斯卡获奖者。用户不仅会关注电影的演员阵容是什么,还会关注这些演员是否获得过奥斯卡奖。

为解决上面提到的问题,本文利用知识图谱来辅助推荐系统,提出了一个端到端的神经网络模型(GAMAT)。本文利用知识图谱中的项目的不同实体类型来丰富用户与物品的描述。本文提出的模型有以下优点。第一,它是端到端的,不需要进行离线相似性或交换矩阵的计算,还不需要手动选择元路径。本文使用图注意力网络来解决这个问题。第二,本文提出一种多视图多头记忆注意力网络层,用来按“组件”提取

收稿日期:2023-03-11

作者简介:薛艳斌(1997-),男,研究生,硕士,主要研究方向:机器学习与智能系统;王宏生(1964-),男,副教授,硕士生导师,主要研究方向:软件设计自动化,智能信息检索。

- [2] 孙天鹏,周宁宁,黄国方.新的基于 GAN 的局部写实感漫画图像风格迁移[J].计算机工程与应用,2022,58(14):167-176.
- [3] 兰天,辛月兰,殷小芳,等.基于生成对抗网络的无监督图像风格迁移[J].计算机工程与科学,2021,43(10):1789-1795.
- [4] 沈瑜,杨倩,苑玉彬,等.目标边缘清晰化的图像风格迁移[J].激光与光电子学进展,2021,58(12):241-253.
- [5] 聂雄峰,王俊英,江曙,等.基于深度学习的动漫风格迁移研究综述[J].长江信息通信,2022,35(02):37-39.
- [6] 杨玥,冯涛,梁虹,等.融合交叉注意力机制的图像任意风格

- 迁移[J].计算机科学,2022,49(S1):345-352+396.
- [7] 张颖涛,张杰,张睿,等.全局信息引导的真实图像风格迁移[J].计算机科学,2022,49(07):100-105.
- [8] 刘建峰,钟国韵.基于神经网络的图像风格迁移研究综述[J].电子技术应用,2022,48(06):14-18.
- [9] 林晓,屈时操,黄伟,等.显著区域保留的图像风格迁移算法[J].图学报,2021,42(02):190-197.
- [10] 毛琳,王萌,杨大伟.内容特征一致性风格迁移网络[J].计算机辅助设计与图形学学报,2022,34(06):892-900.

特定于用户的信息。例如,它能够提取“获得奥斯卡奖的演员”等信息。第三,本文还提出一种注意力机制,它根据不同视图的影响聚合来自它们的知识。本文在电影数据集上进行了实验,并且与几个基线模型进行比较,证明了本文提出的模型的有效性。

## 2 基于知识图谱的推荐模型

在本节中,本文将展示本文提出的模型—GAMAT。模型的整体框架如图1所示。GAMAT有五个构建模块:用户-项目交互矩阵嵌入层、图注意力机制的项目视图嵌入层、多视图多头记忆注意力层(M3-layer)、视图注意力层和预测层。

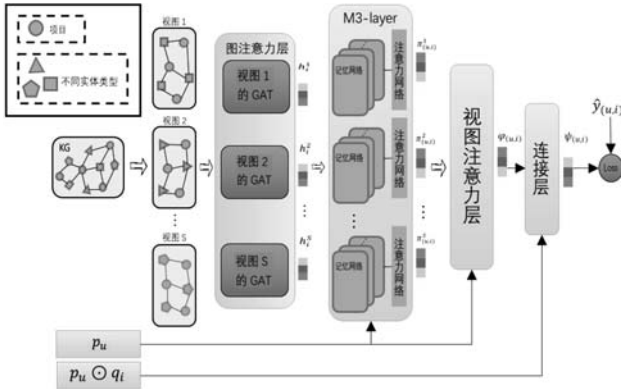


图1 GAMAT 的整体框架

### 2.1 GAMAT 模型

用户-项目交互矩阵嵌入层。这个模块负责从用户-项目交互矩阵中为用户和项目构造嵌入。设  $P$  和  $Q$  分别为用户和项目嵌入矩阵,  $x_u$  和  $z_i$  分别表示用户  $u$  和项目  $i$  的 one-hot 编码表示。本文通过计算能够得到用户嵌入  $p_u$  和项目嵌入  $q_i$ :

$$p_u = P^T x_u, \quad q_i = Q^T z_i \quad (1)$$

图注意力机制的项目视图级嵌入层(GAT Layers)。这个模块负责为来自知识图谱的项目构造初始的视图级嵌入,每个嵌入代表不同的视图。本文首先提取涉及不同实体类型的项目的子图。例如,电影能够和“演员”以及“分类”联系在一起。因此,本文可以构建两个项目视图子图,其中一个子图中只有电影和演员节点,另一个子图只有电影和分类节点,如图1所示。

定义1: 知识图谱。本文将知识图谱定义为一个有向异构图  $G(V, E)$ , 其中  $V = \{c_1, c_2, \dots\}$  表示所有节点的集合;  $E = \{r_1, r_2, \dots\}$  表示所有边的集合。

定义2: 项目视图子图。对于每个视图  $s$ , 本文将定义为项目视图子图  $G_s(V_s, E_s)$ , 其中  $V_s \subseteq V$ ,  $E_s \subseteq E$ 。

本文分别在上述提取到的这些子图上采用多层图注意力机制为这些项目构造第一层视图级嵌入。设  $W_G^l$  和  $c_G^l$  是第  $l$  层的权重矩阵和向量,  $N(i)$  是节点  $i$  的邻居集合。项目的视图级嵌入表示为:

$$f_i^l = \sum_{k \in N(i)} \alpha_{(i,k)}^l W_G^l f_k^{l-1} \quad (2)$$

$$\alpha_{(i,k)}^l = \frac{\exp(a(c_G^l \cdot [W_G^l f_i^{l-1} || W_G^l f_k^{l-1}]))}{\sum_{k \in N(i)} \exp(a(c_G^l \cdot [W_G^l f_i^{l-1} || W_G^l f_k^{l-1}]))} \quad (3)$$

$f_i^l$  是第  $l$  层的项目  $i$  的表示,  $||$  表示连接操作,  $f_i^0 = z_k$  是知

识图谱中项目的 one-hot 编码表示,  $\alpha_{(i,k)}^l$  表示节点  $k$  对第  $l$  层的节点  $i$  的影响因子。每个图注意力层中有多个注意力头作用于节点表示, 然后本文对来自不同注意力头的嵌入进行连接操作, 为了便于解释, 本文用  $h_i^s$  表示来自 GAT 层的项目的最终嵌入。

因此, 本文获得了项目的多个嵌入, 这些为本文提供了项目  $i$  的不同视图  $\{h_i^s\}_{s=1}^S$ , 其中  $S$  表示视图的总数。本文把单个视图的所有节点的嵌入的集合表示为  $h^s = [h_1^s, h_2^s, \dots, h_n^s]$ 。

多视图多头记忆注意力层(M3-layer)。M3-layer 将从之前的 GAT 层获得的嵌入  $\{h^s\}_{s=1}^S$  作为输入, 分别从每个视图  $s$  中专门地提取用户特定的成分。正如本文之前讨论的, 用户可能更偏好于奥斯卡获奖演员演的电影。因此, 本层的目标是使用多个记忆注意力网络来捕获这种更深层的信息, 然后使用注意力网络聚合这种信息。

在这个过程中, 本文为每个视图使用多个记忆网络来捕捉影响组件的不同概念。另外, 本文还利用注意力机制来为每个视图聚集这些构造的特定于用户的项目嵌入, 如图1所示。本文首先说明在一个视图和一个记忆注意力网络中发生了什么, 然后在其他视图和不同的记忆注意力头中分别遵循相同的程序。

设  $\mu = \{\mu_1, \mu_2, \dots, \mu_\tau\}$ ,  $\mu_j \in \mathbb{R}^d$  和  $k = \{k_1, k_2, \dots, k_\tau\}$ ,  $k_j \in \mathbb{R}^d$ , 分别表示记忆组件和其相对应的键,  $\tau$  表示记忆单元的总数。首先, 本文得到用户和项目视图的规范化表示  $\phi_{(u,i)}$ :

$$\phi_{(u,i)} = \frac{p_u \odot h_i}{\|p_u\| \|h_i\|} \quad (4)$$

其中  $\phi_{(u,i)} \in \mathbb{R}^d$ ,  $\odot$  表示 Hadamard 乘积。然后, 将其与键  $k$  结合以提供所需要的影响值, 即注意力值:

$$\beta_{(u,i)}^t = \frac{\exp(\phi_{(u,i)} \cdot k_t)}{\sum_{t=1}^{\tau} \exp(\phi_{(u,i)} \cdot k_t)} \quad (5)$$

这些影响值提供了每个组件对用户兴趣的贡献程度的影响分数。首先, 本文提取所需的信息  $\bar{\mu}_i^t$ , 然后, 把影响值和记忆单元一起使用。本文获取项目表示  $\bar{h}_{(u,i)}$ :

$$\bar{\mu}_i^t = \mu_t \odot h_i, \quad \bar{h}_{(u,i)} = \sum_{t=1}^{\tau} \beta_{(u,i)}^t \bar{\mu}_i^t \quad (6)$$

这里  $\bar{h}_{(u,i)} \in \mathbb{R}^d$  表示来自单个记忆力网络的特定于用户的项目视图表示, 这是在一个视图和一个记忆注意力网络中得到的, 在多个视图和多个注意力头(索引为  $\gamma$ ) 中本文能够得到  $\bar{h}_{(u,i)}^{(s,\gamma)}$ , 然后本文按视图将这些表示聚集到  $\pi_{(u,i)}^S$  中去:

$$\pi_{(u,i)}^S = \mathcal{A}(\{\bar{h}_{(u,i)}^{(s,\gamma)}\}_{\gamma=1}^{\Gamma}, p_u) \quad (7)$$

其中,  $\Gamma$  表示记忆注意力网络的总数,  $\mathcal{A}(\cdot)$  表示注意力网络, 它被定义为:

$$\mathcal{A}(p_u, \{x^\gamma\}_{\gamma=1}^{\Gamma}) = \sum_{\gamma=1}^{\Gamma} \vartheta_{u\gamma} x_{u\gamma} \quad (8)$$

$$\vartheta_{u\gamma} = \frac{\exp(\text{score}(p_u, x^\gamma))}{\sum_{\gamma'=1}^{\Gamma} \exp(\text{score}(x^{\gamma'}, p_u))}, \quad \text{score}(p_u, x^\gamma) = p_u^T W_{\mathcal{A}} x^\gamma \quad (9)$$

$x^\gamma$  是一个向量,  $\vartheta_{u\gamma}$  表示  $\gamma$  对用户  $u$  的影响值,  $W_{\mathcal{A}}$  是注意力网络的权重矩阵。注意, 记忆单元和键是随机初始化的, 并且权重在输入之间共享。

视图注意力层。上一层提供了与不同视图相关的项目嵌

入 $\{\pi_{(u,i)}^s\}_{s=1}^S$ ,由按组件提取的特定于用户的信息组成。本层的目标是按视图聚合存在于项目嵌入中的信息。不同的用户会受到不同项目视图的影响,例如,电影的类型或电影中的演员。因此,我将这些不同的视图组合为:

$$\varphi_{(u,i)} = \mathcal{A}\left(p_u, \{\pi_{(u,i)}^s\}_{s=1}^S\right) \quad (10)$$

$\varphi_{(u,i)}$ 是从知识图谱中提取的最终项目视图表示。

预测层。这一层负责聚集来自用户-项目交互矩阵和知识图谱的信息,并提供用户未来可能与每个项目交互的预测概率。首先,本文对用户-项目交互信息和知识图谱的表示进行连接操作,然后,本文将这些表示传递给全连接层,最后本文得到的预测概率为:

$$\hat{y}_{(u,i)} = g(\psi_{(u,i)}), \psi_{(u,i)} = (p_u \odot q_i) \parallel (p_u \odot \varphi_{(u,i)}) \quad (11)$$

$g(x) = \sigma(\dots \sigma(W_2 \sigma(W_1 x + b_1) + b_2) \dots)$ 表示一个全连接网络, $\sigma(\cdot)$ 一个激活函数, $W_i$ 和 $b_i$ 分别表示权重矩阵和偏差向量。

## 2.2 损失函数

由于本文的评分是隐性的,本文采用交叉熵损失函数。完全损失函数如下:

$$\min_{\theta} \mathcal{L}(\theta) = - \sum_{(u,i) \in \mathcal{D}} y_{(u,i)} \ln \hat{y}_{(u,i)} + (1 - y_{(u,i)}) \ln(1 - \hat{y}_{(u,i)}) + \lambda \mathcal{R}(\theta) \quad (12)$$

其中 $\theta$ 由所有模型参数组成, $\mathcal{R}(\cdot)$ 是正则化, $\lambda$ 是一个非负超参数。本文在训练期间采用负采样策略, $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}_{samp}^-$ , $\mathcal{D}^+ := \{(u,i) \in \Omega\}$ , $\mathcal{D}_{samp}^- := \{(u,i') \notin \Omega\}$ , $\mathcal{D}_{samp}^-$ 是通过负采样得到的。

## 3 实验

### 3.1 实验数据集

实验部分使用来自电影领域的真实数据集: Movielens-100k, 简称 ML-100K。该数据集包含了 MovieLens 网站上由 943 个用户和 1600 多个电影组成的大约 10 万个用户对电影的评分数据,评分数据的分值范围为 1~5 之间的整数,如表 1 所示。

表 1 实验数据集

数据集	用户	项目	特征	评分
ML-100K	943	1682	Users/Items	100,000

### 3.2 评价指标和实验对比分析

本实验采用归一化折损累计增益( $NDCG@N$ )对模型的性能进行衡量。

$$NDCG = \frac{1}{\#users} \sum_{i=1}^{\#users} \frac{1}{\log_2(p_i + 1)} \quad (13)$$

其中 $p_i$ 是第 $i$ 个需求项在模型推荐的项目列表中的位置,若第 $i$ 个需求项不在推荐列表中,则 $\frac{1}{\log_2(p_i + 1)}$ 为 0。 $NDCG$ 的值介于 0 和 1 之间,越接近 1 说明推荐越准确。

在本实验中,本文按照 6:2:2 的比例随机划分数据集来构建训练集,验证集和测试集,本文为每一个用户分别提供一个随机选择的项目用于验证集和测试集,由于在验证期间很难对每个用户的所有可用项目进行排名,因此本文为每个用户抽取 50 个没有交互过的项目,用来与验证集和测试集中所保留的项目进行对比。本文将这个过程重复 5 次,得到 5 种不同的

分割,最后将这 5 种分割的平均值和标准差作为最终结果。

表 2 展示了本文提出的模型与 FM、BPR、CKE 和 RippleNet 这四种经典推荐算法的对比实验结果。

表 2 不同模型在 ML-100K 数据集上的表现

模型	NDCG@5
MF[5]	0.4715
CKE[6]	0.4912
RippleNet[7]	0.5265
GAMAT	0.5317

本文使用验证集对超参数进行调优,并报告其对应的测试集的性能。根据验证集的表现,本文将 Movielens-100K 电影数据集的嵌入维度设置为 64,学习效率 $\alpha$ 设置为 0.002,负样本数量设置为 3,使用 Xavier 初始化将 mini-batch 大小设置为 2048,使用 RMSprop 优化器,将 Dropout 设置为 0.5。本文采用一个两层的 GAT 网络,第一层和第二层的注意力头数分别是 6 和 4,本文将 M3-layer 中的注意力头数 $\Gamma$ 设置为 4。

## 4 结语

本文提出了 GAMAT—用于 Top-N 推荐的图和记忆注意力的神经网络模型,用图注意力机制替代离线计算相似性或计算交换矩阵的繁琐过程,这个技术将本文的模型设计为端到端的模型。另外,本文还提出一种多视图多头记忆注意力层,用它来提取更细粒度的特定于用户的特征。本文在真实的电影数据集上进行实验,实验证明了提出的模型比 Top-N 推荐基线模型的有效性。

### 参考文献:

- [1] Koren, Y., Bell, R. (2015). Advances in Collaborative Filtering. In: Ricci, F., Rokach, L., Shapira, B. (eds) Recommender Systems Handbook. Springer, Boston, MA. [https://doi.org/10.1007/978-1-4899-7637-6\\_3](https://doi.org/10.1007/978-1-4899-7637-6_3)
- [2] Wang X, He X, Wang M, et al. Neural Graph Collaborative Filtering[J]. ACM, 2019.
- [3] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norrick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In Proceedings of the 7th ACM International Conference on Web Search and Data Mining. 283-292.
- [4] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 635-644.
- [5] Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM, pp. 263-272. IEEE (2008).
- [6] Zhang F, Yuan N J, Lian D, et al. Collaborative Knowledge Base Embedding for Recommender Systems[C]// the 22nd ACM SIGKDD International Conference. ACM, 2016.
- [7] Wang, H., et al. "RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems." ACM (2018).