

面向社交推荐的自适应高阶隐式关系建模*

李邵莹¹, 孟丹², 孔超¹, 张丽平¹, 徐辰³

¹(安徽工程大学 计算机与信息学院, 安徽 芜湖 241000)

²(同盾科技有限公司 人工智能研究院, 上海 200032)

³(华东师范大学 数据科学与工程学院, 上海 200062)

通信作者: 孔超, E-mail: kongchao@ahpu.edu.cn



摘要: 近年来, 社交推荐的研究主要聚焦于社交网络中显式、隐式关系的联合建模, 却忽视了高阶隐式关系并非对每个用户都同等重要这一特殊现象. 高阶隐式关系对一个有着足够多邻居的用户与一个仅有少量邻居的用户重要性存在明显差异. 此外, 由于社交关系建立的随机性, 显式关系并不总是可用的. 提出了一种新的自适应高阶隐式关系建模方法 (adaptive high-order implicit relations modeling, AHIRM), 该模型由 3 个部分组成: 首先, 过滤不可靠关系且识别出潜在可靠关系. 旨在避免不可靠关系带来的负面影响, 并部分缓解数据稀疏的问题; 其次, 设计自适应随机游走算法, 结合规范化后的节点中心度为用户捕获不同阶数的邻居, 构建用户间的高阶隐式关系, 进而重构社交网络; 最后, 运用图卷积网络 (graph convolutional network, GCN) 聚合邻居节点信息, 更新用户嵌入, 实现高阶隐式关系建模, 从而进一步缓解数据稀疏问题. 在建模过程中, 同时考虑到社交结构和个人偏好的影响, 模拟并保留了社交影响传播的过程. 在 LastFM、Douban 和 Gowalla 这 3 个数据集上与相关算法做了对比验证, 结果证实了该模型的有效性和合理性.

关键词: 高阶隐式关系建模; 自适应随机游走; 社交网络; 图卷积网络; 社交推荐

中图法分类号: TP311

中文引用格式: 李邵莹, 孟丹, 孔超, 张丽平, 徐辰. 面向社交推荐的自适应高阶隐式关系建模. 软件学报, 2023, 34(10): 4851–4869. <http://www.jos.org.cn/1000-9825/6662.htm>

英文引用格式: Li SY, Meng D, Kong C, Zhang LP, Xu C. Adaptive High-order Implicit Relations Modeling for Social Recommendation. Ruan Jian Xue Bao/Journal of Software, 2023, 34(10): 4851–4869 (in Chinese). <http://www.jos.org.cn/1000-9825/6662.htm>

Adaptive High-order Implicit Relations Modeling for Social Recommendation

LI Shao-Ying¹, MENG Dan², KONG Chao¹, ZHANG Li-Ping¹, XU Chen³

¹(School of Computer and Information, Anhui Polytechnic University, Wuhu 241000, China)

²(AI Institute, Tongdun Technology Co. Ltd., Shanghai 200032, China)

³(School of Data Science and Engineering, East China Normal University, Shanghai 200062, China)

Abstract: Recent research studies on social recommendation have focused on the joint modeling of the explicit and implicit relations in social networks and overlooked the special phenomenon that high-order implicit relations are not equally important to each user. The importance of high-order implicit relations to users with plenty of neighbors differs greatly from that to users with few neighbors. In addition, due to the randomness of social relation construction, explicit relations are not always available. This study proposes a novel adaptive high-order implicit relations modeling (AHIRM) method, and the model consists of three components. Specifically, unreliable relations are filtered, and potential reliable relations are identified, thereby mitigating the adverse effects of unreliable relations and

* 基金项目: 国家自然科学基金 (61902001)

收稿时间: 2021-07-31; 修改时间: 2021-11-14; 采用时间: 2022-02-16; jos 在线出版时间: 2022-12-28

CNKI 网络首发时间: 2022-12-30

alleviating the data sparsity issue. Then, an adaptive random walk algorithm is designed to capture neighbors at different orders for users according to normalized node centrality, construct high-order implicit relations among the users, and ultimately reconstruct the social network. Finally, the graph convolutional network (GCN) is employed to aggregate information about neighbor nodes. User embeddings are thereby updated to model the high-order implicit relations and further alleviate the data sparsity issue. The influence of social structure and personal preference are both considered during modeling, and the process of social influence propagation is simulated and retained. Comparative verification of the proposed model and the existing algorithms are conducted on the LastFM, Douban, and Gowalla datasets, and the results verify the effectiveness and rationality of the proposed AHIRM model.

Key words: high-order implicit relations modeling; adaptive random walk; social network; graph convolutional network (GCN); social recommendation

社交媒体的普及,产生了海量社交网络数据. 社交推荐的目的是利用社交网络数据中存在的社交关系为每个用户生成个性化的推荐项目. 在此过程中,对社交关系的建模至关重要. 典型的社交关系建模方法,如: TrustSVD^[1]和 ContextMF^[2]同时考虑了评分和信任信息对用户建模的影响. 此外,后者还提出了一种新的概率矩阵分解模型来融合潜在空间中的个人偏好和交互信息. 以往的研究证实了结合社交关系建模方法,可以提高传统推荐系统的性能.

目前,现有的社交关系建模方法主要关注显式关系的建模^[1,3]. 这些方法在一定程度上提升了推荐系统的性能,并部分缓解了数据稀疏问题. 然而,在现实世界中,用户的选择和偏好不仅会受到显式关系的影响,还会受到高阶隐式关系的影响,这称为社交影响传播现象^[4]. 图 1 以电影推荐为例,展示了这一传播学现象. 假定用户 u_1 观看并喜欢上一部电影,并将其推荐给了自己的朋友用户 u_2 . 同理,用户 u_2 也会有一定的概率把这部电影推荐给自己的朋友 u_3 ,这就形成了一条传播链: $u_1 \rightarrow u_2 \rightarrow u_3$. 因此, u_1 的选择和偏好间接影响了 u_3 的决定. 此外,现实世界的社交网络数据呈现长尾分布的现象,大部分用户的一阶邻居数量很有限. 因此,近年来,研究人员提出多种高阶隐式关系建模方法^[4,5]. 然而,以往大多数研究都是基于高阶隐式关系对每个用户都具有同等重要性的假设,使得它们的适用性和性能受到很大限制. 尽管上述方法能够有效地提升推荐系统性能并被广泛采用,但是这些方法对高阶隐式关系的利用欠佳,推荐准确性有待提升. 本文主要研究面向社交推荐的高阶隐式关系建模问题. 这是一项具有挑战性的任务,主要有以下几点原因.

(1) 在现实的社交网络中,数据具有稀疏性并且存在噪声.

(2) 传统的社交推荐方法很少考虑到高阶隐式关系对不同用户具有不同重要性这一现象. 直观上看,对于具有很多一阶邻居的节点,建模显式关系可能就足以表征用户. 因此,高阶隐式关系对此类用户而言,可能就不那么重要,而对于具有少量一阶邻居的用户而言,建模高阶隐式关系的重要性则不言而喻.

(3) 在高阶隐式关系建模中,很难模拟和保留社交影响传播过程.

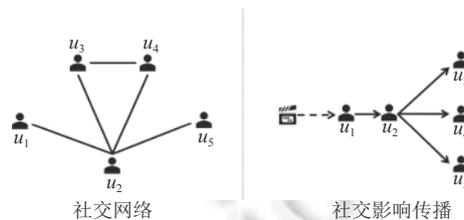


图 1 社交影响传播示例 (实线表示用户间的社交关系,虚线表示用户和项目的交互关系)

针对上述挑战,本文提出的 AHIRM 模型能够自适应地对高阶隐式关系进行建模. 此模型不仅能有效过滤不可靠关系,挖掘出用户间的潜在可靠关系,还能够自适应地构建用户间的高阶隐式关系,并在建模过程中同时考虑社交结构和个人偏好的影响,从而提高推荐性能. 具体而言,该模型由 3 个部分组成: 首先,本文首次提出了潜在用户候选对的概念,利用杰卡德和余弦相似度计算函数去除用户间的不可靠关系并发现一些新的潜在可靠关系,以实现噪声数据的过滤与潜在可靠关系的生成; 其次,考虑到高阶隐式关系建模在不同用户间的差异性,设计了一个自适应随机游走生成器,根据节点的重要性自适应地生成控制随机游走倾向的进出参数,个性化地为用户捕获不同阶数的邻居,使生成的节点序列更加符合社交网络中节点的传播特性,构建用户间的高阶隐式关系,进而进行

图重构; 最后, 为了进一步缓解数据稀疏性带来的不利影响, 本文采用基于 GCN 的用户表示学习组件. 通过为邻居节点分配不同的权重, 灵活地调整学习过程, 能够在更细的粒度上区分不同邻居节点的重要性, 生成更加准确的用户嵌入. 在此过程中, 模拟并保留了社交影响传播过程. 据了解, 还没有相关研究过多关注高阶隐式关系建模在不同用户间的差异性. 在 3 个真实数据集上的大量实验结果表明该模型的有效性和合理性.

本文第 1 节对推荐系统和高阶隐式关系建模的相关工作进行回顾. 第 2 节对基本概念进行描述并给出问题定义. 第 3 节详细介绍本文算法. 第 4 节给出在真实数据集上的实验结果, 并对实验结果加以分析. 第 5 节对本文工作进行总结和展望.

1 相关工作

本文的工作涉及基于图神经网络 (graph neural network, GNN) 的推荐系统和高阶隐式关系建模两个方面, 本节对这两方面工作进行总结与分析.

1.1 基于 GNN 的推荐系统

推荐中的大多数数据都可以抽象成图结构. 近年来, 由于 GNN 在图数据的表示学习方面展现出的卓越性能, 越来越多的人开始研究基于 GNN 的推荐系统. 不同类型的信息网络可以表征为不同结构的图, 例如: 消费网络具有两种类型的节点, 可以抽象成用户-项目二部图; 社交网络是一张同构图, 只包含一种类型的节点; 而知识图谱^[6,7]是异构图, 具有多种类型的节点和边. Malik 等人^[8]根据使用信息的类型, 对现有的基于 GNN 的推荐系统进行了总结, 将推荐系统分为两类: 基于用户-项目交互信息的推荐系统和基于社交网络增强的推荐系统.

(1) 基于用户-项目交互信息的推荐系统

为了充分发挥 GNN 在二部图上的建模优势, 研究人员需要考虑怎样将交互项目以及用户的信息传递给用户和项目, 以获得最终的表示. 图的结构对传递信息的类型和范围至关重要. Sun 等人提出了 NIA-GCN 模型^[9], 直接在原始用户-项目二部图上应用 GNN 进行推荐. 然而, 此方法存在以下两方面的限制: ① 仅使用原始用户-项目二部图, 所包含的信息不足以学习出一个好的用户-项目表示; ② 当数据量非常大的时候, 在原始图上传播信息的开销巨大. 因此, 为了丰富原有图结构, Multi-GCCF^[10]通过建立用户-用户图和项目-项目图来提高推荐质量. 针对在原始图上传播信息的挑战, 研究人员们提出了图采样方法, 让模型能够在大规模图数据上实现高效、准确的图神经网络训练. Ying 等人^[11]提出了图卷积网络算法 PinSage, 结合随机游走和图卷积操作生成包含图结构和节点特征信息的节点嵌入. 这类模型的性能取决于采样策略, 本文对采样策略进行了进一步的研究和优化. 聚合策略也是图结构中信息传播的重要因素, 它决定了邻居传播的信息量. HashGNN^[12]使用一种最直接的平均池化聚合策略, 不加区分地看待每个邻居. 这种方法的实现方式简单, 但现实世界不同邻居对个人偏好的影响程度存在很大差异. 因此, LR-GCCF 模型^[13]依据图结构为节点分配不同的权重. 然而这些方法仅根据图结构决定权重, 而忽视了用户的个人偏好. NGCF^[14]依据人们更有可能向与自己喜好相似的朋友推荐符合用户兴趣的项目这一心理学现象, 提出运用注意力机制区分邻居节点的重要性, 以达到更加准确的推荐结果.

(2) 基于社交网络增强的推荐系统

社交网络增强的推荐系统旨在利用社交关系所隐含的用户相关性来提高传统推荐系统的性能. 由于社交推荐系统引入了多种社交信息的特性, 能获取到更准确的个人偏好, 高效解决冷启动、数据稀疏等问题^[15,16]. 因此, 近年来, 社交推荐的研究已经成为一个热门话题. 例如, LOCABAL 模型^[17]运用局部和全局社会信息进行推荐. TrustSVD^[1], 一种基于信任的矩阵分解技术, 综合使用评级和信任信息进行推荐. ARSE^[18]是一种基于递归神经网络结构的社会推荐模型, 主要从动态兴趣和静态兴趣两方面对用户偏好进行建模. 虽然上述工作部分缓解了数据稀疏的问题, 但它们主要关注于显式关系的建模, 而忽略了高阶隐式关系的影响. DiffNet^[19]模型能够模拟社交网络中的社会扩散过程, 从而捕获用户的高阶关系以生成更加准确的用户和项目嵌入. DiffNet++^[20]是对 DiffNet 的扩展, 此方法能在统一的框架内联合建模影响力扩散和兴趣扩散过程. 此类方法考虑到高阶隐式关系, 但并未考虑到高阶隐式关系对不同用户具有不同重要性. 由于 GNN 能够建模用户社交影响的递归扩散过程, 它已成为社交

关系建模的研究热点. 其中, 具有代表性的工作如 NGCF^[14]、GraphRec^[21]等证实了运用 GNN 进行建模能够有效增强模型性能. 在建模过程中, 一个重要的问题是邻居影响力的建模. 然而, 大多数情况下, 社交网络中仅描述了用户之间的关系, 却没有刻画社交关系的强度. 因此, 研究人员进一步对社交影响强度进行了研究. DiffNet++^[20]运用注意力机制区分不同邻居社交影响力强度, 提高了模型的整体性能. 然而, 上述方法没有考虑到不同邻居由于所处环境以及个人偏好等因素的影响, 其重要性也存在明显差异. 本文所提出的方法将邻居的权重与社交结构、个人偏好联系起来, 既表征了用户的社交关系, 又表征了用户偏好, 以期提升推荐系统性能.

1.2 高阶隐式关系建模

作为一种普遍存在的社交结构, 隐式关系的挖掘有着广泛的应用. 其中, 高阶隐式关系建模是一个非常热门的研究方向. 例如, TFBA^[22], 一种基于张量因子分解的高阶关系模式归纳法. 联合分解从高阶张量导出的多个低阶张量以克服数据稀疏的问题. 高阶隐式关系建模的关键之一是对社交影响传播过程的表征. 受到图卷积神经网络 (GCN) 的逐层更新机制的启发, 人们提出运用 GCN 建模社交影响传播过程. GCN 将卷积运算的思想扩展到图数据上. 现有的 GCN 方法主要分为两类: 谱方法和非谱方法. 谱方法希望借助图谱理论实现图上的卷积操作. 在图形信号处理方面, 通过引入滤波器, 谱方法^[23]将图卷积运算定义为从图形信号中去除噪声. 非谱方法^[24]基于节点的空间关系定义图卷积操作, 将图卷积运算定义为聚合来自相邻节点的特征信息. 消息传递神经网络 (MPNN)^[25]将图卷积定义为消息传递过程, 通过 k 步消息传递的迭代让信息进一步传播. 由于 GCN 训练时依赖于全局的图结构, 不利于大规模数据的处理. 为了解决这个问题, GraphSAGE^[26]和 PinSage^[11]通过采样策略聚合节点的部分邻域信息, 有效生成节点嵌入.

本文的工作深受这些开创性工作的启发. 由于本文提出的任务涉及特定的研究挑战, 以往的工作很少考虑到高阶隐式关系对不同用户具有不同重要性这一特殊现象. 因此, 本文的研究重点在于如何为每个用户自适应地制定相应的高阶隐式关系聚合策略.

2 问题定义

2.1 基本概念

定义 1. 社交网络. 用图 $G = (U, E)$ 来表示社交网络. 其中, $U = \{u_1, u_2, \dots, u_m\} (|U| = m)$ 是节点 (用户) 集合, E 表示边的集合. 若 $(u_i, u_j) \in E$, 表明节点 u_i 和 u_j 之间存在关系, (u_i, u_j) 又称为用户对. 对于某一节点 u_i , N_{u_i} 表示用户 u_i 的一阶邻居集合.

定义 2. 结构相似度. 结构相似度定义为两个用户在社交结构上的相似程度. 采用杰卡德相似系数计算用户 u_i 和 u_j 之间的结构相似性:

$$Sim_structure(u_i, u_j) = \frac{|N_{u_i} \cap N_{u_j}|}{|N_{u_i} \cup N_{u_j}|} \quad (1)$$

其中, N_{u_i} 和 N_{u_j} 是用户 u_i 和 u_j 的一阶邻居集合.

定义 3. 偏好相似度. 偏好相似度定义为两个用户在历史交互项目上的相似程度. 采用余弦相似度计算用户 u_i 和 u_j 之间的偏好相似度, 其公式为:

$$Sim_preference(u_i, u_j) = \frac{\vec{r}_i \cdot \vec{r}_j}{\|\vec{r}_i\| \cdot \|\vec{r}_j\|} \quad (2)$$

其中, \vec{r}_i , \vec{r}_j 分别表示用户 u_i 和 u_j 给出的项目评分向量.

定义 4. 整体相似度. 整体相似度表示用户在社交网络 and 用户-项目交互网络上的综合相似度. 通过用户的社交结构和个人偏好进行计算, 同时使用权重动态决定各部分影响程度, 即通过对结构相似度 $Sim_structure(u_i, u_j)$ 和偏好相似度 $Sim_preference(u_i, u_j)$ 的加权求和, 使用 Sigmoid 函数将结果转换为 u_i 和 u_j 的整体相似度 $Sim(u_i, u_j)$:

$$Sim(u_i, u_j) = \sigma(w_1 Sim_structure(u_i, u_j) + w_2 Sim_preference(u_i, u_j)) \quad (3)$$

其中, $\sigma(z) = \frac{1}{1 + e^{-z}}$, w_1 和 w_2 为控制两部分影响程度的参数, 通过实验训练得出.

定义 5. 潜在用户候选对. 若用户 u_i 和 u_j 在社交网络中不存在关系, 但可能存在一定的偏好相似, 称其为潜在用户候选对, 记为 $\langle u_i, u_j \rangle$. 当偏好相似度大于一定阈值时, u_i, u_j 之间的关系称为潜在可靠关系.

定义 6. k 阶邻居集 $N_{u_i}^{(k)}$. 若用户 u_i 有 k 阶邻居 u_r , 则 u_r 需满足 $\exists(u_j, u_r) \in E$, 且 $u_j \in N_{u_i}^{k-1}$. 例如, 图 1 中用户 u_3 为用户 u_1 的二阶邻居, 在社交网络中存在边 (u_2, u_3) , 且 u_2 为 u_1 的一阶邻居.

定义 7. k 阶嵌入传播. 一阶嵌入传播可以定义为将每个节点的初始表示传播到其邻居节点, 得到其邻居节点的一阶嵌入传播向量, 记为 $\vec{P}_{ij}^{(1)}$. 将所有邻居节点的一阶嵌入传播向量进行聚合并结合节点自身嵌入可以得到节点的一阶嵌入表示 $\vec{u}_i^{(1)}$. 自然地, 将一阶嵌入传播定义扩展到 k 阶嵌入传播, 即重复执行此过程, 进一步传播聚合节点的 $k-1$ 阶嵌入表示得到 k 阶嵌入传播向量 $\vec{P}_{ij}^{(k)}$ 以及节点的 k 阶嵌入表示 $\vec{u}_i^{(k)}$.

如图 2 所示, 假定用户 u_i 存在 l 个邻居, 在一阶嵌入传播时, $u_{j_1}, u_{j_2}, \dots, u_{j_l}$ 节点的初始表示传播到用户 u_i , 得到节点 u_i 的一阶嵌入传播向量 $\vec{P}_{ij_1}^{(1)}, \vec{P}_{ij_2}^{(1)}, \dots, \vec{P}_{ij_l}^{(1)}$. 聚合所有邻居节点的传播嵌入, 并结合节点自身嵌入, 可得节点的一阶嵌入表示 $\vec{u}_i^{(1)}$. 此时, u_i 的一阶嵌入表示聚合了 u_i 所有一阶邻居节点的信息. 与此同时, 节点 $u_{j_1}, u_{j_2}, \dots, u_{j_l}$ 也都聚合其一阶邻居的信息, 得到每个节点的一阶嵌入表示. 在二阶嵌入传播时, 进一步传播聚合邻居节点的一阶嵌入表示. 此时, 邻居节点的一阶嵌入表示其实已经包含了其邻居节点的信息. 因此, 在二阶嵌入传播后, 用户 u_i 聚合了其邻居节点的邻居信息, 即二阶邻居的信息.

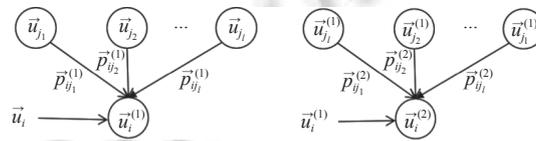


图 2 二阶嵌入传播示例

本文用 $U = \{u_1, u_2, \dots, u_m\} (|U| = m)$ 和 $I = \{i_1, i_2, \dots, i_n\} (|I| = n)$ 分别代表用户集合和项目集合. 此外, 采用用户-项目交互矩阵 $Y \in \mathbb{R}^{m \times n}$ 表示用户和项目之间的交互行为. 其中, 每个元素 $y_{u_i i_j} = 1$ 表明用户 u_i 和项目 i_j 之间存在交互行为 (如购买, 浏览, 点击等). 用户-用户邻接矩阵 $R \in \mathbb{R}^{m \times m}$ 用于表征社交网络 G , 描述用户之间的社交关系. 每个元素 $r_{ij} = 1$ 表明用户 u_i 和用户 u_j 之间存在关系. 高阶隐式关系建模旨在将社交网络中的每个用户映射到低维嵌入空间中, 得到每个用户 u_i 的向量表示 \vec{u}_i , 同时在嵌入空间中要保持节点之间的高阶隐式关系. 社交推荐则利用得到的用户表示 \vec{u}_i 和初始化得到的项目表示 \vec{i}_j , 针对每个用户将项目 $i_j \in I$ 映射为一个实值. 此问题可以形式化地定义为一个预测函数, 它的输入是用户集 U , 项目集 I , 用户-项目交互矩阵 Y , 和社交网络 G , 输出是预测函数 $\mathcal{F}(u, i, \theta, Y, G)$, 其中 θ 为模型参数, 将每个项目 $i_j \in I$ 映射为每个用户 $u_i \in U$ 的实值.

3 基于自适应高阶隐式关系建模的社交推荐

一个好的高阶隐式关系建模方法需要很好地表征现实世界中的网络. 为了在社交推荐场景下实现这一目标, 本文提出的自适应高阶隐式关系建模方法 (AHIRM) 包含以下 3 个部分.

(1) 结合用户的社交结构和个人偏好, 设计不可靠关系过滤及潜在可靠关系发现规则, 更新原始的社交网络, 旨在避免不可靠关系带来的负面影响, 并部分缓解数据稀疏的问题.

(2) 根据网络中节点的中心度, 设计自适应随机游走算法为用户捕获不同阶数的邻居, 构建用户间的高阶隐式关系, 进而重构社交网络.

(3) 基于 GCN 的逐层更新机制, 定义高阶隐式关系的嵌入传播过程, 建模社交影响传播过程, 以学习出更加准确的用户表示.

AHIRM 的整体框架如图 3 所示. 首先, 结合用户的社交结构和个人偏好进行相似度计算, 滤除社交网络中的不可靠关系, 并补全新的潜在可靠关系, 构建一个新的社交网络; 其次, 对更新后的社交网络执行本文提出的自适应随机游走算法, 捕获不同阶数的邻居, 构建用户间的高阶隐式关系, 进而重构社交网络; 最后, 在用户表示学习模块定义了高阶隐式关系的嵌入传播过程, 并利用 GCN 来聚合高阶隐式关系, 从而更有效地对高阶隐式关系进行建模.

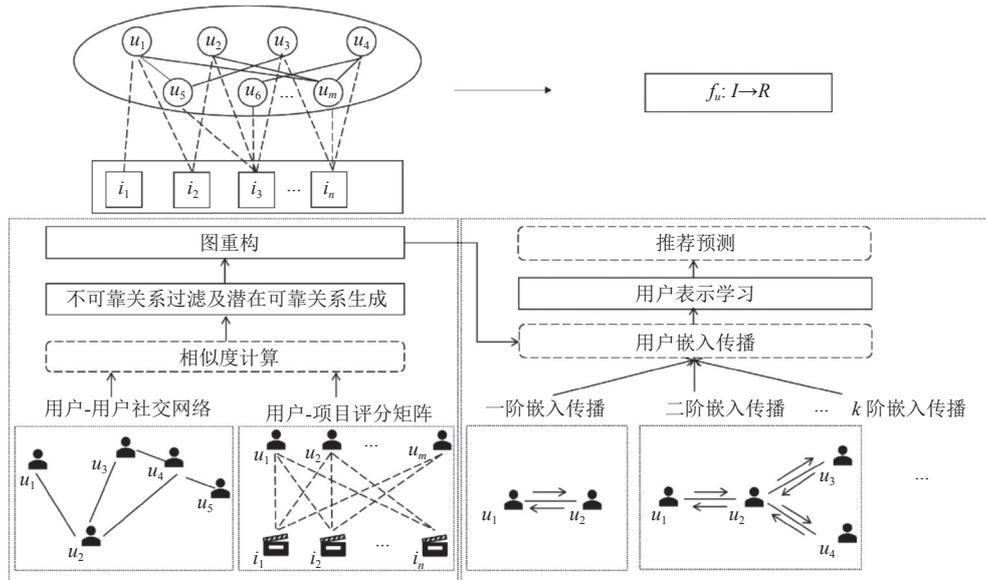


图3 AHIRM模型的整体框架(实线表示用户间的社交关系,虚线表示用户和项目的交互关系)

3.1 不可靠关系过滤及潜在可靠关系生成

通常,将社交关系应用于推荐系统能够提升传统推荐系统的性能。然而, Tang 等人的调查^[27]表明人们高估了此方法所带来的性能提升。主要原因在于:(1) 社交关系质量参差不齐,建立关系的低成本使得一个用户在社交网络中存在很多个朋友。然而,并不是所有社交关系都有利于推荐任务。例如,用户可能存在一些不可靠的朋友,他们之间既没有共同好友,偏好也完全不相似,不加区分的使用所有显式关系,可能会引入噪声数据,对推荐系统性能产生负面影响。此外,由于垃圾邮件发送者以及僵尸程序的存在,用户之间可能会建立虚假的关系,这些关系也会降低推荐性能;(2) 社交关系和用户-项目交互数据一样存在稀疏性。

针对上述两点原因,本文设计了不可靠关系过滤及潜在可靠关系生成模块。通过过滤不可靠关系和生成潜在可靠关系来解决这个问题。针对噪声数据,根据公式(3),通过设置权重 w_1 和 w_2 ,将社交结构和个人偏好进行融合,考虑结构相似度和偏好相似度来综合衡量用户的整体相似度,由此识别出不可靠关系,缓解不可靠关系带来的负面影响,以期提高推荐的准确度。具体地,根据公式(1)–公式(3)计算用户 u_i 和 u_j 之间的整体相似性,通过将其与训练所得的过滤阈值 s 相比较,从该用户的邻居集合中移除低于过滤阈值 s 的不可靠关系。然而,经过过滤后的社交关系仍可能存在数据稀疏的问题。根据协同过滤的思想,当用户间有相似的兴趣时,可能会做出相似的选择。本文引入潜在可靠关系这一定义,通过比较潜在用户候选对的偏好相似度和设定的偏好阈值 t 挖掘出潜在可靠关系,并将其添加到用户邻居集合中,进而重构该集合。本文涉及的阈值均通过实验训练所得。总的来说,不可靠关系过滤及潜在可靠关系生成模块通过相似度计算,实现了不可靠关系的过滤以及潜在可靠关系的发现,部分缓解了不可靠关系带来的负面影响以及数据稀疏问题。该模块的伪代码如下算法1所示。

算法1. 不可靠关系过滤及潜在可靠关系生成。

输入: 用户集 U , 项目集 I , 用户-项目交互矩阵 Y , 用户-用户邻接矩阵 R ;

输出: 滤除不可靠关系, 添加潜在可靠关系的用户邻居集 N_{u_i} 。

1. 根据公式(1)计算用户 u_i 和 u_j 间的结构相似度 $Sim_structure(u_i, u_j)$ 。
2. 根据公式(2)计算用户 u_i 和 u_j 间的偏好相似度 $Sim_preference(u_i, u_j)$ 。
3. 根据公式(3)计算用户 u_i 和 u_j 间的整体相似度 $Sim(u_i, u_j)$ 。

4. 重复步骤 1-3, 直至遍历所有用户对.
5. 对于每一个用户 u_i , 若用户 u_j 是 u_i 的邻居节点, 判断用户 u_i 和 u_j 之间的整体相似性与过滤阈值 s 的大小.
6. 若用户 u_i 和 u_j 之间的整体相似度 $Sim(u_i, u_j)$ 小于过滤阈值 s , 从用户邻居集合 N_{u_i} 中移除 u_j .
7. 对于每一个用户 u_i , 若用户 u_j 不是 u_i 的邻居节点, 判断用户 u_i 和 u_j 之间的偏好相似性与偏好阈值 t 的大小.
8. 若用户 u_i 和 u_j 之间的偏好相似度 $Sim_preference(u_i, u_j)$ 大于偏好阈值 t , 将 u_j 添加到用户邻居集合 N_{u_i} 中.
9. 返回用户邻居集合 N_{u_i} .

3.2 图重构

过滤不可靠关系、生成潜在可靠关系部分缓解了数据稀疏性问题. 然而, 仅利用用户的个人偏好和显式关系, 很难捕获到节点之间更高阶和更复杂的连接模式. 事实上, 先前的研究已经指出了高阶隐式关系和推荐系统性能之间的潜在联系^[4], 因此本文设计了一个能够自适应构建用户间高阶隐式关系的模型.

现有的高阶隐式关系建模方法并没有考虑真实世界的社交网络中不同用户在关系数量上可能存在的较大差异. 因此在聚合邻居信息时, 一般会采用固定阶数的方法为每个用户聚合相同阶数的邻居. 但是, 当一个用户 u_i 存在很多一阶邻居时, 用户 u_i 的最终嵌入会聚合更多各阶邻居的信息, 这可能会导致传播后的用户变得无法区分, 影响模型性能. 因此高阶隐式关系建模方法需要具备根据节点自身情况自适应地运用高阶隐式关系的能力. 考虑到高阶隐式关系对不同用户的差异性, 本文设计了一个自适应随机游走生成器捕获不同阶数的邻居, 构建用户间的高阶隐式关系. 确切地说, 图重构模块将两节点间的转移概率与节点的重要性联系起来, 从而控制随机游走的倾向, 节点重要性通过节点的中心度来评估. 对于一个节点, 其中心度越大, 游走策略更倾向于广度优先搜索 (BFS), 该节点所需的阶数就相对低; 反之, 更倾向于深度优先搜索 (DFS), 节点阶数则相对高. 具体地, 首先通过执行本文提出的自适应随机游走算法, 对社交网络进行转化, 生成融合节点间高阶隐式关系的语料库. 然后, 在此语料库中进行采样生成融合高阶隐式关系的社交网络.

随机游走是将网络转化为节点序列集的一种常见方法, 这种方法已被广泛用于一些网络嵌入方法中^[28,29]. 然而, 若不考虑到高阶隐式关系对不同用户具有不同重要性这一特殊现象, 直接在社交网络上执行随机游走, 则无法表征真实的社交网络. 因此, 此类方法生成的节点语料库质量有待提升. 本文提出的自适应随机游走生成器可以有效捕获不同阶数的邻居, 构建用户间的高阶隐式关系, 较好地刻画了这一现象. 其核心设计如下.

为了模拟真实世界的偏好传播情况, 本文采用自适应随机游走生成器将两节点间的转移概率与节点的重要性联系起来, 以保留节点的重要性特征, 为每个用户制定个性化的游走策略. 首先, 规范化每个节点的中心度:

$$c = \text{Softmax}\left(\frac{\text{Centrality}(u_i)}{m}\right) \quad (4)$$

此时, $\text{Centrality}(u_i)$ 代表当前用户节点 u_i 的中心度, m 代表全体用户数. 其次, 受 Node2vec^[29] 启发, 本文引入进出参数 q 用来控制随机游走倾向, 当 $q > 1$ 时算法更倾向于广度优先搜索 (BFS), 当 $q < 1$ 时更倾向于深度优先搜索 (DFS). 通过以下公式针对每个用户更新用来控制随机游走倾向的参数 q :

$$q = \begin{cases} e^c, & c \geq g \\ e^{c-1}, & c < g \end{cases} \quad (5)$$

其中, g 是用户节点重要性阈值. 本文选用所有节点中心度的中位数作为节点重要性阈值. 通过更新 q 值, 可以自适应地根据规范化后的节点中心度为用户定制游走策略, 让生成的节点序列更加符合真实社交网络中节点特征的传播过程. 对于大规模图数据而言, 为了降低计算开销, 常用的方式是图采样以及图划分的方法. 本文的图采样方法受到 PinSage^[11] 的启发, 在自适应随机游走算法生成的节点序列中采样高频节点生成子图, 重构社交网络. 值得注意的是, 为了保持计算模式固定且有效, 本文根据节点频率为每个用户统一采样一组固定大小为 z 的邻域, 即采样在序列集中出现频率最大的前 z 个节点进行图重构. 此方法在一定程度上保持原始网络特征以及不同用户之间的信息平衡.

本文提出的自适应随机游走策略具体步骤见算法 2.

算法 2. 自适应随机游走算法.

输入: 图 $G = (U, E)$, 游走步数 r , 游走步长 l , 返回概率 p ;

输出: 节点序列集 D^U .

1. 初始化节点序列集 D^U 为空集.
2. 根据公式 (4) 计算节点重要程度 c .
3. 根据公式 (5) 计算节点进出参数 q .
4. 将每个节点进出参数 q 添加到节点进出参数集合 Q 中.
5. 重复步骤 2-4 直至遍历用户集中的每一个用户.
6. 预处理计算转移概率 $\pi = \text{PreprocessModifiedWeights}(G, p, q)$.
7. 将这个转移概率加到图 G 中, 形成 $G' = (U, E, \pi)$.
8. 执行随机游走, 生成节点序列.
9. 将节点序列 D_{u_i} 添加到节点序列集合 D^U 中.
10. 重复步骤 8 和 9, 进行迭代更新, 直到达到设置的最大迭代次数 r 为止.
11. 返回节点序列集 D^U .

自适应随机游走算法保留了 Node2vec 算法在 DeepWalk 算法^[30]上进行的改进, 通过引入规范化后的节点中心度控制随机游走的倾向. 本文提出的自适应随机游走算法考虑到不同用户对高阶隐式关系的差异性, 依赖更符合实际情况的中心度计算得到进出参数 q , 代替了传统 Node2vec 算法中简单的固定 q 值, 从而得到更符合实际情况的节点序列.

3.3 用户表示学习

针对重构后的社交网络, 本文采用表示学习方法来建模显式和高阶隐式关系, 以获取更准确的用户表示, 并进一步缓解数据稀疏问题. 给定用户-项目交互矩阵 Y 和用户-用户邻接矩阵 R , 本模块的目标是获得每个用户 u_i 的低维嵌入表示, 希望学习到的嵌入能较好地保留社交影响传播的特征. 本文从两个角度重构社交网络: 显式关系建模和高阶隐式关系建模. 值得注意的是, 在此过程中, 本文不仅考虑了同质性理论^[31], 还同时考虑了社交结构和个人偏好对建模的影响.

(1) 显式关系建模. 根据社交影响传播现象, 用户的偏好会间接受到社交网络中显式关系的影响. 因此, 本文首先对社交网络中的显式关系进行建模. 第 1 步, 模拟两个相关联用户之间的相互作用; 第 2 步, 从用户不同的显式关系中聚合嵌入.

每一个用户 u_i 都对应于一个用户嵌入 \vec{u}_i . 给定一个用户对 (u_i, u_j) , 它的传播嵌入 \vec{P}_{ij} 将用户 u_i 的一阶邻居 u_j 的嵌入表示聚合到 u_i 的用户表示中. 具体公式如下:

$$\vec{P}_{ij} = \frac{1}{\sqrt{|N_{u_i}| |N_{u_j}|}} \vec{u}_j W \quad (6)$$

其中, W 为可训练的权重矩阵. $|N_{u_i}|$ 和 $|N_{u_j}|$ 分别代表用户 u_i 和 u_j 的一阶邻居数量, $\frac{1}{\sqrt{|N_{u_i}| |N_{u_j}|}}$ 为每一次传播的衰减系数. 此外, 以往的研究表明不同社交结构和个人偏好的用户对邻居可能会具有不同大小的社交影响强度. 因此, AHIRM 会为每一个用户对 (u_i, u_j) 分配特定的权重 w_{ij} , 以在嵌入聚合阶段获得更加准确的用户表示. 确切地说, 本文使用在不可靠关系过滤及潜在可靠关系生成阶段得到的整体相似度 $\text{Sim}(u_i, u_j)$ 作为用户 u_i 和 u_j 之间的权重 w_{ij} . 最后, 通过传播的第 1 阶段, 根据以下公式, 可以得到聚合用户 u_i 所有显式关系的传播嵌入:

$$\vec{u}_i^{(1)} = \tanh \left(\sum_{u_j \in N_{u_i} \cup \{u_i\}} w_{ij} \vec{P}_{ij} \right) \quad (7)$$

值得注意的是, 在聚合阶段 AHIRM 模型不仅考虑到显式关系而且考虑到用户的自连接. 本文选取双曲函数

$\tanh()$ 作为非线性激活函数.

(2) 高阶隐式关系建模. 通过对显式社交关系的建模, 用户 u_i 的第 1 个 GCN 层的输出嵌入聚合用户 u_i 的一阶邻居的信息. 在本文中, 通过叠加多个这样的 GCN, 模型可以聚合 u_i 的高阶隐式关系的传播嵌入. 因此, 每个用户 u_i 的嵌入递归建模为:

$$\vec{u}_i^{(k)} = \tanh \left(\sum_{u_j \in N_{u_i} \cup \{u_i\}} w_{ij} \vec{P}_{ij}^{(k)} \right) \quad (8)$$

对应的传播嵌入 $\vec{P}_{ij}^{(k)}$ 定义为:

$$\vec{P}_{ij}^{(k)} = \frac{1}{\sqrt{|N_{u_i}| |N_{u_j}|}} \vec{u}_j^{(k-1)} W^{(k)} \quad (9)$$

其中, $\vec{u}_j^{(k-1)}$ 和 $W^{(k)}$ 分别为用户 u_j 第 $k-1$ 层的输出嵌入和可训练的权重矩阵.

在得到用户 u_i 的最终嵌入后, 本文基于用户的最终嵌入 $\vec{u}_i^{(k)}$ 以及对应项目的嵌入 \vec{i}_j^T 计算用户 u_i 对某项目 i_j 的评分 $\hat{y}_{u_i i_j}$, 计算公式如下:

$$\hat{y}_{u_i i_j} = \vec{u}_i^{(k)} \cdot \vec{i}_j^T \quad (10)$$

为了让计算更加的高效, 本文在训练过程中运用了负采样策略, 损失函数 L 如下所示:

$$L = \sum_{u_i \in U} \left(\sum_{i_j: y_{u_i i_j} = 1} \mathcal{J}(y_{u_i i_j}, \hat{y}_{u_i i_j}) - \sum_{j=1}^{F_u} \mathbb{E}_{i_j \sim p(i_j)} \mathcal{J}(y_{u_i i_j}, \hat{y}_{u_i i_j}) \right) + \lambda \|\mathcal{F}\|_2 \quad (11)$$

其中, \mathcal{J} 表示交叉熵损失, p 是负采样的分布, F_u 为针对用户 u 的负采样样本数量. 最终本文计算出损失并使用 Adam 优化器^[32]对 AHIRM 模型进行优化. AHIRM 的伪代码如算法 3 所示.

算法 3. AHIRM 算法.

输入: 用户集 U , 项目集 I , 用户-项目交互矩阵 Y , 社交网络 G , 阶数 k ;

输出: 预测函数 $\mathcal{F}(u, i|\theta, Y, G)$.

1. 初始化用户和项目嵌入向量 \vec{u}_i 和 \vec{i}_j .
 2. 用 \vec{u}_i 初始化用户 u_i 的 0 阶嵌入.
 3. 计算用户 u_i 的 $k-1$ 阶传播嵌入向量 $\vec{P}_{ij}^{(k-1)}$.
 4. 更新用户 u_i 的 k 阶传播嵌入表示.
 5. 重复步骤 3 和 4 直至遍历用户 u_i 的 k 阶邻居集中的用户.
 6. 更新节点的最终嵌入表示 $\vec{u}_i^{(k)}$.
 7. 重复步骤 2-6 直至遍历用户-项目交互矩阵 Y 中的所有用户-项目对 (u_i, u_j) .
 8. 计算最终预测概率 $\hat{y}_{u_i i_j} = f(\vec{u}_i^{(k)}, \vec{i}_j)$.
 9. 用梯度下降法更新参数.
 10. 直至算法收敛, 返回预测函数 $\mathcal{F}(u, i|\theta, Y, G)$.
-

3.4 算法分析

定理 1. AHIRM 算法的时间复杂度为 $O(m^2 + m^2 r l + k(|R| + |S|)d^2 + |Y|d)$, 其中, m, r, l, k 分别表示用户个数、节点游走次数、游走步长和阶数. $|R|, |S|, d$ 分别表示用户-用户邻接矩阵 R 、用户的自连接矩阵 S 中非零元素的个数和嵌入维度.

证明: 由于 AHIRM 模型的 3 个关键过程为不可靠关系过滤及潜在可靠关系生成、图重构和嵌入传播聚合. 不可靠关系过滤及潜在可靠关系生成包括相似度计算, 过滤和潜在可靠关系的发现 3 个步骤, 时间复杂度为 $O(m^2)$, 其中 m 表示用户个数. 在图重构阶段, 自适应随机游走时间复杂度为 $O(m^2 r l)$, 其中 r 表示节点游走次数, l 为游走

步长. 在影响力传播的过程中, 需要对每一层进行矩阵乘法运算, 此时的时间复杂度为 $O(k(|R|+|S|)d^2)$. $|R|$ 和 $|S|$ 表示用户-用户邻接矩阵 R 和表示用户的自连接矩阵 S 中非零元素的个数, d 为嵌入维度, k 为阶数. 值得注意的是, 偏好预测主要通过内积实现. 内积的计算复杂度为 $O(|Y|d)$. 因此, 该算法的复杂度为 $O(m^2+m^2rl+k(|R|+|S|)d^2+|Y|d)$. 证毕.

定理 2. AHIRM 算法的空间复杂度为 $O(m^2+|E|+a^2m)$. 其中, $m, |E|, a$ 分别代表用户的数量, 其直接邻居数以及用户的度数.

证明: 在不可靠关系过滤及潜在可靠关系生成阶段, 由于有 m 个用户, 本文以矩阵存储两两之间的相似度, 空间复杂度为 $O(m^2)$; 在自适应随机游走阶段和最终的用户表示学习阶段, 需要存储用户的直接(一阶)邻居以及二阶邻居, 此阶段的空间复杂度为 $O(|E|+a^2m)$. 因此, 该算法的空间复杂度为 $O(m^2+|E|+a^2m)$. 证毕.

定理 3. 本文提出的参数的启发式设置策略是合理的.

证明: AHIRM 模型引入的节点重要程度 c 可定义为 $c = \text{Softmax}\left(\frac{\text{Centrality}(u_i)}{m}\right)$. 因为节点度数 $\text{Centrality}(u_i) > 0$, 可以推出 $c > 0$, 则 e^c 及 e^{c-1} 有一定的启发式意义. 由于 $0 < c < 1$, 因此, $e^c > 1$, $e^{c-1} < 1$. 根据 Node2vec 的结论, 当 $q > 1$ 时算法更倾向于广度优先搜索 (BFS), 当 $q < 1$ 时更倾向于深度优先搜索 (DFS). 因此我们可以将 e^c 及 e^{c-1} 与进出参数 q 联系起来, 即可基于具有启发意义的 e^c 和 e^{c-1} 设置进出参数 q 的值. 证毕.

3.5 实例分析

为了更加直观地理解本文提出的 AHIRM 算法, 本文采用了由 6 个节点和 8 条边组成的小型社交网络以及 12 个节点组成的简单用户-项目交互网络, 来说明本文算法的计算过程. 具体如图 4 所示.

计算流程: 设置过滤阈值 $s=0.15$, 偏好阈值 $t=0.80$, 相似度权重 w_1, w_2 均为 0.5. 不可靠关系过滤及潜在可靠关系生成阶段主要包括不可靠关系的过滤以及潜在可靠关系的发现. 针对不可靠关系的过滤, 首先根据输入的社交网络和用户-项目交互网络计算任意两关联用户之间的整体相似度. 其中整体相似度根据公式 (3) 由结构相似度和偏好相似度共同得到. 例如, 以用户 u_1 作为初始节点, 首先根据输入的社交网络计算 u_1 与其邻居节点 u_2 的结构相似度 $\text{Sim_structure}(u_1, u_2) = \frac{|N_{u_1} \cap N_{u_2}|}{|N_{u_1} \cup N_{u_2}|} = \frac{|{\{u_2, u_3, u_5\} \cap \{u_1, u_3, u_4\}}|}{|{\{u_2, u_3, u_5\} \cup \{u_1, u_3, u_4\}}|} = \frac{1}{5} = 0.2$; 然后根据用户-项目交互网络的信息计算 u_1 与其邻居节点 u_2 之间的偏好相似度. 具体而言, 首先根据交互信息得到用户 u_1 的评分向量 $\vec{r}_1 = (1, 0, 1, 0, 1, 1)$ 以及用户 u_2 的评分向量 $\vec{r}_2 = (1, 0, 1, 0, 0, 0)$. 接下来计算两用户间的偏好相似度 $\text{Sim_preference}(u_1, u_2) = \frac{\vec{r}_1 \cdot \vec{r}_2}{\|\vec{r}_1\| \cdot \|\vec{r}_2\|} = \frac{(1, 0, 1, 0, 1, 1) \cdot (1, 0, 1, 0, 0, 0)}{\sqrt{1^2+1^2+1^2+1^2} \times \sqrt{1^2+1^2}} = 0.70$. 最后根据公式 (3) 得出用户 u_1 和 u_2 的整体相似度 $\text{Sim}(u_1, u_2) = \sigma(w_1 \text{Sim_structure}(u_1, u_2) + w_2 \text{Sim_preference}(u_1, u_2)) = \sigma(0.5 \times 0.2 + 0.5 \times 0.7) = \frac{1}{1+e^{-0.45}} = 0.61$. 然后将整体相似度与过滤阈值 $s = 0.15$ 进行比较, 由于整体相似度 0.61 大于过滤阈值, 因此不进行过滤. 同理, 计算用户 u_1 与其他邻居节点的整体相似度. 结果如图 4 中步骤 2 所示. 由于 (u_4, u_5) 之间的整体相似度低于阈值, 将其从用户邻居集合中移除, 完成不可靠关系的过滤. 潜在可靠关系的生成, 主要针对潜在用户候选对, 不考虑其结构相似性, 仅考虑偏好相似度, 将其偏好相似度与偏好阈值 $t = 0.80$ 进行比较, 具体结果如图 4 中步骤 2 所示, 由于潜在用户候选对中 (u_1, u_4) 和 (u_3, u_6) 的偏好相似度均大于偏好阈值 t , 因此 u_1 和 u_4 及 u_3 和 u_6 具有潜在可靠关系, 将其加入社交网络. 其余潜在用户候选对的偏好相似度均小于偏好阈值 t , 因此, 将它们从潜在用户候选对中删除, 完成潜在可靠关系的生成. 经过不可靠关系过滤及潜在可靠关系生成阶段, 得到了新的社交网络. 针对此网络, 根据规范化后的节点中心度为每个用户计算出控制随机游走倾向的进出参数 q , 并执行自适应随机游走算法, 得到可以保存节点特征的节点序列. q 值计算及路径结果如图 4 步骤 3 图重构所示. 接下来, 计算各个节点在序列集中的出现频率, 采样高频节点生成子图, 进而重构社交网络. 最后, 用户表示学习阶段运用 GCN 对高阶隐式关系进行建模. 用户 u_i 在第 1 次迭代时, 通过传播其邻域节点的初始嵌入, 结合节点自身嵌入, 得到节点的一阶表示, 经过多次迭代得到节点的最终嵌入表示. 结合节点最终嵌入表示和项目嵌入, 得到最终预测概率 $\hat{y}_{u_i j}$.

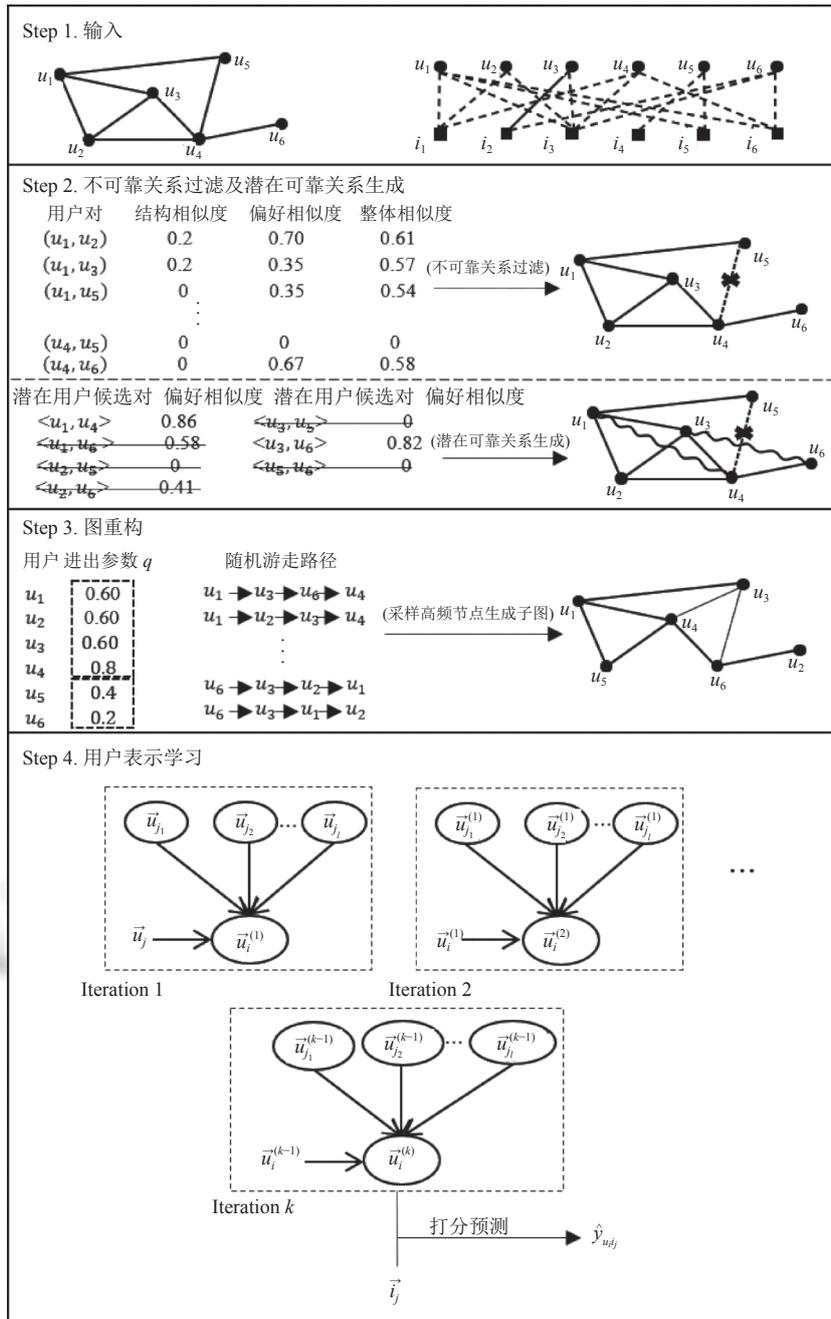


图 4 AHIRM 运行示例

4 实验

为了评价本文提出的 AHIRM 模型的性能, 在两个真实数据集上进行了多组实验. 通过实验分析, 本文旨在回答以下研究问题.

RQ1: 与最先进的社交推荐方法相比, AHIRM 模型的表现如何?

RQ2: 不可靠关系过滤及潜在可靠关系生成能否提高现有社交推荐方法的有效性?

RQ3: 本文提出的自适应随机游走生成器如何影响 AHIRM 模型的性能?

RQ4: 关键参数如何影响 AHIRM 模型的性能?

接下来, 首先介绍实验设置, 然后依次回答上述研究问题.

4.1 实验设置

(1) 数据集. 本文在 LastFM (<http://files.grouplens.org/datasets/hetrec2011/>)、Douban (http://smiles.xjtu.edu.cn/Download/download_Douban.html) 和 Gowalla (<https://www.dropbox.com/sh/qy3s8rs66nirh19/AAClmTnFO-rR-4ecEYO-jU4ba?dl=0>) 数据集上进行实验. LastFM 数据集包含 <https://www.last.fm> 社交音乐网站内用户的社交网络数据和用户收听偏好信息. Douban 数据集被广泛用于电影推荐领域, 作为基准数据集, 记录了人们根据自己的喜好对电影的评分, 以及用户间的社交关系. Gowalla 数据集包含用户的好友及签到信息. 由于 Douban 数据集是显式反馈, 本文将其转换为隐式反馈再进行计算. 在 Douban 数据集上将评分大于 3 分的视为正例, 负例通过负采样获取. 关于数据集的描述性统计信息如表 1 所示.

表 1 数据集的描述

数据集	# Users	# Items	# Ratings	# Links	Density (%)
LastFM	1 892	17 632	92 834	25 434	0.278 3
Douban	2 848	39 586	894 887	35 770	0.793 8
Gowalla	18 737	32 510	1 278 274	86 985	0.209 8

(2) 评价指标. 对于每个数据集, 随机抽取 60% 的实例作为训练集, 20% 的实例作为验证集, 其余 20% 的实例作为测试集. 为了评估模型的推荐性能, 采用了学术界广泛使用的 Precision@K, Recall@K, F1@K 以及 NDCG@K 评价评分预测任务的性能.

(3) 对比方法. 为了验证本文提出的 AHIRM 算法在推荐结果上的有效性, 将 AHIRM 算法与以下对比方法进行比较.

- BPR^[33]: 一个贝叶斯个性化排序模型. 通过最小化成对排序损失优化了矩阵分解. 该模型仅适用于用户与项目之间存在隐性反馈的情境.
- SBPR^[34]: 此模型在贝叶斯个性化排序中结合用户的社交反馈信息进行排序.
- IF-BPR^[35]: 利用基于元路径的嵌入表示学习来识别异构信息网络中的隐式关系.
- DiffNet++^[20]: 一种标准的跨领域社交推荐方法, 系统地聚合信息域的用户-项目交互信息和社交域的关系信息.
- RSGAN^[36]: 一种基于生成对抗网络的端到端社交推荐框架, 通过将对抗性学习引入到社交推荐中, 模型可以自适应的在观察到和未观察到的社交网络中识别可靠的朋友, 预测当前用户的偏好.
- LightGCN^[37]: 一种新的协同过滤模型, 旨在简化 GCN 的设计, 使其更简洁, 更适合推荐. 该模型通过在用户-项目交互网络上进行线性传播来学习用户和项目的嵌入.
- ESRF^[38]: 一个基于图卷积网络 (GCN) 的深度对抗框架, 将对抗性训练和 GCNs 结合起来解决社交推荐系统的挑战.

(4) 参数设置. 本文提出的 AHIRM 算法以及所有对比方法均在 PyTorch 上实现. 实验涉及参数均通过网格实验选取最佳性能参数. 本文将所有模型的潜在因子向量维度均设置为 50, 正则化系数 λ 设置为 0.005, 批处理大小为 512, 以进行公平的比较. 对于所有基于 GCN 的模型, 其层数根据在各个数据集上的最佳性能选取. 最终, LastFM、Douban 和 Gowalla 数据集上的层数分别为 4、3 和 4. 在 ESRF 模型中, 可选邻居数 k 在 LastFM、Douban 和 Gowalla 数据集上的取值分别为 40、30 和 40, 控制对抗训练规模的参数 β 分别取值为 0.3、0.6 和 0.5. 由于 AHIRM 算法中采用自适应随机游走算法进行图重构具有随机性, 所以每个实验重复进行 20 次, 结果给出 20 次实验的平均值.

4.2 性能比较 (RQ1)

表 2 展示了对比方法和 AHIRM 的性能, 实验结果及分析如下.

(1) 社交推荐模型 SBPR、IF-BPR、RSGAN 和 LightGCN 的推荐性能均显著优于一般推荐模型 BPR. 这表明在推荐系统中引入社交关系有助于提高推荐系统的性能.

(2) 社交推荐模型中 IF-BPR 和 RSGAN 推荐性能优于 SBPR, 表明直接使用显示关系是次优的. 可能的原因在于显示关系存在噪声和稀疏性现象, 降低了推荐模型的性能. 因此, 有必要滤除噪声数据和探索隐式关系.

(3) 一般推荐模型中 LightGCN 性能优于 BPR 方法. 在直接运用显示关系进行推荐的模型中, DiffNet++ 性能优于 SBPR. 这可能是由于 LightGCN 和 DiffNet++ 均为基于 GCN 的推荐模型, 表明在推荐模型中运用 GCN 的有效性.

(4) 在 LastFM 和 Douban 数据集上, ESRF 和 AHIRM 性能优于其他对比方法, 表明在推荐过程中建模高阶隐式关系的有效性. 值得注意的是, 虽然 IF-BPR 也对高阶隐式关系进行建模, 但并没有呈现较好的性能. 这可能是因为它没有自适应根据用户对的情况为关系分配不同的权重, 降低了模型的性能.

(5) 为了验证 AHIRM 能部分缓解数据稀疏问题, 我们在更加稀疏的 Gowalla 数据集上进行实验. 由于 Gowalla 数据集的稀疏性, 相较于 LastFM 和 Douban 数据集, 在 Gowalla 数据集上模型整体性能有所下降. 值得注意的是, AHIRM 在此数据集上实现了 $F1@10$ 以及 $NDCG@10$ 两个指标上的最大性能提升, 分别为 10.13% 和 10.71%. 这表明本文提出的模型能有效缓解数据稀疏问题. 此外, 除了本文提出的 AHIRM 模型, IF-BPR、RSGAN 以及 ESRF 模型性能优于其他模型, 表明过滤不可靠关系及探索隐式关系的必要性.

(6) AHIRM 在 3 个数据集上的推荐性能优于所有对比方法. 在 Douban 数据集上, 本算法的 $F1@10$ 和 $NDCG@10$ 值相较于目前最优的模型 (ESRF) 分别提升了 9.76% 和 10.27%, 表明本文提出的高阶隐式关系建模方法的有效性.

表 2 LastFM, Douban 和 Gowalla 数据集上 Top-10 推荐的性能比较

Dataset	Metric	BPR	SBPR	IF-BPR	DiffNet++	RSGAN	LightGCN	ESRF	AHIRM	Improv.
LastFM	Precision@10 (%)	7.59	8.28	9.10	10.08	9.48	10.14	10.72	11.58**	8.02
	Recall@10 (%)	11.31	12.76	13.63	14.80	14.13	14.96	16.07	17.34**	7.84
	$F1@10$ (%)	9.08	10.04	10.91	11.99	11.34	12.09	12.87	13.89**	7.93
	$NDCG@10$	0.1076	0.1219	0.1304	0.1432	0.1350	0.1437	0.1534	0.1638**	6.78%
Douban	Precision@10 (%)	14.89	15.50	16.43	17.38	17.26	17.27	18.23	19.46**	6.75
	Recall@10 (%)	5.02	5.42	5.54	6.14	6.03	6.10	6.55	7.27**	10.99
	$F1@10$ (%)	7.51	8.03	8.29	9.07	8.94	9.10	9.63	10.57**	9.76
	$NDCG@10$	0.1586	0.1725	0.1855	0.1964	0.1947	0.1949	0.2103	0.2319**	10.27%
Gowalla	Precision@10 (%)	4.41	4.02	5.21	5.10	5.38	5.19	5.84	6.41**	9.76
	Recall@10 (%)	5.45	4.81	6.29	6.01	6.27	6.10	6.89	7.62**	10.59
	$F1@10$ (%)	4.86	4.38	5.70	5.52	5.79	5.61	6.32	6.96**	10.13
	$NDCG@10$	0.0627	0.0558	0.0734	0.0710	0.0743	0.0720	0.0803	0.0889**	10.71%

注: ** indicates that the improvements are statistically significant for $p < 0.01$ judged by paired t-test

4.3 不可靠关系过滤及潜在可靠关系生成的影响 (RQ2)

不可靠关系过滤及潜在可靠关系生成旨在缓解不可靠关系带来的不利影响, 并部分缓解数据稀疏性问题. 为了验证其有效性, 本节通过将 AHIRM 和其 3 个模型变体: 仅使用结构相似度的模型 (AHIRM_{OS}), 仅使用偏好相似度的模型 (AHIRM_{OP}) 以及移除整个不可靠关系过滤及潜在可靠关系生成模块的模型 (AHIRM_{NS}) 进行比较. 结果如表 3 所示: (1) 通过将 AHIRM 模型和 AHIRM_{NS} 模型比较, 可知: AHIRM 模型性能显著优于 AHIRM_{NS}, 表明引入不可靠关系过滤及潜在可靠关系生成模块的重要性; (2) 为了进一步分析结构相似度与偏好相似度对模型性能的影响, 我们将 AHIRM_{OS} 和 AHIRM_{OP} 两个变体方法与另一变体 AHIRM_{NS} 进行比较: AHIRM_{OS} 和 AHIRM_{OP} 的性能与 AHIRM_{NS} 相比均有一定的提升, 表明引入结构相似度和偏好相似度对提高推荐系统性能都具有重要作用; (3) 在 3 个数据集上, AHIRM_{OP} 的性能均高于 AHIRM_{OS}, 表明偏好相似度相对于结构相似度而言, 对推荐系统性能具有更大的影响. 这可能是由于社交关系建立的随机性, 部分不可靠的社交关系会带来负面影响. 总的来说, 在 3 个数据集上 AHIRM 模型都获得了最佳性能, 表明综合考虑社交结构和个人偏好进行不可靠关系过滤及潜在可靠关系生成的必要性和合理性.

表 3 不可靠关系过滤及潜在可靠关系生成的作用

Dataset	Metric	AHIRM _{OS}	AHIRM _{OP}	AHIRM _{NS}	AHIRM
LastFM	Precision@10 (%)	7.32	8.05	6.95	11.58**
	Recall@10 (%)	11.97	12.57	11.94	17.34**
	F1@10 (%)	9.08	9.81	8.58	13.89**
	NDCG@10	0.1215	0.1291	0.1221	0.1638**
Douban	Precision@10 (%)	12.71	15.57	12.01	19.46**
	Recall@10 (%)	4.73	5.38	4.58	7.27**
	F1@10 (%)	6.89	8.00	6.48	10.57**
	NDCG@10	0.1657	0.1902	0.1348	0.2319**

注: ** indicates that the improvements are statistically significant for $p < 0.01$ judged by paired t-test

4.4 自适应随机游走算法的影响 (RQ3)

为了验证本文提出的自适应随机游走算法的有效性, 本节将 AHIRM 与采用不同随机游走算法的变体进行比较. 在保持其他条件完全相同的情况下, 使用自适应随机游走算法的替代算法来替换本文提出的自适应随机游走生成器, 替换后的模型分别记为 AHIRM_{Deep} 和 AHIRM_{Bine}. 结果如表 4 所示, AHIRM_{Bine} 和 AHIRM 模型的性能均优于 AHIRM_{Deep} 模型, 这表明自适应概念的引入能更好地刻画高阶隐式关系对不同用户具有不同重要性这一特殊现象, 提高了推荐系统的性能. 在 LastFM 数据集上, 相较于 AHIRM_{Bine} 算法, AHIRM 模型分别在 F1@10 和 NDCG@10 两个指标上分别提升了 6.03% 和 4.20%. 这表明本文提出的根据用户节点重要性为每个用户制定不同的随机游走策略是有效的.

表 4 不同随机游走策略的 AHIRM

Dataset	Metric	AHIRM _{Deep}	AHIRM _{Bine}	AHIRM	Improv.
LastFM	Precision@10 (%)	8.93	11.26	11.58**	2.84
	Recall@10 (%)	13.99	15.85	17.34**	9.40
	F1@10 (%)	10.79	13.10	13.89**	6.03
	NDCG@10	0.1411	0.1572	0.1638**	4.20%
Douban	Precision@10 (%)	14.08	15.84	19.46**	22.85
	Recall@10 (%)	5.52	6.19	7.27**	17.45
	F1@10 (%)	7.86	8.80	10.57**	20.11
	NDCG@10	0.1790	0.2008	0.2319**	15.49%

注: ** indicates that the improvements are statistically significant for $p < 0.01$ judged by paired t-test

4.5 关键参数的影响 (RQ4)

在本节中, 分析了 AHIRM 中 4 个关键参数对模型性能的影响, 包括用来控制结构相似度和偏好相似度的权重 w_1 和 w_2 , 过滤阈值 s 以及图重构模块中用于控制采样邻居个数的参数 z . 由于本文权重 w_1 和 w_2 的和为 1, 因此我们仅对 3 个参数 w_1 , s 及 z 进行网格测试.

本文先固定 w_1 为 0.5, 过滤阈值 s 为 0.05, 改变两个数据集的 z . 通过改变采样邻居的数量来研究采样的性能. 如图 5 所示, AHIRM 在 LastFM 数据集与 Douban 数据集下分别在 $z=4$ 以及 $z=7$ 时达到最佳性能. 此外, 随着 z 的增加, 性能总体呈现先上升后下降趋势. 这表明, 更大的 z 可以捕获更多的用户、项目信息, 从而进行更有效的建模. 值得注意的是, 当 z 超过最佳性能时, 性能有所下降, 这可能因为当数据稀疏时, 增加数据信息会有所帮助, 但过度的增加也会引入噪声数据, 从而降低模型性能.

然后, 分别将两个数据集的 z 定为 4 和 7, 仍然固定 w_1 为 0.5, 改变过滤阈值 s . 从图 6 可以看出, 过滤阈值 s 对推荐性能存在影响. 随着过滤阈值 s 的变化, 性能先提升然后下降. 这可能是由于当过滤阈值 s 过小时, 无法滤除噪声节点导致性能一般, 而过滤阈值 s 过大又可能会过滤掉部分可用节点, 降低模型性能. LastFM 数据集和 Douban 数据集分别在 $s=0.04$ 和 $s=0.06$ 时达到最佳性能.

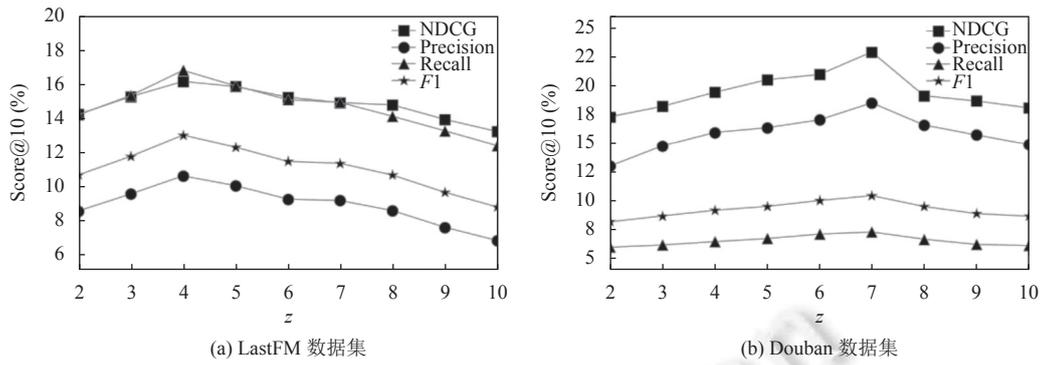


图 5 采样邻居数量 z 对算法性能的影响

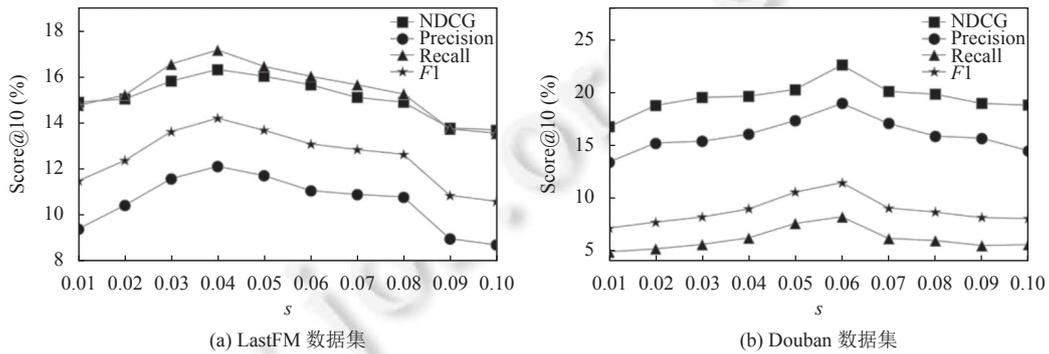


图 6 过滤阈值 s 对算法性能的影响

最后, 在 LastFM 数据集的实验和 Douban 数据集的实验分别把 z 固定为 4 和 7, 然后把 s 固定到 0.04 和 0.06 并改变 w_1 . 如图 7 所示, 可以看出 w_1 对维持社交结构和个人偏好之间的平衡起着至关重要的作用. 在 LastFM 数据集中, w_1 的最优值是 0.5, 表明在此数据集上结构相似性与偏好相似性对整体相似度的影响差别不大. 值得注意的是, Douban 数据集的实验中 w_1 的最优值是 0.3. 这表明在 Douban 数据集上, 偏好相似性更能有效测量用户之间的整体相似性. 这可能是由于 Douban 数据集的项目以及评分数量远远大于 LastFM 数据集, 因此其偏好相似性更能准确反映用户之间的关系. 以上实验结果表明: 使用权重 w_1 动态决定结构相似性和偏好相似性对整体相似度的影响程度是必要且合理的.

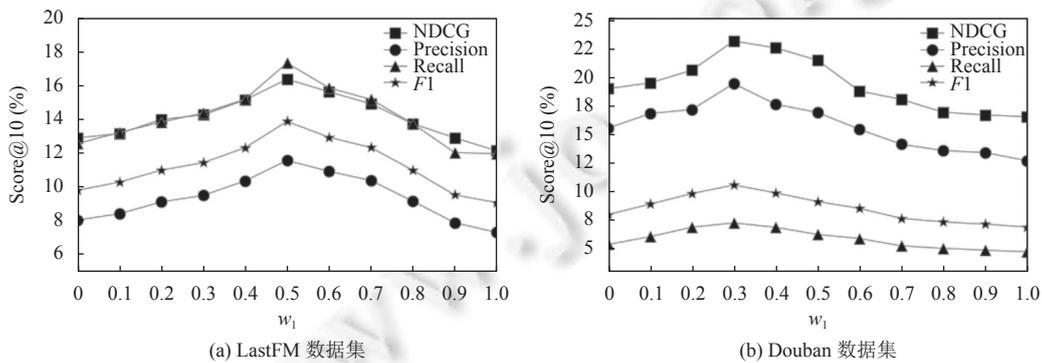


图 7 权重 w_1 对算法性能的影响

4.6 案例分析

为了详细分析模型如何建模高阶隐式关系和捕获不同阶数的邻居, 进行案例分析. 本文随机选取 6 个用

户 $U = \{u_1, u_2, \dots, u_6\}$ 以及相应的项目 $I = \{i_1, i_2, \dots, i_6\}$, 表 5 是这 6 个用户的社交关系数据, 其中 1 表示两用户间存在社交关系, 0 代表不存在社交关系. 表 6 是用户与项目之间的交互信息, 其中值为 1 代表用户观看过此项目, 缺失值代表用户没看过该项目. 其中 u_1, u_6 是目标用户. 首先基于该数据集, 根据用户的历史信息, 利用公式 (1)–公式 (3) 分别计算用户间的结构相似度, 偏好相似度和整体相似度, 并以此为基础进行不可靠关系的过滤和潜在可靠关系的生成, 具体步骤可见第 3.5 节实例分析, 在此过程中我们可以得到新的社交网络如表 7 所示.

表 5 初始社交网络数据

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	0	1	1	0	1	0
u_2	1	0	1	1	0	0
u_3	1	1	0	1	0	0
u_4	0	1	1	0	1	1
u_5	1	0	0	1	0	0
u_6	0	0	0	1	0	0

表 6 用户对项目的交互数据

用户	i_1	i_2	i_3	i_4	i_5	i_6
u_1	1	—	1	—	1	1
u_2	1	—	1	—	—	—
u_3	—	1	1	—	—	—
u_4	1	—	1	—	—	1
u_5	—	—	—	1	1	—
u_6	—	1	1	—	—	1

表 7 更新后的社交网络数据

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	0	1	1	0	1	0
u_2	1	0	1	1	0	0
u_3	1	1	0	0	0	1
u_4	0	1	0	0	0	0
u_5	1	0	0	0	0	0
u_6	0	0	1	0	0	0

基于上述得到的社交网络, 执行本文提出的自适应随机游走, 首先依据公式 (4) 和公式 (5) 计算出不同用户规范化后的节点中心度 c 和控制随机游走倾向的进出参数 q . 执行随机游走, 目标用户 u_1, u_6 生成的路径如表 8 所示. 其次我们分别采样生成路径中的高频节点生成目标节点的子图, 进而重构社交网络. 例如, 针对目标用户 u_1 , 我们得到的采样节点为 u_3 和 u_4 , 它们分别是用户 u_1 的一阶和二阶邻居. 同样地, 针对目标用户 u_6 , 我们得到的采样节点为 u_1 和 u_2 , 它们分别是用户 u_6 的三阶和二阶邻居. 重构后的社交网络如表 9 所示. 针对目标用户, 重新构建的社交关系聚集了各阶邻居信息. 然后, 依据本文的用户表示学习模块的公式 (6) 和公式 (7) 聚集邻居节点信息生成用户嵌入. 最后根据公式 (10) 预测表 6 中目标用户 u_1, u_6 的对未交互过的项目的偏好, 即预测表 6 中的空缺值, 得到表 10.

表 8 自适应随机游走相关信息

目标用户	游走路径	进出参数 q	采样节点	阶数
u_1	$u_1 \rightarrow u_4 \rightarrow u_2 \rightarrow u_3$	0.37	u_3, u_4	1, 2
	$u_1 \rightarrow u_4 \rightarrow u_6 \rightarrow u_3$			
	$u_1 \rightarrow u_4 \rightarrow u_3 \rightarrow u_6$			
	$u_1 \rightarrow u_3 \rightarrow u_4 \rightarrow u_6$			
	$u_1 \rightarrow u_3 \rightarrow u_2 \rightarrow u_1$			
u_6	$u_6 \rightarrow u_3 \rightarrow u_1 \rightarrow u_2$	2.72	u_1, u_2	3, 2
	$u_6 \rightarrow u_1 \rightarrow u_2 \rightarrow u_4$			
	$u_6 \rightarrow u_1 \rightarrow u_2 \rightarrow u_3$			
	$u_6 \rightarrow u_2 \rightarrow u_1 \rightarrow u_3$			
	$u_4 \rightarrow u_3 \rightarrow u_2 \rightarrow u_1$			

表 9 重构后的社交网络数据

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	0	0	1	1	1	0
u_2	0	0	0	0	0	1
u_3	1	0	0	1	0	1
u_4	1	0	1	0	1	1
u_5	1	0	0	1	0	0
u_6	0	1	1	1	0	0

表 10 用户对项目的偏好

用户	i_1	i_2	i_3	i_4	i_5	i_6
u_1	1	0.81	1	0.73	1	1
u_6	1	0.74	1	0.71	0.82	1

最终, 根据表 10 中的预测偏好数据, 按由高到低排列. 最终, 目标用户 u_1, u_6 中除去已知存在交互的项目, 其余评分最高的两个项目分别为 i_2, i_4 和 i_2, i_5 , 若采用 top-2 推荐, 为目标用户 u_1, u_6 分别推荐项目 i_2, i_4 和 i_2, i_5 . 到此整个推荐过程全部完成.

5 结束语

本文研究社交推荐场景下的高阶隐式关系建模问题. 与已有的高阶隐式关系建模方法不同, 本文研究如何自适应地为用户聚合高阶隐式关系, 使生成的用户嵌入更加符合推荐场景. 本文提出的高阶隐式关系建模方法 (AHIRM), 结合社交结构和个人偏好过滤不可靠关系并生成潜在可靠关系; 基于规范化后的节点中心度设计自适应随机游走算法捕获不同阶数的邻居, 构建用户间的高阶隐式关系; 最后, 运用 GCN 模拟社交影响传播过程, 并对高阶隐式关系进行建模. 在社交推荐任务上的大量实验验证了 AHIRM 方法的有效性和合理性.

尽管考虑到高阶隐式关系在不同用户间的差异性, 但是用户的偏好可能会随着时间和其他因素动态变化. 因此, 未来计划研究如何有效地更新动态网络的嵌入. 此外, 尽管本文通过潜在可靠关系生成和高阶隐式关系建模部分缓解了数据稀疏问题. 但是, 就推荐系统而言, 数据稀疏问题依然是极具挑战的, 在未来的工作中, 我们计划引入更多的辅助信息, 如: 用户简档、项目特征和其他属性等, 进行更准确的建模.

References:

- [1] Guo GB, Zhang J, Yorke-Smith N. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: Proc. of the 29th AAAI Conf. on Artificial Intelligence. Austin: AAAI Press, 2015. 123–129.
- [2] Jiang M, Cui P, Wang F, Zhu WW, Yang SQ. Scalable recommendation with social contextual information. IEEE Trans. on Knowledge and Data Engineering, 2014, 26(11): 2789–2802. [doi: 10.1109/TKDE.2014.2300487]
- [3] Wang X, He XN, Nie LQ, Chua TS. Item silk road: Recommending items from information domains to social users. In: Proc. of the 40th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Shinjuku: ACM, 2017. 185–194. [doi: 10.1145/3077136.3080771]
- [4] Liu Y, Chen L, He XN, Peng JY, Zheng ZB, Tang J. Modelling high-order social relations for item recommendation. arXiv:2003.10149, 2020. [doi: 10.1109/TKDE.2020.3039463]
- [5] Qiu JZ, Tang J, Ma H, Dong YX, Wang KS, Tang J. DeepInf: Social influence prediction with deep learning. In: Proc. of the 24th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. London: ACM, 2018. 2110–2119. [doi: 10.1145/3219819.3220077]
- [6] Wang X, Chen WX, Yang YJ, Zhang XW, Feng ZY. Research on knowledge graph partitioning algorithms: A survey. Chinese Journal of Computers, 2021, 44(1): 235–260 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2021.00235]
- [7] Dou JH, Tian B, Zhang Y, Xing CX. A novel embedding model for knowledge graph completion based on multi-task learning. In: Proc. of the 26th Int'l Conf. on Database Systems for Advanced Applications. Taipei: Springer, 2021. 240–255. [doi: 10.1007/978-3-030-73194-6_17]
- [8] Malik S, Rana A, Bansal M. A survey of recommendation systems. Information Resources Management Journal, 2020, 33(4): 4. [doi: 10.4018/IRMJ.2020100104]
- [9] Sun JN, Zhang YX, Guo W, Guo HF, Tang RM, He XQ, Ma C, Coates M. Neighbor interaction aware graph convolution networks for recommendation. In: Proc. of the 43rd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020. 1289–1298. [doi: 10.1145/3397271.3401123]
- [10] Sun JN, Zhang YX, Ma C, Coates M, Guo HF, Tang RM, He XQ. Multi-graph convolution collaborative filtering. In: Proc. of the 19th IEEE Int'l Conf. on Data Mining. Beijing: IEEE, 2019. 1306–1311. [doi: 10.1109/ICDM.2019.00165]
- [11] Ying R, He RN, Chen KF, Eksombatchai P, Hamilton WL, Leskovec J. Graph convolutional neural networks for Web-scale recommender systems. In: Proc. of the 24th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. London: ACM, 2018. 974–983. [doi: 10.1145/3219819.3219890]
- [12] Tan QY, Liu NH, Zhao X, Yang HX, Zhou JR, Hu X. Learning to hash with graph neural networks for recommender systems. In: Proc. of the 2020 Web Conf. Taipei: ACM, 2020. 1988–1998. [doi: 10.1145/3366423.3380266]
- [13] Chen L, Wu L, Hong RC, Zhang K, Wang M. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI Press, 2020. 27–34. [doi: 10.1609/aaai.v34i01.5330]
- [14] Wang X, He XN, Wang M, Feng FL, Chua TS. Neural graph collaborative filtering. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on

- Research and Development in Information Retrieval. Paris: ACM, 2019. 165–174. [doi: [10.1145/3331184.3331267](https://doi.org/10.1145/3331184.3331267)]
- [15] Liao GQ, Lan TM, Huang XM, Chen H, Wan CX, Liu DX, Liu XP. Survey on recommendation systems in event-based social networks. *Ruan Jian Xue Bao/Journal of Software*, 2021, 32(2): 424–444 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6145.htm> [doi: [10.13328/j.cnki.jos.006145](https://doi.org/10.13328/j.cnki.jos.006145)]
- [16] Zhao S, Liu XM, Duan Z, Zhang YP, Tang J. A survey on social ties mining. *Chinese Journal of Computers*, 2017, 40(3): 535–555 (in Chinese with English abstract). [doi: [10.11897/SP.J.1016.2017.00535](https://doi.org/10.11897/SP.J.1016.2017.00535)]
- [17] Tang JL, Hu X, Gao HJ, Liu H. Exploiting local and global social context for recommendation. In: Proc. of the 23rd Int'l Joint Conf. on Artificial Intelligence. Beijing: AAAI Press, 2013. 2712–2718.
- [18] Sun PJ, Wu L, Wang M. Attentive recurrent social recommendation. In: Proc. of the 41st Int'l ACM SIGIR Conf. on Research & Development in Information Retrieval. Ann Arbor: ACM, 2018. 185–194. [doi: [10.1145/3209978.3210023](https://doi.org/10.1145/3209978.3210023)]
- [19] Wu L, Sun PJ, Fu YJ, Hong RC, Wang XT, Wang M. A neural influence diffusion model for social recommendation. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. Paris: ACM, 2019. 235–244. [doi: [10.1145/3331184.3331214](https://doi.org/10.1145/3331184.3331214)]
- [20] Wu L, Li JW, Sun PJ, Hong RC, Ge Y, Wang M. DiffNet++: A neural influence and interest diffusion network for social recommendation. arXiv:2002.00844, 2021. [doi: [10.1109/TKDE.2020.3048414](https://doi.org/10.1109/TKDE.2020.3048414)]
- [21] Yin HZ, Zhou XF, Cui B, Wang H, Zheng K, Nguyen QVH. Adapting to user interest drift for POI recommendation. *IEEE Trans. on Knowledge and Data Engineering*, 2016, 28(10): 2566–2581. [doi: [10.1109/TKDE.2016.2580511](https://doi.org/10.1109/TKDE.2016.2580511)]
- [22] Nimishakavi M, Gupta M, Talukdar P. Higher-order relation schema induction using tensor factorization with back-off and aggregation. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: Association for Computational Linguistics, 2018. 1575–1584. [doi: [10.18653/v1/P18-1146](https://doi.org/10.18653/v1/P18-1146)]
- [23] Monti F, Bronstein MM, Bresson X. Geometric matrix completion with recurrent multi-graph neural networks. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 3700–3710.
- [24] Zhuang CY, Ma Q. Dual graph convolutional networks for graph-based semi-supervised classification. In: Proc. of the 2018 World Wide Web Conf. Lyon: ACM, 2018. 499–508. [doi: [10.1145/3178876.3186116](https://doi.org/10.1145/3178876.3186116)]
- [25] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for Quantum chemistry. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 1263–1272.
- [26] Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 1025–1035.
- [27] Tang JL, Hu X, Liu H. Social recommendation: A review. *Social Network Analysis and Mining*, 2013, 3(4): 1113–1133. [doi: [10.1007/s13278-013-0141-9](https://doi.org/10.1007/s13278-013-0141-9)]
- [28] Gao M, Chen LH, He XN, Zhou AY. BiNE: Bipartite network embedding. In: Proc. of the 41st Int'l ACM SIGIR Conf. on Research & Development in Information Retrieval. Ann Arbor: ACM, 2018. 715–724. [doi: [10.1145/3209978.3209987](https://doi.org/10.1145/3209978.3209987)]
- [29] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Francisco: ACM, 2016. 855–864. [doi: [10.1145/2939672.2939754](https://doi.org/10.1145/2939672.2939754)]
- [30] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. In: Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2014. 701–710. [doi: [10.1145/2623330.2623732](https://doi.org/10.1145/2623330.2623732)]
- [31] Dahl RA. Freedom and control in modern society. *American Sociological Review*, 1954, 19(6): 807–808. [doi: [10.2307/2087947](https://doi.org/10.2307/2087947)]
- [32] Kingma DP, Ba J. Adam: A method for stochastic optimization. In: Proc. of the 3rd Int'l Conf. on Learning Representations. San Diego, 2015. [doi: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980)]
- [33] Rendle S, Freudenthaler C, Gantner Z, Schmidt-Thieme L. BPR: Bayesian personalized ranking from implicit feedback. In: Proc. of the 25th Conf. on Uncertainty in Artificial Intelligence. Montreal: AUAI Press, 2009. 452–461.
- [34] Zhao T, McAuley J, King I. Leveraging social connections to improve personalized ranking for collaborative filtering. In: Proc. of the 23rd ACM Int'l Conf. on Information and Knowledge Management. Shanghai: ACM, 2014. 261–270. [doi: [10.1145/2661829.2661998](https://doi.org/10.1145/2661829.2661998)]
- [35] Yu JL, Gao M, Li JD, Yin HZ, Liu H. Adaptive implicit friends identification over heterogeneous network for social recommendation. In: Proc. of the 27th ACM Int'l Conf. on Information and Knowledge Management. Torino: ACM, 2018. 357–366. [doi: [10.1145/3269206.3271725](https://doi.org/10.1145/3269206.3271725)]
- [36] Yu JL, Gao M, Yin HZ, Li JD, Gao CM, Wang QY. Generating reliable friends via adversarial training to improve social recommendation. In: Proc. of the 2019 IEEE Int'l Conf. on Data Mining. Beijing: IEEE, 2019. 768–777. [doi: [10.1109/ICDM.2019.00087](https://doi.org/10.1109/ICDM.2019.00087)]
- [37] He XN, Deng K, Wang X, Li Y, Zhang YD, Wang M. LightGCN: Simplifying and powering graph convolution network for recommendation. In: Proc. of the 43rd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM, 2020.

639–648. [doi: 10.1145/3397271.3401063]

- [38] Yu JL, Yin HZ, Li JD, Gao M, Huang Z, Cui LZ. Enhancing social recommendation with adversarial graph convolutional networks. IEEE Trans. on Knowledge and Data Engineering, 2022, 34(8): 3727–3739. [doi: 10.1109/TKDE.2020.3033673]

附中文参考文献:

- [6] 王鑫, 陈蔚雪, 杨雅君, 张小旺, 冯志勇. 知识图谱划分算法研究综述. 计算机学报, 2021, 44(1): 235–260. [doi: 10.11897/SP.J.1016.2021.00235]
- [15] 廖国琼, 蓝天明, 黄晓梅, 陈辉, 万常选, 刘德喜, 刘喜平. 基于事件社会网络推荐系统综述. 软件学报, 2021, 32(2): 424–444. <http://www.jos.org.cn/1000-9825/6145.htm> [doi: 10.13328/j.cnki.jos.006145]
- [16] 赵姝, 刘晓曼, 段震, 张燕平, 唐杰. 社交关系挖掘研究综述. 计算机学报, 2017, 40(3): 535–555. [doi: 10.11897/SP.J.1016.2017.00535]



李邵莹(1996—), 女, 硕士生, CCF 学生会员, 主要研究领域为社会网络分析.



张丽平(1980—), 女, 硕士, CCF 专业会员, 主要研究领域为机器学习, 数据挖掘.



孟丹(1990—), 女, 博士, 主要研究领域为模式识别, 机器学习, 隐私计算在金融、政务等领域的应用.



徐辰(1988—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为大规模数据处理系统, 分布式机器学习系统, 面向新硬件的数据管理技术.



孔超(1986—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为 Web 数据管理, 海量数据挖掘, 社会网络分析.