计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
63(2): 352−377, 2026

一种社交特征自适应融合的生成式对抗网络推荐系统

钱忠胜　朱　辉　俞情媛　李玉龙　万子珑
（江西财经大学计算机与人工智能学院　南昌　330013）
（changesme@163.com）

A Generative Adversarial Networks Recommender System for Adaptive Fusion of Social Features

Qian Zhongsheng, Zhu Hui, Yu Qingyuan, Li Yulong, and Wan Zilong
（*School of Computing and Artificial Intelligence*, *Jiangxi University of Finance and Economics*, *Nanchang* 330013）

Abstract　Social recommender systems explore collaborative data behind social labels to offer personalized suggestions. Despite many users lacking explicit social connections, they share similar item interaction histories. Prior studies have tried extracting high-order implicit social features via complex metapaths, which often diminishes model practicality. Additionally, the noise in these features can reduce adaptability when combined with explicit features using concatenation or deep learning. Recently, generative adversarial networks (GANs) have supported robust data augmentation but pose challenges with model convergence, impacting efficiency in social recommendation contexts. To address these issues, we introduce AFS-GAN (generative adversarial networks recommender systems for adaptive fusion of social features), a generative adversarial recommendation model with an adaptive fusion strategy for user social features. Initially, two simple metapaths extract first-order explicit and second-order implicit social features, improving model practicality and reducing subjective bias. An adaptive factor then flexibly fuses these features, capturing the diversity of user behaviors and enhancing recommendation adaptability. The generator uses straight-through Gumbel Softmax to speed up pseudo-item generation, while the discriminator employs a quartic BPR (bayesian personalized ranking) loss function to optimize discrimination loss, thus speeding convergence and simplifying the model. Extensive comparisons on 4 benchmark recommendation datasets with 8 leading social recommendation models have demonstrated that our approach excels in *Precision*, *Recall*, and *NDCG* metrics.

Key words　metapath; adaptive factor; GAN; straight-through Gumbel Softmax; social recommender system

摘　要　社交推荐系统旨在探索社交网络用户社交标签背后的协同信息，为用户提供个性化推荐。然而，社交网络中大量的用户之间没有显式社交关系，但他们却共享相同的项目历史交互行为。以往研究者主观上期望通过复杂元路径挖掘用户间的高阶隐式社交特征，客观上却降低了模型的实用性。而且，高阶隐式社交特征中的噪声较大，根据特征拼接或深度学习的方式与显式社交特征融合后反而会降低模型的适应能力。近年来，生成式对抗网络（GAN）为数据增强提供了有力的支持，但其复杂的结构令模型收敛困难，导致其应用于社交推荐场景时使得模型整体效率不高。基于此，提出一种社交特征自适应融合的生成式对抗网络推荐模型 AFS-GAN（generative adversarial networks recommender systems for adaptive

收稿日期：2024-09-14；**修回日期**：2025-06-06
基金项目：国家自然科学基金项目（62262025）；江西省自然科学基金重点项目（20224ACB202012）；赣鄱俊才支持计划-主要学科学术和技术带头人培养项目-领军人才（学术类）（20243BCE51024）
This work was supported by the National Natural Science Foundation of China (62262025), the Key Program of the Natural Science Foundation of Jiangxi Province (20224ACB202012), and the Jiangxi Poyang Support Program for Talents of China-Major Discipline Academic and Technical Leaders Training Program of Jiangxi Province-Leading Talent (Academic) (20243BCE51024).

fusion of social features)。首先，采用 2 个简单元路径分别提取用户的 1 阶显式社交特征和 2 阶隐式社交特征，以消除研究者主观判断的不利影响，提高模型的实用性；其次，设计自适应因子灵活地融合显示和隐式社交特征，充分体现用户社交行为的多样性，提升推荐的适应能力；最后，在生成器中采用直通 Gumbel Softmax 加速生成伪项目，在判别器中采用四元 BPR（Bayesian personalized ranking）损失函数直接最大化判别损失，既简化了模型，又提升了其收敛速度，从而整体上提高了模型的效率。在 4 个基准推荐数据集上与 8 种目前较先进的社交推荐模型进行了广泛的比较，实验结果表明，所提方法在 *Precision*，*Recall*，*NDCG* 这 3 个指标表现卓越。

关键词 元路径；自适应因子；生成式对抗网络；直通 Gumbel Softmax；社交推荐系统

中图法分类号 TP399

DOI: 10.7544/issn1000-1239.202440737 **CSTR:** 32373.14.issn1000-1239.202440737

根据社交同质性的假设[1]，拥有显式社交关系（互加好友、关注、点赞等）的用户具有相同或相似的项目偏好。当目标用户选择潜在的项目时，会倾向于选择与他们有社交关系的用户已选择或高度评价的项目。但是，显式社交信息比隐式社交信息稀疏得多。例如，在 LastFM 数据集中[2-3]，有 25 434 对用户具有显式社交关系；相应地，拥有超过 10 个共享项目却没有显式社交关系的用户对的数量为 142 780，是显式社交关系数量的 5.6 倍。这说明尽管许多用户在社交网络中缺乏直接的联系，但其行为很可能与其他用户具有某种相似性，尤其是具有相同的项目历史交互记录。例如，2 个用户可能无直接的社交关系（如好友或关注），但他们可能在多个项目上有重叠的互动或相似的评价。这种基于项目交互的相似性可帮助我们推测出潜在的隐式社交关系。通过对用户共同交互历史等行为模式的深入分析，挖掘隐式社交关系是揭示用户偏好的一个重要途径。

在推荐系统中，准确地挖掘和利用这些隐式社交关系，不仅能够提高对用户偏好的理解，还能进一步提升推荐的个性化和准确性。与显式社交关系相比，隐式社交关系在大规模数据集中更为普遍，其潜力未被充分挖掘。通过综合显式和隐式社交信息，系统能够识别出那些虽然没有直接联系但在项目偏好上非常相似的用户，从而为这些用户提供更加个性化和精确的推荐。鉴于显式社交关系的稀疏性限制了社交推荐系统的性能，本文将深入探索和利用隐式社交关系，从不同的视角和途径研究提升社交推荐系统性能的新方法。

现有采用元路径和生成式对抗网络（generative adversarial network，GAN）构建社交推荐的方法存在一些不足，包括：元路径复杂、实用性差；融合后的用户特征过于依赖高阶关系，放大了随机噪声；GAN 训练效率偏低等方面。

研究动机 1：元路径过于复杂，模型的实用性差。

由于元路径可直观地描述用户之间的社交关系，以及用户和项目的交互，能为社交网络提供丰富的语义信息。因此，基于元路径的推荐系统在挖掘隐式社交关系方面取得了一定成效[4-5]。在许多现有基于元路径的社交推荐方法中，元路径设计通常涉及多个节点和复杂的关系链路。例如，Song 等人[6]设计了 6 种元路径，其中最长的元路径包含了 5 个节点。这种复杂的设计旨在捕捉多层次的社交关系，理论上能增强模型对社交行为的理解。然而，过于复杂的元路径往往会导致以下 2 个问题。

1）应用场景适应性差。很多元路径方法依赖于研究者的领域知识来设计，有很大的主观性和局限性。每当推荐场景或数据集发生变化时，需重新设计元路径。显然，这种方法的实用性受到很大限制，尤其是在面对跨领域或动态变化的社交场景时。例如，Yu 等人[7]设计的元路径虽然在特定领域内能有效提升推荐性能，但当数据分布发生变化时，元路径的有效性和灵活性无法得到保证，这使得该方法缺乏普适性。

2）计算复杂度高。随着元路径节点数目的增加，训练和推理时所需的时间和计算资源也大幅度提高，限制了模型在大规模应用中的效率。复杂的多节点路径设计需对大量的社交关系和项目交互进行精细建模，这使得模型在处理大规模数据集时出现性能瓶颈。

因此，基于复杂元路径的推荐系统在可扩展性和灵活性上存在一定的局限，难以广泛应用于实际场景。

研究动机 2：融合后的用户特征过度依赖高阶关系，放大了随机噪声。

现有基于元路径的社交推荐方法在特征融合时,直接将显式和隐式社交特征拼接或采用复杂神经网络的方式提取融合特征。尽管这些方法能捕捉到多维社交信息,但也会引入较大的噪声。

1)我们已经知道显式社交信息往往比隐式社交信息稀疏得多。当两者简单拼接时,可能导致无关信息与真正相关的社交信号混淆,从而放大了数据中的随机噪声。例如,Li 等人[8]从用户的角度设计了 4 条元路径定义用户的显式和隐式社交邻居,再通过加权和的方法将它们直接拼接为当前用户的特征。由于显式社交关系较少,而隐式社交关系的数量较多,加权和可能赋予低相关性的隐式关系过高的权重,进而干扰用户特征的表示。

2)用户的行为和偏好通常是复杂多变的,深度学习模型往往会过度依赖高阶关系,使模型学习到错误的关联模式,导致预测结果不准确。例如,Jiang 等人[9]从用户和项目的角度各自设计了 6 条和 5 条元路径,采用 GCN 和轻量级的 RNN 网络分别初始化用户特征和项目特征,再利用注意力机制生成目标用户或项目的特征。虽然该方法能在一定程度上丰富用户的社交信息,但在复杂社交网络中,冗余信息和噪声的影响仍较大,导致模型无法有效识别用户的真实偏好。

因此,特征融合方法的设计需避免引入过多的噪声,从而提高模型的准确性和稳定性。

研究动机 3:训练复杂 GAN 的难度较大,降低了推荐模型的效率。

近年来,GAN[10]在推荐系统领域已取得一定成功。IRGAN[11], CFGAN[12], GraphGAN[13]开创了将 GAN 范式集成到推荐系统的先河。IRGAN 和 CFGAN 均使用策略梯度[14]来使生成器产生非连续输出,增强其欺骗判别器的能力。GraphGAN 使用 Graph Softmax 为每个节点生成一个广度优先搜索树用于捕捉用户和项目在异构图结构中的局部邻域信息和全局结构信息。Fan 等人[15]提出一种高度复杂的社会推荐系统,该系统分别针对项目域和社交域构建 2 套独立的 GAN 框架以增强稀疏交互数据并揭示用户间的潜在社交关系。Dervishaj 等人[16]将矩阵分解与 GAN 结合,缓解隐式反馈和数据稀疏的问题。Yu 等人[17]挑选种子用户来生成能增强推荐性能的朋友,允许模型根据当前用户偏好动态预测拟推荐的项目。GAN 的优势主要体现在其强大的生成能力,可在数据稀疏或复杂的环境下生成伪数据,增强模型的训练效果。然而,GAN 的使用也带来了训练不稳定和收敛速度慢等问题。

GAN 模型的训练过程本质上是一个博弈过程,需不断调整生成器和判别器,缓解模式崩溃或梯度消失问题。这一过程常常会导致训练不稳定。

由于 GAN 模型的高复杂性和对抗性,收敛速度通常较慢,特别是在面对大规模数据集时,训练时间和计算成本显著增加。这在实际应用中对计算资源提出了更高的要求,限制了模型在资源有限环境下的应用。

因此,虽然 GAN 在增强推荐性能方面表现出色,但其较高的训练复杂度和收敛难度是目前应用中的一个问题。

针对这 3 个研究动机我们提出 3 个方面的改进:1)优化复杂的元路径,提升模型的实用性;2)改进用户特征融合方法,减少噪声的带入,增强模型的适应能力;3)简化 GAN,提升推荐模型的效率。因此本文提出一种社交特征自适应融合的生成式对抗网络推荐系统(generative adversarial networks recommender systems for adaptive fusion of social features, AFS-GAN)。首先,采用 2 个简化元路径分别提取用户的 1 阶显式和 2 阶隐式社交特征;其次,利用自适应因子灵活地融合显式和隐式社交特征,对用户社交特征进行编码;最后,在生成器中采用直通 Gumbel Softmax 加速生成伪项目,在判别器中采用四元 BPR 损失函数直接最大化判别损失,当生成器和判别器的损失函数值达到均衡时,高效生成推荐项目。本文的主要贡献有 4 点:

1)为减少对特定应用场景的依赖,本研究仅使用 2 种简单元路径来挖掘用户之间的显式和隐式社交特征,从而减少了样本探索中的复杂性和主观偏差,增强了推荐模型的实用性。

2)构建了与用户社交和项目交互直接相关的自适应因子,将显式和隐式社交特征自适应融合为用户社交特征,既避免带入过多的噪声,又获取含有丰富社交关系的用户特征,提升了模型的适应能力。

3)利用直通 Gumbel Softmax 提高伪项目的效率,采用四元 BPR 损失函数直接最大化判别损失,简化了判别器的结构,加快了模型的收敛速度,降低训练复杂度,提高了整体推荐效率。

4)在 4 个公开数据集上与 8 种目前较先进的推荐模型进行了充分对比实验,以及针对模型变体的消融实验,表明 AFS-GAN 模型在推荐性能上占优,以及表明模型中各组件的必要性。

1 相关工作

社交推荐研究可分为3个方面：社交推荐中基于元路径的特征学习、应用于社交推荐的GAN以及利用图神经网络挖掘高阶社交关系。

1.1 社交推荐中基于元路径的特征学习

已有研究证明元路径在捕获异构图中丰富语义信息方面是有效的[18]，而社交推荐网络是异构图的典型实例之一。因此，现有社交推荐广泛使用元路径来有效学习用户和项目特征。

元路径是异构图中节点和边的序列，它定义了不同节点间的特定语义关系[4]。早期的研究表明，元路径在提高推荐系统性能方面具有良好的潜力。较早的研究者提出基于元路径的异构图推荐方法[5]，已显示出该方法的优越性。在文献[19]中，研究者展示了蕴涵在用户和项目二部图中的元路径可改善推荐性能。这些研究均利用元路径来模拟用户-项目交互和用户-用户社交关系，为社交推荐开创了一个新的研究方向。

最近的研究将元路径在社交推荐领域的应用推向了新的高度。Yu等人[7]提出一种在异构图中通过种子用户识别用户间的显式和隐式社交关系的方法。研究者首先将整个社交推荐表示为异构网络，然后使用EM算法迭代优化依据6条元路径初始化的用户特征（其中，最长的元路径有5个不同类型的节点，2种类型和方向的边），进而得到最终的用户特征。在复杂元路径与处理隐变量的EM算法的共同作用下，该方法会在训练集上学到过于特殊的用户行为模式，从而导致模型在测试集中出现过拟合的现象。与该方法类似，为说明在社交网络中偏好相似的用户相距甚远，Song等人[6]同样设计了6条复杂元路径，每条元路径有2种类型的边，每条边还有2个方向，其目的是通过不同边的不同方向学习高阶的社交与交互信息。由于缺少有效的特征融合手段，该方法得到的用户和项目特征会带入高阶噪声。

Dhelim等人[20]使用3条仅有3个节点的元路径，覆盖用户兴趣、社交关系和项目，采用基于图的链接预测（graph-based link prediction）技术来预测用户和项目间的潜在链接。虽然元路径的结构简单，简化了模型的整体结构，但未能将隐式社交特征融合到用户中。

Miao等人[21]为减少复杂元路径（7条元路径，最长4个节点）引入的噪声，将具有高依赖概率的节点输入LightGCN学习高阶的用户项目特征。但是，每条元路径均依赖特定数据集中的领域知识和研究者的经验，特别是当元路径涉及多个节点类型和多层连接时，不恰当的元路径可能无法有效区分有用信号和噪声，甚至会引入新的噪声。Jiang等人[9]提出元路径应考虑更复杂的依赖关系，以及更全面地利用社交关系和项目特征，进而设计一个双编码器模块。一个编码器面向用户侧，编码6条元路径，负责捕捉相似用户；另一个面向项目侧，编码5条元路径，负责捕捉项目交互的依赖关系。

通过上述分析，我们认为以上工作存在2方面的问题：一方面，研究者期望通过数量众多且结构复杂的元路径提升特定场景的信息覆盖面，获得针对具体场景的高阶社交和项目交互特征，这可能会导致推荐模型受限于具体的应用场景，降低实用性；另一方面，由于未能充分考虑高阶特征可能引入的噪声，影响了推荐模型的适应性。针对第1个问题，本文主要弱化元路径与具体的应用场景的关联度，将重心放在直观揭示社交推荐的基本特性和交互关系上，而不是过于关注元路径的数量和覆盖面，增强模型的实用性。针对第2个问题，本文引入一种简洁且高效的自适应特征融合方法，以缓解噪声对推荐质量的影响，提升模型适应性。

1.2 应用于社交推荐的GAN

近年来，GAN网络在数据增强等领域取得显著的成效。通过模拟最小最大博弈，生成器模仿真实数据分布生成伪样本，而判别器则尝试鉴别伪样本和真实数据。在这一过程中，生成器和判别器会不断改进与优化，最终获得高质量的生成结果。

目前，将GAN专门应用于社交推荐的研究仍处于发展阶段。Li等人[22]提出一种基于用户消费地理位置兴趣点的对抗学习推荐方法。在此研究中，生成器由自注意力机制构成，生成用户和项目特征；判别器采用在损失函数中梯度加入人为扰动的方法，增强模型对输入数据变化的鲁棒性。然而，引入对抗扰动会使模型学到的特征偏离真实数据的自然分布，无法准确反映实际应用中的数据特性，而且对于梯度的扰动会降低推荐性能和泛化能力。

Fan等人[15]使用一个双GAN框架DASO，用于学习项目域和社交域中的交互和社交信息。在项目域中，生成器生成虚拟的用户-项目交互；在社交域中，生成器生成虚拟的用户-社交关系。相应地，判别器旨在准确鉴别生成器输出的虚拟信息与真实信息。

对于项目域，GAN 有助于提高模型预测用户偏好的准确性；对于社交域，GAN 有助于模拟用户的社交动态，为社交推荐提供丰富的辅助信息。从 2 个 GAN 中获得的项目、社交特征被输入 MLP 用于计算用户对于目标项目的评分。虽然这种双 GAN 模型在理论上能够从多个维度捕捉用户行为和社交动态，为推荐系统提供丰富的特征输入，但项目域和社交域的特征在本质上差异很大，通过 MLP 可能无法充分捕捉这些特征间复杂的相互关系。同时，将 2 个 GAN 整合到 1 个推荐模型中会加剧训练不稳定和难以收敛现象的发生。

Yu 等人[17] 提出一种社交推荐框架 RSGAN，首先为每个用户选择少数社交相似度高的种子用户，再通过生成器模拟出逼近用户社交偏好的“新用户”，判别器评估这些新用户对于推荐结果的贡献，以便更准确地预测用户偏好。前面我们已知，社交推荐任务中的用户社交关系是较稀疏的，基于少数种子用户生成的特征无法全面反映用户的社交偏好，使得推荐系统只能从一个较小的、偏差的视角观察用户，影响系统发掘更广泛社交特征的能力，从而影响其实用性和准确性。

Dervishaj 等人[16] 在生成器中采用矩阵分解的方法将用户 ID 映射为用户偏好特征，再输入判别器中利用自编码器重构用户特征，并最大化该特征与生成器输出特征间的差异。尽管该模型在一定程度上缓解了隐式反馈数据稀疏的问题，但它未利用社交信息来提高推荐准确性，而且在判别器中最大化自编码器的输入输出差异，会导致模型训练不稳定，收敛困难。

通过上述分析，我们认为在用户交互和显式社交数据稀疏的情况下，采用 GAN 生成伪用户或伪项目，对于提高推荐质量至关重要，但也暴露出复杂的网络结构会引起推荐模型效率差的问题。针对这种情况，本文在保持 GAN 网络数据增强优势的前提下，引入直通 Gumbel Softmax 技术来提升生成器的梯度反向传播速度，直接采用 BPR 损失函数替代神经网络构建判别器，从而提升模型整体效率。

1.3 利用图神经网络挖掘高阶社交关系

除了 1.1 节和 1.2 节所述的 2 类方法外，社交推荐领域中另一个典型的手段是利用图神经网络挖掘高阶社交关系。

Chen 等人[23] 利用社交上下文感知函数自适应地为不同的交互数据分配不同的置信度，并基于个性化随机游走的采样策略，从大规模未观察数据中选取信息量丰富的训练实例，进而加速梯度估计并减少采样方差。该方法过于依赖社交网络，认为社交关系显著影响了用户偏好。当显式社交关系稀疏时，采用该方法可能会导致推荐性能下降。Wu 等人[24] 通过模拟社交影响的递归扩散过程来提升社交推荐的效果，从而优化用户和项目的特征。在此基础上，他们通过同时建模社交网络和兴趣网络中的高阶社交影响扩散过程来提高推荐效果[25]。文献 [23−24] 的方法会使社交信息中冗余部分在扩散过程中不断累加和放大，从而不能有效缓解社交信息在高阶社交网络中的噪声传播问题。

Yu 等人[26] 利用 GCN 捕捉社交关系的多层次信息以提升推荐的个性化水平。但引入对抗训练会导致训练不稳定、收敛速度慢。随后，他们通过自监督机制结合多通道超图卷积网络来处理多维度的社交关系，对不同类型的社交关系进行独立建模，使模型能够更全面地捕捉社交行为中的复杂性和多样性[27]。但是，当社交信息冗余时，多通道模型易受到信息过载影响。另外，他们还采用自监督机制帮助模型根据用户的历史行为自动生成相关的预测项目，比较适应数据稀疏的场景[28]。但是，在应对高阶社交关系时，该方法过度依赖于显式社交信息，而忽视了隐式社交关系的作用。

Liu 等人[29] 提出一种基于 GNN 的会话式社交推荐模型 GNNRec。该模型旨在同时处理用户的会话信息和社交网络信息，缓解传统会话推荐模型仅依赖用户历史交互而忽略社交关系的问题。GNNRec 引入图注意力网络自适应地聚合社交信息，虽然能为不同节点分配权重，但在显式社交数据稀疏的情况下，可能无法有效区分有效的社交信息和噪声，会出现过度依赖有限显式社交信息的情况，导致模型在社交信息稀疏的环境下表现不稳定或过度拟合。Yang 等人[30] 提出的社交推荐方法 GBSR 通过对社交图去噪来强化推荐任务的表现，从而提升推荐系统的鲁棒性。尽管该方法有效减小了冗余社交关系对推荐结果的负面影响，但其在去噪过程中可能会过度删除有价值的社交连接，导致推荐系统对某些用户的偏好捕捉不准确。Wang 等人[31] 将 GNN 视为平滑正则器，增强模型对样本分布变化的鲁棒性。针对数据稀疏问题，该方法增加图中的边，扩展目标用户的邻居分布，引入非邻居节点作为伪邻居，从而扩大训练集样本量，有效缓解稀疏问题。当数据集的分布变化较大时，该方法易发生过拟合，导致模型的泛化性降低。

Chen 等人[32]采用 Transformer 架构处理包含正负反馈的符号图，图中使用符号谱编码和符号路径编码 2 种位置编码分别从图的谱特性和路径信息的角度对图进行编码，使模型能充分利用图中的协同信息。在此基础上，他们继续在 Transformer 模型中根据全局和特定层的注意力权重区分正负样本，使模型能更好地捕捉用户之间、项目之间的协同信号[33]。

通过上述分析，我们认为利用图神经网络挖掘高阶社交关系的方法能捕捉社交关系的多层次信息，从而增强推荐系统的性能。然而，这些方法也暴露出一些不足，例如，在网络传播过程中无法有效缓解冗余信息的积累，导致噪声在传播过程中不断放大。尽管显式社交数据稀疏，但模型还是会过度依赖此类有限信息，忽视了隐式社交关系的潜在价值，进而导致预测结果不稳定或过度拟合。此外，部分方法对高阶社交关系的建模过于复杂，导致训练的时间成本过高，限制了模型的适用性和普适性。因此，结合本文的 3 个研究动机，我们设计一种新颖的社交推荐方法 AFS-GAN，能够更有效地挖掘和利用隐式社交特征，缓解数据稀疏现象，提高推荐系统的准确性与适用性。

2 AFS-GAN 模型的构建

AFS-GAN 模型的整体框架如图 1 所示，主要包括 4 个模块。

1）用户社交特征的构建与自适应融合。通过 UTU 和 UIU 元路径分别得到用户的 1 阶显式社交特征和 2 阶隐式社交特征，经过自编码器重构后融合为用户自适应社交特征。

2）伪项目生成器。经过 ST-GS 层重参数化后的用户自适应社交特征转化为独热向量，可在项目偏好矩阵中查询得到与用户有交互历史的项目，再通过 ST-GS 层重参数化后得到伪项目。伪项目生成器的目标是最小化伪项目与真实项目之间的差异。

3）伪项目判别器。通过最大化伪项目与真实项目之间的差异，迫使伪项目生成器不断优化。

4）模型训练与推荐。当生成器与判别器的损失函数优化迭代至均衡时，伪项目就代表了用户的真实偏好，其列表即为模型的最终推荐列表。

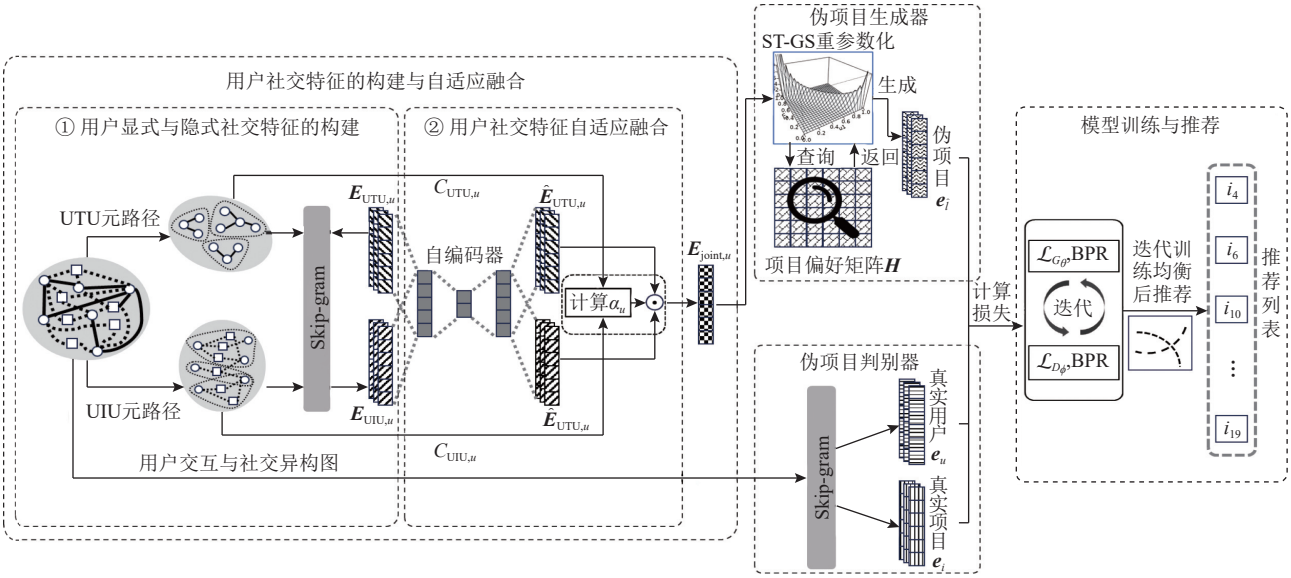


Fig. 1 The framework of AFS-GAN model

图 1 AFS-GAN 模型框架

2.1 问题定义与符号说明

给定用户集 $U=\{u_1,u_2,\cdots,u_m\}$ 与项目集 $I=\{i_1,i_2,\cdots,i_n\}$，以及它们的交互矩阵 $\boldsymbol{A}\in\mathbb{R}^{m\times n}$，其中 m 和 n 分别表示用户数和项目数。若用户 u 与项目 i 交互，则 $r_{ui}=1$，否则 $r_{ui}=0$。若用户 u 与 u' 具有显式社交关系，则 $r_{uu'}=1$，否则 $r_{uu'}=0$。用户与项目的交互，以及用户间的社交关系构成异构图 $G(V,E)$，其中节点集 $V=U\cup I$ 包含所有的用户和项目节点，边集 $E=\{r_{ui}|r_{ui}=1\}\cup\{r_{uu'}|r_{uu'}=1\}$ 表示节点的交互及社交关系集。本文模型的任务是向用户 u 推荐项目 i 的列表。为便于阐述，本文中一些主要符号如表 1 所示。

2.2 用户社交特征的构建与自适应融合

为最大程度保留并充分利用异构图中的用户社交信息，我们先依据元路径搜索的方式定位潜在用户，

Table 1 Description of Main Symbols

表 1 主要符号说明

符号	说明
$\boldsymbol{e}_u$	真实用户的嵌入向量
$\boldsymbol{e}_i$	真实项目的嵌入向量
$\boldsymbol{e}_{i^+}$	表示用户正交互的项目
$\boldsymbol{e}_{i^-}$	表示用户负交互的项目
$\boldsymbol{E}_{\mathrm{UTU},u}$	用户 1 阶显式社交特征的嵌入向量
$\boldsymbol{E}_{\mathrm{UIU},u}$	用户 2 阶隐式社交特征的嵌入向量
$\boldsymbol{E}_{\mathrm{joint},u}$	用户自适应融合社交特征的嵌入向量
$\boldsymbol{e}_{\hat{i}}$	伪项目的嵌入向量
$\boldsymbol{H}$	项目偏好矩阵
x_{ui^+}	真实用户 u 与正交互项目 i^+ 的交互偏好
x_{ui^-}	真实用户 u 与负交互项目 i^- 的交互偏好
$x_{u\hat{i}}$	真实用户 u 与伪项目 $\hat{i}$ 的交互偏好
σ	Sigmoid 函数
η	学习率
θ	生成器参数
ϕ	判别器参数

再采用 Skip-gram 模型[34] 提取显式和隐式社交特征。

2.2.1 用户显式与隐式社交特征的构建

从元路径搜索和 Skip-gram 模型编码 2 个阶段获得用户的显式与隐式社交特征。

1)通过元路径搜索 1 阶显式和 2 阶隐式社交用户

对应“研究动机 1”, 即“优化复杂的元路径, 提升模型的实用性”, 我们提出一种根据简单元路径搜索与目标用户具有显式与隐式社交关系用户的方法。如图 2 左图所示, 异构图 $G(V, E)$ 中不同的节点表示用户或项目, 不同的边表示用户和项目之间的社交关系或交互行为。通过 2 个元路径 UTU 和 UIU 将

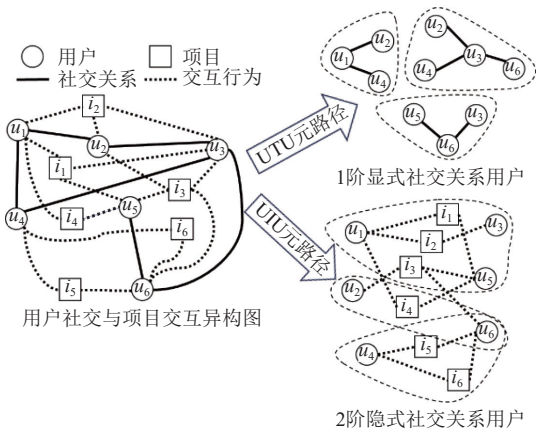


Fig. 2 Searching for first-order explicit and second-order implicit social users via metapaths

图 2 通过元路径搜索 1 阶显式和 2 阶隐式社交用户

图 2 左图分解为图 2 右上图和右下图。其中, UTU 为显式社交元路径, UIU 为隐式社交元路径。

①依据 UTU 构建 1 阶显式社交用户集。元路径 UTU 中 2 个用户节点间通过一个无向边连接, 表示他们之间存在显式的社交关系。据此元路径, 图 2 右上图围绕不同用户构建相应的同质子图, 标识出具有显式社交关系的 1 阶用户集。

尽管 2 阶显式社交关系可通过信息传播计算得到, 并有可能进一步增强模型对显式社交的建模能力, 但引入它会显著增加模型的计算复杂度, 影响训练和预测效率。现有研究也表明[35-37], 基于 1 阶显式社交关系的推荐已能充分捕捉用户的社交偏好。进一步引入 2 阶信息反而可能带来过拟合的风险。因此, 在确保推荐结果的有效性和准确性的前提下, 本文聚焦于 1 阶显式社交关系, 尽可能避免增加不必要的计算。

②依据 UIU 构建 2 阶隐式社交用户集。元路径 UIU 表示 2 个用户节点间无显式社交关系, 但具有相同的历史交互项目。据此元路径, 图 2 右下图可识别出目标用户无显式社交关系但具有相同历史交互项目的用户集。该元路径对于挖掘隐式社交关系至关重要, 在许多情况下, 用户之间的交互行为或偏好并不直接通过显式社交关系体现, 而是通过共同参与或互动的项目反映出来。因此, 利用 UIU 元路径可揭示那些看似无显式社交关系但通过交互项目存在共同潜在偏好的用户。这种方式不仅能更全面地捕捉用户的社交行为, 还能弥补仅通过显式社交关系挖掘用户偏好的局限性。

图 2 右下图提供了一个不同的观察视角。基于传统的社交同质性假设, 推荐系统主要利用显式社交关系来丰富用户的特征表示。在现实应用中, 不具有显式社交的用户数量往往很多, 而他们的项目交互行为比较相似。因此, 通过 UIU 元路径发现这些用户, 可有效地利用传统社交推荐系统所忽视的隐式社交信息, 进而增强推荐的个性化和多样性。

2)通过 Skip-gram 模型编码 1 阶显式和 2 阶隐式社交特征

我们采用图神经网络中经典的 Skip-gram 模型将前一步得到的用户投影到高维向量空间中并编码得到 1 阶显式和 2 阶隐式社交特征, 即 $\boldsymbol{E}_{\mathrm{UTU},u}$ 和 $\boldsymbol{E}_{\mathrm{UIU},u}$。具体为以下 2 个步骤。

①选择训练样本。在 2 个用户集中, 将每个用户作为中心节点, 处于滑动窗口范围内的节点视为正样本, 其他节点作为负样本。已有研究[38-39] 发现, 数

据特性显著影响窗口大小的选择，稀疏数据适合较小窗口，可聚焦短期行为，而稠密数据集需较大窗口，适用于捕捉长期行为信息。这些研究通过实验说明，窗口大小为 5 时可有效捕获上下文信息。故本研究将窗口大小取值为 5，主要基于 3 点考虑：第一，该参数的取值在跨领域研究中被验证为局部与全局信息的平衡点；第二，实验表明该参数的取值可使模型在稀疏与稠密场景中均能保持良好的基准性能；第三，固定该参数的取值可降低模型复杂度，避免动态调整带来的计算负担。我们的实验结果也表明，该设定在保证推荐效果的同时，有效支撑了研究目标的高效达成。

②学习嵌入向量。将训练样本送入 Skip-gram 模型以获得用户的高维嵌入向量。Skip-gram 模型的目标是最大化对数似然函数，并对训练样本进行迭代优化，如式(1)(2)所示。

$$\mathcal{L}=\log\left(\sigma\left(\boldsymbol{e}_{u^+}^{\mathrm{T}}\boldsymbol{e}_u\right)\right)+\sum_{i=1}^{l}\log\left(\sigma\left(-\boldsymbol{e}_{u^-}^{\mathrm{T}}\boldsymbol{e}_u\right)\right), \tag{1}$$

$$\begin{cases}\boldsymbol{e}_{u^+}\leftarrow\boldsymbol{e}_{u^+}-\eta\dfrac{\partial\mathcal{L}}{\partial\boldsymbol{e}_{u^+}},\\ \boldsymbol{e}_{u^-}\leftarrow\boldsymbol{e}_{u^-}-\eta\dfrac{\partial\mathcal{L}}{\partial\boldsymbol{e}_{u^-}},\\ \boldsymbol{e}_u\leftarrow\boldsymbol{e}_u-\eta\dfrac{\partial\mathcal{L}}{\partial\boldsymbol{e}_u},\end{cases} \tag{2}$$

其中$\boldsymbol{e}_{u^+}$，$\boldsymbol{e}_{u^-}$，$\boldsymbol{e}_u$分别是正样本、负样本和中心用户的嵌入向量，l是负样本数，$\sigma(\cdot)$是 Sigmoid 函数。通过使用随机梯度下降优化器，当迭代次数达到阈值即可得到$\boldsymbol{E}_{\mathrm{UTU},u}$和$\boldsymbol{E}_{\mathrm{UIU},u}$，具体生成过程如算法 1 所示。

算法 1. 社交特征$\boldsymbol{E}_{\mathrm{UTU},u}$与$\boldsymbol{E}_{\mathrm{UIU},u}$生成算法。

输入：异构图$G(V,E)$，元路径 UTU 和 UIU，迭代阈值，嵌入维度，负样本数l；

输出：1 阶显式社交特征$\boldsymbol{E}_{\mathrm{UTU},u}$，2 阶隐式社交特征$\boldsymbol{E}_{\mathrm{UIU},u}$。

① 初始化正样本嵌入向量$\boldsymbol{e}_{u^+}$，负样本嵌入向量$\boldsymbol{e}_{u^-}$，中心用户嵌入向量$\boldsymbol{e}_u$；
② for 根据 UTU 元路径搜索到的用户 do
③ 生成节点序列以学习嵌入向量；
④ 计算损失函数；/*见式(1)*/
⑤ 迭代更新嵌入向量，生成$\boldsymbol{E}_{\mathrm{UTU},u}$；/*见式(2)*/
⑥ end for
⑦ for 根据 UIU 元路径搜索到的用户 do
⑧ 生成节点序列以学习嵌入向量；
⑨ 计算损失函数；/*见式(1)*/
⑩ 迭代更新嵌入向量，生成$\boldsymbol{E}_{\mathrm{UIU},u}$；/*见式(2)*/
⑪ end for
⑫ 输出$\boldsymbol{E}_{\mathrm{UTU},u}$，$\boldsymbol{E}_{\mathrm{UIU},u}$。

算法 1 的核心有 2 个部分，行②~⑥生成$\boldsymbol{E}_{\mathrm{UTU},u}$，行⑦~⑪生成$\boldsymbol{E}_{\mathrm{UIU},u}$。

2.2.2 用户社交特征自适应融合

对应“研究动机 2”，即“改善用户特征融合方法，减少噪声的引入”，我们提出一种自适应的方法来融合用户显式与隐式社交特征。

1)采用自编码器学习社交特征的最佳低维表示

自编码器通过其编码器部分压缩输入的嵌入向量到一个较低维度的潜在空间，然后根据解码器部分尝试重构原始输入，这个过程强制模型捕捉数据中最重要的特征信息。在社交推荐系统中，自编码器可自适应调整其参数以最大优化重构输入数据(即显式和隐式社交特征)。这意味着模型可根据数据本身的结构和内在关系动态调整其学习过程，促使其有效地捕获用户社交特征，同时尽可能减少噪声。通过这种方式融合得到的特征可更全面、准确地代表用户的社交关系，增强推荐模型的精度和效果。现有研究[40-42]表明，3 层自编码器网络结构在计算复杂度和性能之间有较好的平衡性，故我们也采用 3 层编码器和解码器，$\boldsymbol{E}_{\mathrm{UTU},u}$的编码如式(3)所示。

$$\boldsymbol{Z}_{\mathrm{UTU},u}=f(\boldsymbol{W}_3\cdot f(\boldsymbol{W}_2\cdot f(\boldsymbol{W}_1\cdot\boldsymbol{E}_{\mathrm{UTU},u}+b_1)+b_2)+b_3), \tag{3}$$

其中$\boldsymbol{Z}_{\mathrm{UTU},u}$为编码后的潜在表示，$\boldsymbol{W}_1$，$\boldsymbol{W}_2$，$\boldsymbol{W}_3$为编码器的权重，$b_1$，$b_2$，$b_3$为编码器的偏置，$f(\cdot)$代表激活函数 RELU。$\boldsymbol{Z}_{\mathrm{UTU},u}$的解码如式(4)所示。

$$\hat{\boldsymbol{E}}_{\mathrm{UTU},u}=f(\boldsymbol{W}_6\cdot f(\boldsymbol{W}_5\cdot f(\boldsymbol{W}_4\cdot\boldsymbol{Z}_{\mathrm{UTU},u}+b_4)+b_5)+b_6), \tag{4}$$

其中$\hat{\boldsymbol{E}}_{\mathrm{UTU},u}$为重构后的显式社交特征，$\boldsymbol{W}_4$，$\boldsymbol{W}_5$，$\boldsymbol{W}_6$为解码器的权重，$b_4$，$b_5$，$b_6$为解码器的偏置。因网络结构相同，$\boldsymbol{E}_{\mathrm{UIU},u}$的编码、解码分别同式(3)和式(4)，不再赘述。自编码器的损失$\mathcal{L}_{AE}$为均方误差，如式(5)所示。

$$\mathcal{L}_{AE}=\left\|\hat{\boldsymbol{E}}_{\mathrm{UTU},u}-\boldsymbol{E}_{\mathrm{UTU},u}\right\|^2+\left\|\hat{\boldsymbol{E}}_{\mathrm{UIU},u}-\boldsymbol{E}_{\mathrm{UIU},u}\right\|^2+\lambda_1\left(\|\psi_{\mathrm{enc}}\|^2+\|\psi_{\mathrm{dec}}\|^2\right), \tag{5}$$

其中λ_1是正则化系数，ψ_{enc}和ψ_{dec}分别是编码器和解码器参数。

2)获取自适应用户社交融合特征$\boldsymbol{E}_{\mathrm{joint},u}$

通过以下 2 个步骤，提取自适应用户社交融合特征$\boldsymbol{E}_{\mathrm{joint},u}$。

①计算基于用户社交行为的自适应因子α_u：

$$\alpha_u=\frac{C_{\mathrm{UTU},u}}{C_{\mathrm{UTU},u}+C_{\mathrm{UIU},u}}, \tag{6}$$

其中$C_{\mathrm{UTU},u}$表示用户u拥有显式社交 1 跳连接的用户数，$C_{\mathrm{UIU},u}$表示用户u与无显式社交连接但共享交互项目的 2 跳用户数。

例 1. 在图 2 中, 假设用户 u_1 通过 UTU 元路径搜索到的显式社交用户集为 $\{u_2,u_4\}$, 则 $C_{\mathrm{UTU},u}$=2。通过 UIU 元路径搜索到的隐式社交用户集为 $\{u_3,u_5\}$, 则 $C_{\mathrm{UIU},u}$=2。通过式(6)计算 α_{u_3}=1/2。

②计算自适应融合社交特征 $\boldsymbol{E}_{\mathrm{joint},u}$。我们通过加权哈达马积(⊙)融合 $\hat{\boldsymbol{E}}_{\mathrm{UTU},u}$ 和 $\hat{\boldsymbol{E}}_{\mathrm{UIU},u}$ 得到用户 u 的 $\boldsymbol{E}_{\mathrm{joint},u}$:

$$\boldsymbol{E}_{\mathrm{joint},u}=\left(\alpha_u\cdot\hat{\boldsymbol{E}}_{\mathrm{UTU},u}\right)\odot\left((1-\alpha_u)\cdot\hat{\boldsymbol{E}}_{\mathrm{UIU},u}\right)。\tag{7}$$

例 2. 从例 1 的结果可知, 用户 u_1 的自适应融合社交特征 $\boldsymbol{E}_{\mathrm{joint},u_1}=\left(\alpha_{u_1}\cdot\hat{\boldsymbol{E}}_{\mathrm{UTU},u_1}\right)\odot\left((1-\alpha_{u_1})\cdot\hat{\boldsymbol{E}}_{\mathrm{UIU},u_1}\right)$。

通过例 1 和例 2 可明显发现每个用户可根据自己的显式和隐式社交关系来调整融合因子 α_u, 确保融合特征可更好地反映个体用户的社交特性。这种基于用户的实际社交模式定制的特征融合方式可帮助模型更准确地理解用户的社交动态和偏好, 从而在推荐时提供更为精确的个性化项目。生成自适应融合社交特征 $\boldsymbol{E}_{\mathrm{joint},u}$ 的具体过程如算法 2 所示。

自适应融合特征 $\boldsymbol{E}_{\mathrm{joint},u}$ 实现了去噪, 降低了冗余或无关社交信息的干扰, 可根据用户的历史行为动态调整社交特征, 有效减少了噪声干扰。这样的特征融合方式确保了模型在处理社交信息时能更精确地捕捉用户的真实社交信号, 进而提升推荐的整体性能。

算法 2. 自适应融合社交特征 $\boldsymbol{E}_{\mathrm{joint},u}$ 生成算法。

输入: 1 阶显式社交特征 $\boldsymbol{E}_{\mathrm{UTU},u}$, 2 阶隐式社交特征 $\boldsymbol{E}_{\mathrm{UIU},u}$, 损失函数超参数 λ_1, 学习率 η;

输出: 自适应融合社交特征 $\boldsymbol{E}_{\mathrm{joint},u}$。

① 初始化编码器和解码器的参数 ψ_{enc} 和 ψ_{dec};

② 迭代优化损失函数并更新 ψ_{enc} 和 ψ_{dec}, 生成 $\hat{\boldsymbol{E}}_{\mathrm{UTU},u}$ 和 $\hat{\boldsymbol{E}}_{\mathrm{UIU},u}$; /*见式(3)~(5)*/

③ for 每个用户 u do

④ 遍历 u 的 1 跳显式社交连接的用户;

⑤ 统计用户数, 计算 $C_{\mathrm{UTU},u}$;

⑥ 遍历与 u 无显式社交连接但共享交互项目的 2 跳隐式用户;

⑦ 统计用户数, 计算 $C_{\mathrm{UIU},u}$;

⑧ 计算自适应因子 α_u; /*见式(6)*/

⑨ end for

⑩ 生成 $\boldsymbol{E}_{\mathrm{joint},u}$; /*见式(7)*/

⑪ 输出 $\boldsymbol{E}_{\mathrm{joint},u}$。

在算法 2 中, 行①②利用自编码器得到重构后的特征, 行③~⑨计算自适应因子, 行⑩生成 $\boldsymbol{E}_{\mathrm{joint},u}$。

2.3 伪项目生成器

对应“研究动机 3”, 即“简化 GAN, 提升模型的效率”, 我们提出一种相对简化的 GAN, 在本节与 2.4 节中详细介绍。

伪项目生成器的目标是通过对用户行为模式建模生成符合用户偏好的伪项目, 进而提升推荐的准确性。伪项目建立在 $\boldsymbol{E}_{\mathrm{joint},u}$ 基础之上, 蕴涵了丰富的用户显式和隐式社交信息。因此, 经过 GAN 迭代训练后, 伪项目与隐式社交关系有着紧密的联系, 能够有效捕捉那些未能通过显式社交关系直接体现的用户行为模式。

用户-项目交互数据的离散性质使得生成器的损失函数变得不可微分, 不能采用常规的方法更新梯度优化模型。对于此情况, 以往的研究通常采用重参数化技术, 如传统的 Gumbel Softmax 方法[43-45], 来估计不可微分函数的梯度。尽管这种技术可通过重新参数化技巧进行端到端的梯度优化, 使得梯度的估计在理论上是无偏的, 但其渐进性质和在离散选择上的精度尚未达到最优, 且在高维设置下效率较低, 学习过程可能仍较缓慢。

本文采用直通 ST-GS(straight through Gumbel-Softmax)[43] 方法, 其主要原因有 3 点: ①ST-GS 通过引入 Gumbel-Softmax 分布来连续化离散的输出空间, 使得原本不可微分的损失函数可通过梯度下降方法优化。这种方法通过在前向传播中使用离散采样, 在反向传播中使用连续的 Softmax 近似, 允许使用标准的反向传播算法进行平滑的梯度估计, 有助于减少梯度估计的方差。②ST-GS 通过对概率输出的全概率分布进行编码, 能从每个训练实例中学习更多信息, 而不是仅依赖于单个采样结果, 从而提高对训练数据的利用效率, 加速学习过程。③直通技术简化了模型设计, 使得离散变量能更直接地集成到神经网络中, 减少训练过程中梯度消失的不利现象。伪项目生成器的构造过程包括 2 个步骤:

第 1 步。将 $\boldsymbol{E}_{\mathrm{joint},u}$ 输入 ST-GS 层并输出 $\hat{\boldsymbol{E}}_{\mathrm{joint},u}$, 如式(8)所示。

$$\begin{cases}\hat{E}^1_{\mathrm{joint},u}=\dfrac{\exp\left(\left(\log E^1_{\mathrm{joint},u}+g_1\right)/\tau\right)}{\sum\limits_{i=1}^{D}\exp\left(\left(\log E^i_{\mathrm{joint},u}+g_i\right)/\tau\right)},\\ \hat{E}^2_{\mathrm{joint},u}=\dfrac{\exp\left(\left(\log E^2_{\mathrm{joint},u}+g_1\right)/\tau\right)}{\sum\limits_{i=1}^{D}\exp\left(\left(\log E^i_{\mathrm{joint},u}+g_i\right)/\tau\right)},\\ \vdots\\ \hat{E}^D_{\mathrm{joint},u}=\dfrac{\exp\left(\left(\log E^D_{\mathrm{joint},u}+g_1\right)/\tau\right)}{\sum\limits_{i=1}^{D}\exp\left(\left(\log E^i_{\mathrm{joint},u}+g_i\right)/\tau\right)},\\ \hat{\boldsymbol{E}}_{\mathrm{joint},u}=\left(\hat{E}^1_{\mathrm{joint},u},\ \hat{E}^2_{\mathrm{joint},u},\ \cdots,\ \hat{E}^D_{\mathrm{joint},u}\right)^{\mathrm{T}},\end{cases}\tag{8}$$

其中$E^i_{\text{joint},u}$是向量$\boldsymbol{E}_{\text{joint},u}$的第$i$个元素，$D$是嵌入维度。$g_i$是添加的Gumbel噪声，$g_i=-\log\left(-\log(x_i)\right)$，$x_i\sim Uniform(0,1)$，$\tau$是温度参数，它控制输出的嵌入向量接近独热(one-hot)向量的程度，τ越接近0，输出就越接近独热向量。

第2步。在ST-GS层输出$\hat{\boldsymbol{E}}_{\text{joint},u}$后，通过与项目偏好矩阵$\boldsymbol{H}$做哈达马积的方式查询得到伪项目特征$\boldsymbol{e}_{\hat{i}}$：

$$\boldsymbol{e}_{\hat{i}}=one\text{-}hot(\arg\max_i(\hat{\boldsymbol{E}}^i_{\text{joint},u}))\odot\boldsymbol{H}, \tag{9}$$

其中$one\text{-}hot(\arg\max(\cdot))$表示将嵌入向量转换为独热编码。矩阵$\boldsymbol{H}\in\mathbb{R}^{m\times n}$可表示用户对候选项目的动态偏好权重，其核心作用是通过与$\hat{\boldsymbol{E}}_{\text{joint},u}$做哈达玛积，得到伪项目的嵌入向量。为避免引入噪声，矩阵$\boldsymbol{H}$中的元素初始值服从[0,1]均匀分布，可确保它的偏好权重均相同。获得伪项目的初始化嵌入向量后，通过对抗训练的梯度反向传播，对$\boldsymbol{H}$持续更新，最终生成逼近用户真实偏好的伪项目。

我们采用贝叶斯个性化排名(Bayesian personalized ranking, BPR)[46]作为损失函数，目标是最小化伪项目$\boldsymbol{e}_{\hat{i}}$与真实项目$\boldsymbol{e}_{i^+}$之间的差异。其基本思想是提高$\boldsymbol{e}_{\hat{i}}$在推荐列表中的排序位置，使其接近用户交互的正项目$\boldsymbol{e}_{i^+}$，如式(10)所示。

$$\mathcal{L}_{G_\theta,\text{BPR}}=-\sum_{u,i^+,\hat{i}}\log\sigma(x_{ui^+}-x_{u\hat{i}})+\lambda_2\|\theta\|^2, \tag{10}$$

其中θ是生成器的参数，λ_2是其正则化系数，$\sigma(\cdot)$是Sigmoid函数。x_{ui^+}和$x_{u\hat{i}}$分别代表真实用户u与正交互项目i^+、伪项目$\hat{i}$的交互偏好。

最小化该损失函数后得到的伪项目与用户已交互的真实项目是高度相似的。据此，生成器可为判别器提供难以区分的伪项目。

2.4 伪项目判别器

针对“研究动机3”，判别器并没有采用常见的神经网络，而是直接将伪项目输入到四元BPR构成的损失函数中，通过最大化判别器内的排序差异来确保它能有效地区分伪项目和真实项目，从而指导生成器创建更逼真的伪项目，如式(11)所示。

$$\mathcal{L}_{D_\phi,\text{BPR}}=-\sum_{u,i^+,\hat{i},i^-}(\log\sigma(x_{ui^+}-x_{u\hat{i}})+\log\sigma(x_{u\hat{i}}-x_{ui^-}))+\lambda_3\|\phi\|^2, \tag{11}$$

其中ϕ是判别器的参数，λ_3是其正则化系数，x_{ui^-}是真实用户u与负交互项目i^-的交互偏好。

在判别器的BPR损失函数中，伪项目与正负样本同时比较，有效地利用了Sigmoid函数将偏好评分的差异映射到概率空间中，从而使得模型能够清楚地学习到哪些是用户交互过的项目(正样本)，哪些是未交互过的项目(伪项目或负样本)。这种明确的对比有助于判别器在每次迭代中尽可能找出真实项目与伪项目间的差异，进而不断推动生成器优化伪项目。

2.5 模型训练与推荐

由2.3节和2.4节构造的GAN增强了推荐模型对伪项目的鉴别能力，从而提高模型在面对真实世界多样数据时的推荐质量。通过生成器与判别器之间的迭代训练，模型能更好地理解用户的真实偏好，从而提供更为精准的推荐。此外，这还可防止模型过于依赖生成器的输出，避免了因生成器输出的结果不佳而导致的推荐质量的波动。接下来对AFS-GAN模型进行训练与预测。

模型训练的目标是通过持续的迭代优化，使得生成器和判别器达到最小最大博弈的平衡状态，即生成器产生伪项目的能力与判别器区分伪项目和真实项目的能力是平衡的。形式上，判别器对真实x的判断概率可表示为$P(D(x))=P(D(G(y)))=0.5P(D(x))$，$P(D(G(y)))$是判别器对生成器产生的$y$的判断概率。在平衡状态下，判别器的排序输出可作为AFS-GAN模型的最终推荐列表。AFS-GAN的目标函数为

$$\mathcal{L}_{G_\theta,D_\phi,\text{BPR}}=\min_{D_\phi}\max_{G_\theta}\left(\mathcal{L}_{G_\theta,\text{BPR}}+\mathcal{L}_{D_\phi,\text{BPR}}\right)。 \tag{12}$$

生成器和判别器的参数将分别采用随机梯度上升和下降的方式进行更新，如式(13)(14)所示。

$$\theta^{t+1}=\theta^t+\eta\times\nabla_\theta\mathcal{L}_{G_\theta,\text{BPR}}(f_\theta(\boldsymbol{e}_{i^+}\mid\boldsymbol{e}_u)-f_\theta(\boldsymbol{e}_{\hat{i}}\mid\boldsymbol{e}_u)), \tag{13}$$

$$\phi^{t+1}=\phi^t-\eta\times\nabla_\phi\mathcal{L}_{D_\phi,\text{BPR}}((f_\phi(\boldsymbol{e}_{i^+}|\boldsymbol{e}_u)-f_\phi(\boldsymbol{e}_{\hat{i}}|\boldsymbol{e}_u))+(f_\phi(\boldsymbol{e}_{\hat{i}}|\boldsymbol{e}_u)-f_\phi(\boldsymbol{e}_{i^-}|\boldsymbol{e}_u))), \tag{14}$$

其中$f_\theta(\cdot|\boldsymbol{e}_u)$和$f_\phi(\cdot|\boldsymbol{e}_u)$分别计算用户$u$在生成器和判别器中与项目$\boldsymbol{e}_{i^+}$，$\boldsymbol{e}_{i^-}$，$\boldsymbol{e}_{\hat{i}}$交互的偏好程度。AFS-GAN模型训练与预测的具体过程如算法3所示。

算法3. AFS-GAN模型训练与预测算法。

输入：自适应融合社交特征$\boldsymbol{E}_{\text{joint},u}$，正样本嵌入向量$\boldsymbol{e}_{u^+}$，负样本嵌入向量$\boldsymbol{e}_{u^-}$，中心用户嵌入向量$\boldsymbol{e}_u$，伪项目嵌入向量$\boldsymbol{e}_{\hat{i}}$和项目偏好矩阵$\boldsymbol{H}$；

输出：推荐项目列表$List$。

① 初始化生成器G_θ，判别器D_ϕ，项目偏好矩阵$\boldsymbol{H}$，生成器与判别器的平衡阈值ε，迭代阈值ξ；

② while $\left|\mathcal{L}_{G_\theta}-\mathcal{L}_{D_\phi}\right|\geqslant\varepsilon$或迭代次数小于$\xi$ do

③ for G_θ的每一轮训练 do

④ for 每个用户u do

⑤ $\boldsymbol{E}_{\text{joint},u}$输入ST-GS层得到$\hat{\boldsymbol{E}}_{\text{joint},u}$；/*见式

(8)*/
⑥ 在 $\boldsymbol{H}$ 中查询得到伪项目 $\boldsymbol{e}_i$; /*见式(9)*/
⑦ 计算生成器损失函数值; /*见式(10)*/
⑧ 将 $\boldsymbol{e}_i$ 输入判别器;
⑨ end for
⑩ end for
⑪ for D_ϕ 的每一轮训练 do
⑫ for 每个用户 u do
⑬ 计算判别器损失函数值鉴别 $\boldsymbol{e}_i$; /*见式(11)*/
⑭ end for
⑮ end for
⑯ 冻结参数 G_θ, 迭代更新 D_ϕ; /*见式(13)*/
⑰ 冻结参数 D_ϕ, 迭代更新 G_θ; /*见式(14)*/
⑱ end while
⑲ 输出推荐项目列表 *List*。

在算法 3 中, 行③~⑩由生成器产生 $\boldsymbol{e}_i$, 行⑪~⑮获得判别器损失函数值以鉴别 $\boldsymbol{e}_i$, 行⑯⑰是生成器和判别器的迭代优化。

2.6 方法讨论

所提模型 AFS-GAN 利用简洁的元路径高效构建显式和隐式社交特征, 减少了对特定应用场景的依赖, 提升了模型的实用性。它采用自适应融合机制在去除噪声的同时, 充分保留了社交关系中的有效信息, 提高了推荐的准确性和适用性。在生成器中利用直通 Gumbel Softmax 加速生成伪项目的效率, 通过四元 BPR 损失函数简化判别器结构, 整体提升了 GAN 的计算效率。相较于第 4 节中对比的三大类模型, AFS-GAN 更能挖掘和利用隐式社交关系, 在个性化推荐、数据稀疏性处理和计算成本优化方面展现出明显的优势。

1)与基于图神经网络的社交推荐模型不同的是, AFS-GAN 只采用了 2 种简单的元路径(UTU 和 UIU), 分别用来挖掘 1 阶显式和 2 阶隐式社交关系。这种设计避免了对图结构的多层次多迭代搜索, 使得模型计算更加高效, 减少了在特定应用场景中应用的难度, 也大大降低了模型在样本探索中的复杂性和主观偏差。

2)与基于异构图解耦的社交推荐模型不同的是, AFS-GAN 采用的社交特征自适应融合机制避免在异构图多次解耦和重新组合的过程中引入过多的噪声, 尽可能保留了丰富的社交特征信息。这一特点提升了模型的适应性, 使其能够更好地应对社交网络中的信息稀疏和噪声问题。

3)与基于 GAN 的社交推荐模型不同的是, AFS-GAN 利用直通 Gumbel Softmax 优化伪项目的生成效率, 并通过四元 BPR 损失函数直接最大化判别损失。该方法简化了判别器的结构, 加快了对抗训练的收敛速度, 同时提高了推荐的准确性和效率。

在第 3 节中, 我们继续通过丰富的数据集、严谨的实验设计, 从多个角度论述 AFS-GAN 在推荐上的优越性。在第 4 节中, 通过时间复杂度的比较, 再一次阐明 AFS-GAN 是简洁易用的推荐方法, 具有良好的适用性。

3 实验结果与分析

为评估 AFS-GAN 模型的性能, 我们通过多个综合实验来分析下面 4 个问题。

问题 1. 相较于经典的、最新的对比模型, AFS-GAN 能否提升社交推荐模型的性能。

针对此问题, 我们在 3.2 节通过对比实验做了详细分析。实验结果表明, 在 4 个数据集上与 8 种对比模型相比, 在精确率(*Precision*)、召回率(*Recall*)、归一化折损累计增益(*NDCG*)指标上 AFS-GAN 表现卓越。

问题 2. AFS-GAN 模型的不同组件对其性能有何影响, 是否有存在的必要性。

针对此问题, 我们在 3.3 节开展了消融实验, 重点研究了 AFS-GAN 模型中的 1 阶显式社交特征、2 阶隐式社交特征、自编码器、自适应因子对模型性能的影响, 涉及 4 种变体模型。实验结果表明, 变体模型的推荐效果或效率均在不同程度上弱于 AFS-GAN 模型, 这表明各个组件是有效的、必要的。

问题 3. AFS-GAN 模型中用户社交特征的自适应融合方法是否有效。

针对此问题, 我们在 3.4 节通过选择不同水平的自适应因子或不同的自适应融合方法来研究各种融合背景下 AFS-GAN 模型的性能表现。实验结果表明, 本文提出的用户社交特征的自适应方法更加有效。

问题 4. 如何为 AFS-GAN 模型中的关键组件选择恰当的超参数。

针对此问题, 我们在 3.5 节设置了超参数的敏感度实验, 选择 ST-GS 层的温度 τ、伪项目生成器的正则化系数 λ_2 和伪项目判别器的正则化系数 λ_3 进行研究, 为 AFS-GAN 模型的关键组件确定恰当的超参数。

3.1 实验设置

3.1.1 数据集

本文采用4个常用的社交推荐数据集LastFM[①], FilmTrust[②], Yelp[③], Epinions[④]评估AFS-GAN模型的性能，如表2所示。在这些数据集中，80%的数据用于训练，10%用于验证，10%用于测试。

Table 2 Statistics of Four Datasets

表2 4个数据集的统计信息

统计项目	LastFM	FilmTrust	Yelp	Epinions
用户数	1 892	1 508	19 539	40 163
项目数	17 632	2 071	22 228	139 738
交互数	92 834	35 497	450 884	664 824
社交关系数	25 434	1 853	864 157	487 183
交互密度/%	0.278	1.137	0.104	0.012
社交密度/%	0.711	0.081	0.226	0.030

1）LastFM数据集包含用户的音乐欣赏行为、艺术家偏好及社交网络信息，特别关注如何利用用户的社交关系和音乐偏好来改进社交推荐效果。

2）FilmTrust数据集包含来自FilmTrust网站的用户电影评分和社交网络信息，主要用于研究信任关系如何影响个人的评分行为。

3）Yelp数据集由Yelp网站提供，囊括了用户对餐厅、商铺和本地服务的评价、评分以及商家信息，还包含用户间的友谊链接和用户个人资料，主要用于研究如何通过用户评价和社交网络向用户推荐商家。

4）Epinions数据集来自于Epinions网站，涵盖了用户之间的社交信任网络和对各种产品的评价信息，广泛用于分析用户如何基于他人意见形成自己的购买决策和产品评价。

3.1.2 评价指标

本文采用3个指标进行评估，包括基于相关性的指标*Precision*和*Recall*，还有基于排序的指标*NDCG*。

*Precision*衡量用户实际交互项目占top-N推荐项目的比例，如式(15)所示。

$$Precision=\frac{|S_{\mathrm{r}}\cap S_{\mathrm{t}}|}{|S_{\mathrm{r}}|}, \tag{15}$$

其中S_{r}是推荐项目集，S_{t}是用户实际交互的项目集。*Precision*可反映AFS-GAN生成的top-N推荐列表是否更符合用户偏好。

*Recall*衡量top-N推荐项目占用户实际交互项目的比例，可反映AFS-GAN识别用户偏好以及推荐结果是否更好地覆盖用户兴趣的能力，如式(16)所示。

$$Recall=\frac{|S_{\mathrm{r}}\cap S_{\mathrm{t}}|}{|S_{\mathrm{t}}|}。 \tag{16}$$

*NDCG*是一种用于评估排序结果的指标，是对折损累计增益(*DCG*)的改良。由于不同用户的列表长度可能不同，直接比较不同用户的*DCG*会不准确。因此，通过归一化处理，*NDCG*能够更公平地比较不同的用户。该指标可反映AFS-GAN生成的推荐项目排序列表是否具有更高的质量，如式(17)所示。

$$NDCG=\sum_{u\in U_{\mathrm{test}}}\frac{1}{Y_u}\sum_{i=1}^{k}\frac{2^{t_i}-1}{\mathrm{lb}(i+1)}, \tag{17}$$

其中U_{test}表示测试用户集，Y_u表示用户u的最大*NDCG*值；$t_i=1$表示预测项目在列表中的位置与实际情况一致，$t_i=0$，反之。

3.1.3 对比模型

为全面评估和展示AFS-GAN在处理社交推荐任务上的有效性和优势，本文选择了8个较新的、先进的模型作对比。

1）基于图神经网络的社交推荐模型ESRF[26], GNNRec[29], GBSR[30]；

2）基于异构图解耦的社交推荐模型CDRSB[47], DGNN[48]；

3）基于GAN的社交推荐模型DASO[15], GANMF[16], DAEGAN[49]。详细信息如表3所示。

3.1.4 超参数与实验环境

通过网格搜索的方式，选择AFS-GAN中使用到的常规超参数。其中，嵌入向量维度为64，每个训练批量大小为1 024，总训练周期为100，初始学习率为0.001，每10个周期衰减0.1。正则化系数λ_1为0.01。ST-GS的温度τ，以及生成器正则化系数λ_2和判别器正则化系数λ_3作为主要超参数，在3.5节会展开进一步分析。

实验环境为操作系统ubuntu22.04，深度学习框架PyTorch2.1.0，编程语言Python3.10，硬件版本Cuda 12.1，GPU RTX 4090(24 GB)，CPU 16vCPU Intel® Xeon® Platinum 8352V CPU @ 2.10 GHz。

① https://guoguibing.github.io/librec/datasets.html

② http://files.grouplens.org/datasets/hetrec2011

③ https://www.dropbox.com/sh/

④ http://www.trustlet.org/wiki/Epinions_dataset

Table 3 Comparative Analysis of Eight Baseline Models

表 3 8 个基线模型的对比分析

模型类型	对比模型	模型特点
基于图神经网络的社交推荐模型	ESRF	通过对抗图卷积网络生成并优化隐式社交邻居，结合注意力机制动态调整关系权重，缓解社交稀疏性
	GNNRec	利用门控图神经网络处理用户会话信息，并结合图注意力网络来聚合社交网络上的用户和其朋友的社交信息，用于模拟用户兴趣
	GBSR	通过对社交网络去噪来调整用户偏好，过滤掉不必要的社交关系，同时保留对准确推荐至关重要的关系
基于异构图解耦的社交推荐模型	CDRSB	通过解耦编码器将用户和项目嵌入向量分解为兴趣和社交影响 2 部分，再通过调节解码器动态学习社交影响的权重
	DGNN	解耦社交网络中用户和项目间的多种关系，通过消息传递机制来处理社交关系和用户项目交互，提高用户和项目的特征整合能力
基于 GAN 的社交推荐模型	DASO	利用对抗学习引入一种双向映射方法，在对偶域（项目域和社交域）中学习用户表征，并将其从社交域映射到项目域
	GANMF	采用特征匹配损失提升推荐项目质量，判别器采用自编码器重构评分矩阵，生成器产生自编码器无法与真实评分矩阵区分的伪评分矩阵
	DAEGAN	采用去噪自编码器作为生成器产生用户-项目交互向量，再由 MLP 构成的判别器区分真实和虚假的交互向量

3.2 对比实验分析（问题 1）

我们对每个模型分别在 3 个水平，即 top-10, top-15, top-20, 进行 25 次实验，采用单因素方差分析方法检验实验结果的置信度，对 3 组实验结果进行统计显著性检验，便于更细致地评估模型性能，增强实验结果的可靠性，如表 4 所示。

由表 4 可知，AFS-GAN 模型绝大部分实验结果领先对比模型，在不同的业务背景、规模、交互密度和社交密度的数据集中均表现出良好的实用性和适用性，下面是详细分析。

1）AFS-GAN 在较小规模数据集 LastFM 和 FilmTrust 上的性能表现显著。在规模较小且信息交互和社交关系密度相对较高的数据集上，AFS-GAN 表现出优异的性能。对于 *Precision*@15, AFS-GAN 在 2 个数据集上分别提升了 1.61 个百分点和 1.37 个百分点，在其他指标和层次上也可看到类似的优越趋势。这可归因于以下因素：①AFS-GAN 模型通过 UTU 和 UIU 这 2 种简洁的元路径有效地提取了 1 阶显式社交特征和 2 阶隐式社交特征。这种方法直接关联了用户的直接社交环境和通过共同交互项目间接连接的社交网络，从而提高了模型捕获和理解用户社交行为的能力。在社交关系和交互密集的小规模数据集上，这种精确的社交特征映射尤其有效，因为社交网络的结构和用户的行为模式在这类数据集中更易于解析和利用。②AFS-GAN 采用自编码器来自适应地融合显式和隐式社交特征，通过学习最佳的低维表示来强化用户的综合社交特征。这种自适应融合机制不仅确保了融合过程中显式与隐式特征的均衡，还优化了特征的综合效用，使得生成的用户社交特征更全面、更能反映用户的实际社交偏好。

2）在较大规模数据集 Yelp 上的性能表现突出。尽管 Yelp 的数据规模很大，但其社交和交互信息的稀疏性并不是很高。从表 4 可看出，当排序序列长度增加时，AFS-GAN 在 Yelp 上的性能变化平稳，随着推荐项目序列长度的增加，*Precision* 和 *Recall* 都有所提高，*Precision*@15 较最佳的对比模型 GBSR 提升了 0.94 个百分点；*Recall*@15 较最佳的 CDRSB 提升了 1.28 个百分点；但 *NDCG*@15 没有提升，相较于最佳的 DASO 下降了 3.61 个百分点。这可归因于以下因素：①在规模大且社交密度较低的数据集中，用户社交特征自适应融合机制仍可捕捉到丰富的用户社交信息。②随着推荐项目序列长度的增加，目标用户可利用的 1 阶显式社交关系在减少，但 2 阶隐式社交关系并未受到太多影响，从而使得融合后的用户社交特征依然保持了丰富的用户信息，却未引入过多的噪声。而 GBSR 在移除不相关的社交连接过程中不可避免地过滤掉有价值的信息，导致重要的社交特征丢失。CDRSB 期望对社交网络进行解耦，同样也无法避免损失有价值的社交特征。DASO 通过双 GANs 生成的特征再经过 MLP 的非线性融合可精准捕捉特定序列长度下的用户真实偏好，这使得该模型根据 *NDCG*@15 指标衡量在项目列表顶部推荐质量时表现比 AFS-GAN 更有优势。

3）在数据稀疏度最高的大规模数据集 Epinions 上的性能优势依然明显。随着排序序列长度的增加，AFS-GAN 在 Epinions 数据集上的指标值与其他数据集相比有所波动，*Precision*@15 相较于 DASO 下降了 2.80 个百分点，*NDCG*@20 相较于 DASO 下降了 3.80 个百分点，但总体上还是好于各对比模型，表现出了良好的适应性。这可归因于以下因素：①AFS-GAN

Table 4 Performance Comparison Between AFS-GAN and Baselines

表 4 AFS-GAN 与基线模型性能比较

数据集	指标	ESRF	GNNRec	GBSR	CDRSB	DGNN	DASO	GANMF	DAEGAN	AFS-GAN	性能变化/%
LastFM	*Precision*@10	0.108 0	0.112 7	0.111 7	0.111 2	0.118 6	0.119 2	0.114 4	0.1193	**0.120 8**	1.26
	Recall@10	0.1509	0.143 3	0.147 1	0.150 2	0.142 9	0.142 6	0.147 8	0.143 5	**0.151 2**	0.20
	NDCG@10	0.152 9	0.153 9	0.155 1	0.151 7	0.135 4	0.135 1	0.1563	0.135 6	**0.160 7**	2.82
	Precision@15	0.155 7	0.155 3	0.156 4	0.1586	0.103 9	0.093 5	0.098 6	0.104 5	**0.161 1**	1.58
	Recall@15	0.166 1	0.169 6	0.163 6	0.168 2	0.143 0	0.142 9	0.1698	0.143 8	**0.171 6**	1.06
	NDCG@15	0.186 7	0.205 4	0.205 6	0.208 2	0.137 2	0.137 4	0.2083	0.137 9	**0.211 7**	1.63
	Precision@20	0.092 1	0.092 9	0.0998	0.094 2	0.088 1	0.088 8	0.094 4	0.088 8	**0.100 8**	1.00
	Recall@20	0.191 9	0.201 0	0.2113	0.204 6	0.144 7	0.144 4	0.175 9	0.145 3	**0.210 9**	−0.19
	NDCG@20	0.207 4	0.218 1	0.210 9	0.2183	0.140 9	0.141 2	0.213 4	0.141 5	**0.220 7**	1.10
FilmTrust	*Precision*@10	0.159 7	0.161 9	0.161 4	0.166 2	0.125 1	0.125 3	0.1674	0.125 9	**0.171 0**	2.15
	Recall@10	0.172 8	0.1809	0.174 1	0.176 4	0.149 5	0.149 2	0.176 5	0.150 0	**0.181 2**	0.17
	NDCG@10	0.190 8	0.2097	0.204 8	0.207 5	0.209 2	0.208 6	0.141 3	0.209 4	**0.211 6**	0.91
	Precision@15	0.1678	0.160 1	0.167 5	0.164 2	0.159 2	0.098 9	0.161 5	0.159 8	**0.169 8**	1.19
	Recall@15	0.177 5	0.180 7	0.1809	0.174 6	0.173 3	0.175 1	0.148 2	0.174 3	**0.181 4**	0.28
	NDCG@15	0.203 9	0.211 5	0.2231	0.219 8	0.142 6	0.142 6	0.213 4	0.143 5	**0.221 0**	−0.94
	Precision@20	0.150 6	0.154 3	0.154 4	0.1601	0.156 6	0.156 5	0.094 7	0.156 8	**0.160 7**	0.37
	Recall@20	0.189 0	0.1894	0.183 4	0.181 1	0.150 8	0.151 0	0.182 5	0.151 6	**0.191 1**	0.90
	NDCG@20	0.201 6	0.215 1	0.212 0	0.215 2	0.146 3	0.146 1	0.2202	0.147 0	**0.218 8**	−0.64
Yelp	*Precision*@10	0.155 1	0.155 5	0.154 1	0.151 2	0.155 9	0.155 5	0.140 4	0.1562	**0.158 6**	1.54
	Recall@10	0.164 8	0.169 9	0.168 0	0.1703	0.124 6	0.124 1	0.168 4	0.124 6	**0.171 6**	0.76
	NDCG@10	0.186 2	0.198 2	0.199 6	0.2012	0.136 7	0.136 6	0.124 2	0.136 9	**0.202 0**	0.40
	Precision@15	0.154 1	0.148 8	0.154 3	0.1552	0.129 4	0.129 6	0.150 7	0.129 9	**0.156 7**	0.97
	Recall@15	0.1794	0.171 7	0.172 4	0.178 0	0.175 1	0.175 4	0.099 1	0.175 7	**0.180 3**	0.50
	NDCG@15	0.192 1	0.197 4	0.193 1	0.193 6	0.186 8	0.2104	0.137 4	0.187 7	**0.202 8**	−3.61
	Precision@20	0.138 7	0.137 4	0.140 4	0.136 9	0.141 5	0.131 5	0.1448	0.142 0	**0.146 4**	1.10
	Recall@20	0.178 8	0.181 4	0.181 1	0.180 9	0.184 6	0.095 1	0.182 1	0.1853	**0.186 4**	0.59
	NDCG@20	0.195 3	0.204 7	0.204 7	0.2118	0.138 3	0.208 6	0.139 1	0.139 1	**0.212 4**	0.28
Epinions	*Precision*@10	0.140 0	0.146 2	0.145 4	0.149 1	0.149 9	0.014 6	0.1503	0.150 2	**0.151 9**	1.06
	Recall@10	0.135 3	0.168 2	0.1741	0.165 6	0.128 5	0.128 9	0.173 2	0.129 1	**0.174 3**	0.11
	NDCG@10	0.184 1	0.1967	0.196 0	0.195 2	0.121 8	0.022 7	0.194 2	0.122 8	**0.201 7**	2.54
	Precision@15	0.142 6	0.156 4	0.150 6	0.152 1	0.155 9	0.1610	0.124 7	0.156 7	**0.156 5**	−2.80
	Recall@15	0.185 0	0.186 5	0.188 2	0.186 1	0.175 6	0.105 4	0.1887	0.175 8	**0.192 5**	2.01
	NDCG@15	0.153 9	0.196 3	0.197 5	0.1983	0.192 8	0.192 9	0.132 4	0.193 4	**0.198 5**	0.09
	Precision@20	0.143 6	0.1484	0.144 9	0.142 0	0.125 7	0.125 6	0.147 1	0.126 1	**0.150 1**	1.15
	Recall@20	0.188 5	0.193 3	0.1968	0.193 9	0.182 0	0.101 6	0.189 9	0.182 0	**0.196 9**	0.05
	NDCG@20	0.204 0	0.200 9	0.197 6	0.199 5	0.202 6	0.2104	0.132 9	0.202 9	**0.202 4**	−3.80

注：实验结果具有统计显著性，即 $p<0.05$。AFS-GAN 的指标值用黑体数值标识，对比模型的最优值用下划线数值标识，“性能变化”列中的数值表示 AFS-GAN 相较于对比模型中最优结果的变化幅度。例如，就 LastFM 的 *Precision*@10 指标，在对比模型中，DAEGAN 表现最优，AFS-GAN 相较于它提升了 1.26%。

有效利用简洁的元路径提取 1 阶显式和 2 阶隐式社交关系，这提供了不同的用户信息，从而即使在用户与项目交互较少的情况下，也能通过自适应融合的社交特征推断出用户潜在偏好。②自适应融合机制可根据每个用户的具体情况自动调整显式和隐式特征的权重，从而在稀疏的环境中提升推荐的准确性

和个性化。③在数据稀疏的环境中, AFS-GAN 通过对抗学习持续优化生成的用户特征, 能够生成更贴近真实用户偏好的伪项目, 从而改善训练效果以及模型实用性和适应性。

对问题 1 的回答:

在绝大多数指标中, AFS-GAN 模型优于对比模型, 说明其在社交特征自适应融合机制的支持下能有效地提取和融合 4 种各具代表性数据集中的 1 阶显式及 2 阶隐式社交特征, 并通过 GAN 增强对稀疏数据的处理能力, 使其在处理复杂和多样化数据时具有较明显的实用性和适用性。

在规模更大、更稀疏的数据集中, AFS-GAN 模型在个别排序指标上相较于对比模型 DASO 稍弱, 主要原因在于处理稀疏社交数据和捕获隐式社交关系的方式有所不同。AFS-GAN 模型通过自适应融合显式与隐式社交特征, 并利用对抗训练生成伪项目, 以挖掘用户潜在偏好。然而, 由于训练迭代次数的限制, 伪项目的生成和噪声过滤机制尚未达到最佳状态。相比而言, 模型 DASO 通过双向映射策略以及生成更具挑战性的“难负样本”, 有效地捕获了隐式社交信号, 在有限的迭代次数内实现了较好的对抗判别能力。然而需强调的是, 这种微小差距并非普遍现象, 仅在个别对比模型和特定实验环境下存在。同时, 尽管在个别排序指标上稍差, AFS-GAN 在反映推荐结果准确性的 *Precision* 和 *Recall* 指标上仍表现出显著优势。

3.3 消融实验分析(问题 2)

这里对比 AFS-GAN 模型与其变体模型在 3 个指标上的表现来分析不同组件对于推荐质量和效率的影响。根据是否纳入隐式或显式社交特征以及纳入的方式, 我们将 AFS-GAN 改造成 4 种变体:

1)变体模型 AFS-GAN_1 剔除隐式社交特征, 保留显式社交特征。即, 模型只用 $\boldsymbol{E}_{\mathrm{UTU},u}$ 生成 $\boldsymbol{E}_{\mathrm{joint},u}$, 不再融合 $\boldsymbol{E}_{\mathrm{UIU},u}$。

2)变体模型 AFS-GAN_2 剔除显式社交特征, 保留隐式社交特征。即, 模型只用 $\boldsymbol{E}_{\mathrm{UIU},u}$ 生成 $\boldsymbol{E}_{\mathrm{joint},u}$, 不再融合 $\boldsymbol{E}_{\mathrm{UTU},u}$。

3)变体模型 AFS-GAN_3 不用自编码器重构 $\boldsymbol{E}_{\mathrm{UTU},u}$ 和 $\boldsymbol{E}_{\mathrm{UIU},u}$。即, 用 $\boldsymbol{E}_{\mathrm{UTU},u}$ 和 $\boldsymbol{E}_{\mathrm{UIU},u}$ 替换掉式(7)中的 $\hat{\boldsymbol{E}}_{\mathrm{UTU},u}$ 和 $\hat{\boldsymbol{E}}_{\mathrm{UIU},u}$。

4)变体模型 AFS-GAN_4 剔除了自适应因子 α_u。即, 式(7)变更为 $\boldsymbol{E}_{\mathrm{joint},u}=\hat{\boldsymbol{E}}_{\mathrm{UTU},u}\odot\hat{\boldsymbol{E}}_{\mathrm{UIU},u}$。

表 5 是以上 4 种变体模型的组件构成情况。

Table 5 Component Composition of AFS-GAN Variant Models

表 5 AFS-GAN 的变体模型组件构成

模型	$\boldsymbol{E}_{\mathrm{UTU},u}$	$\boldsymbol{E}_{\mathrm{UIU},u}$	自编码器	自适应因子 α_u
AFS-GAN_1	×	√	√	√
AFS-GAN_2	√	×	√	√
AFS-GAN_3	√	√	×	√
AFS-GAN_4	√	√	√	×
AFS-GAN	√	√	√	√

注: “×”表示变体模型中剔除该组件, “√”表示变体模型中包含该组件。

由 3.2 节的对比可知, AFS-GAN 在 top-15 时性能表现均衡, 故这里选择 *Precision*@15, *Recall*@15, *NDCG*@15 作为对比指标(后续各项实验均采用这 3 个指标, 不再赘述), 如图 3 所示。

从图 3 中可看出, 各种变体模型的推荐指标均低于 AFS-GAN 模型, 尤其是未采用融合方法的变体模型 AFS-GAN_1 和 AFS-GAN_2 的结果最差, 采用了融合方法的 AFS-GAN_3 和 AFS-GAN_4 的推荐指标有所提升, 但在不同的数据集上有不同的表现。

1)融合用户 1 阶显式和 2 阶隐式社交特征是提升推荐指标的重要基础。在面对具有较丰富的社交和交互信息的数据集时, 如 LastFM 和 Yelp 这样的社交媒体和评价平台中的用户互动频繁, 显式的社交关系能够提供强有力的推荐信息。AFS-GAN_1 的推荐指标在 4 个变体模型中相对 FilmTrust 和 Epinions 数据集较好, *Precision*@15 分别为 0.114 5, 0.106 7, 0.108 9, 0.094 5。

这是因为显式的社交数据通常更结构化、更清晰, 处理起来相对简单, 易于模型理解和利用, 能够有效减少噪声和不相关信息的干扰。

用户显式社交不够显著的较大规模稀疏数据集, 如 FilmTrust 和 Epinions 通常包含更广泛的用户行为和更复杂的交互类型, 单一的显式社交特征可能不足以完全捕获用户间的隐式联系, 而 AFS-GAN_2 在其他 2 个数据集表现较好, *Precision*@15 分别为 0.099 8, 0.100 4, 0.081 7, 0.093 8。这是因为: ①隐式社交特征能够利用用户间的间接交互(如共同关注的项目或相似的评价行为)来推断潜在的社交联系和相似偏好, 这在显式联系不明显的环境中尤其有价值。②隐式特征能够揭示用户不易观察到的深层次行为模式, 比如通过用户对相同或相似项目的评分和浏览来推测其潜在的兴趣和偏好。在数据稀疏的情况下, 这种方法可有效加深模型对用户行为的理解, 从而提高推荐系统的准确性和个性化程度。

2)采用恰当的方式融合用户显式和隐式社交特

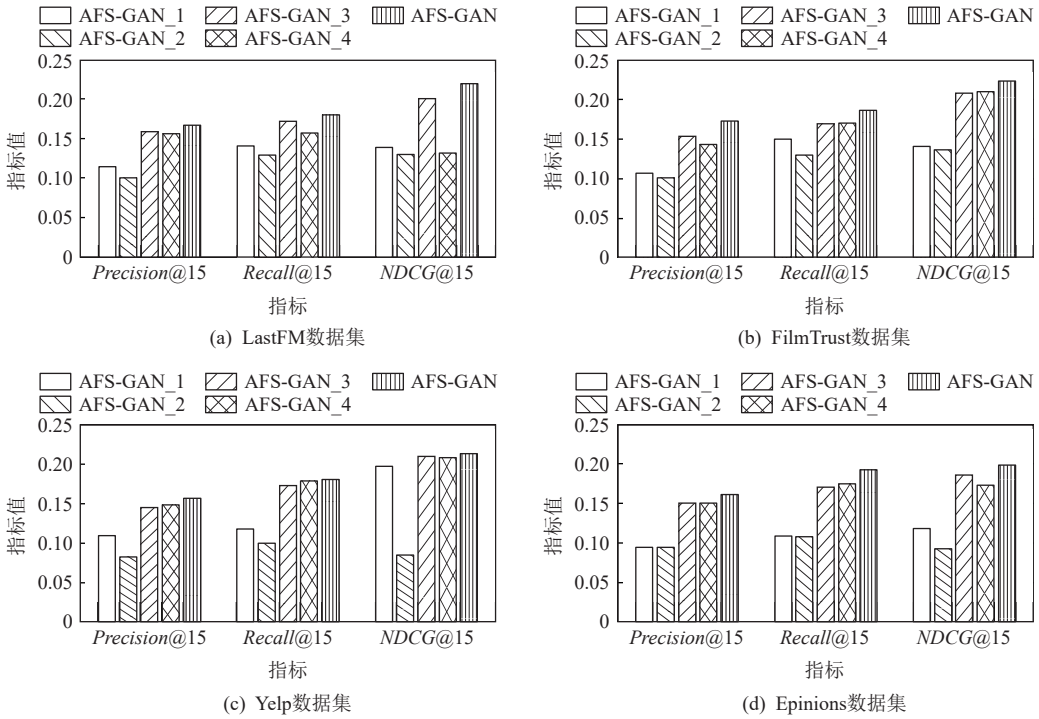


Fig. 3 The ablation experiment

图 3 消融实验

征是影响推荐指标的关键因素。变体模型 AFS-GAN_3 在对应 4 个数据集中的 *Precision*@15 分别为 0.159 2, 0.153 4, 0.144 5, 0.149 9, 在 4 个变体模型中, 其性能相对 AFS-GAN_1 和 AFS-GAN_2 均有较大提升, 与 AFS-GAN_4 大体持平, 但弱于 AFS-GAN。这是因为: ①AFS-GAN_3 直接用 $\boldsymbol{E}_{\mathrm{UTU},u}$ 和 $\boldsymbol{E}_{\mathrm{UIU},u}$ 融合得到 $\boldsymbol{E}_{\mathrm{joint},u}$, 未经过自编码器重构的 1 阶显式和 2 阶隐式社交特征含有的噪声, 这会降低融合特征的有效性和质量。②采用自编码器能压缩和精炼这些特征, 剔除多余的噪声, 通过最小化重建误差来捕捉最有价值的信息, 使得 $\boldsymbol{E}_{\mathrm{joint},u}$ 的分布更加有序, 异常值更少(这一点在 3.4.3 节中有进一步说明)。③自编码器的结构允许模型根据数据自适应调整参数, 以学习最佳的特征表示。剔除自编码器后, 模型失去了自我调整的能力。由此, AFS-GAN_3 可充分说明自编码器在用户社交特征融合中的关键作用。

AFS-GAN_4 的 *Precision*@15 分别为 0.156 2, 0.142 8, 0.148 7, 0.150 3, 在 LastFM 和 Yelp 这 2 个社交密度相对较高的数据集上, 其结果相较于 AFS-GAN_1 和 AFS-GAN_2 均有所提升; 在 FilmTrust 和 Epinions 数据集上, 其结果不及 AFS-GAN_3。出现这一结果的原因有以下 2 点。①AFS-GAN_4 未引入自适应因子 α_u 使得推荐丧失了自适应性, 意味着模型无法根据每个用户的实际社交环境自适应优化特征融合, 导致在不同用户和环境中的推荐效果出现差异(这一点在图 4 中有所表现)。②正是因为自适应性的缺乏, 使得 AFS-GAN_4 在面对不同数据集时不能根据用户的具体情况融合 1 阶显式和 2 阶隐式社交特征, 粗糙的融合会平滑或抹去 2 个特征间的对比度, 减少特征间的差异性(这一点在 3.4.2 节中有进一步说明), 从而影响推荐系统的表现。

3)融合用户 1 阶显式和 2 阶隐式社交特征是提升推荐指标的重要基础。在面对具有较丰富的社交和交互信息的数据集时, 如 LastFM 和 Yelp 这样的社交媒体和评价平台, 用户互动频繁, 显式的社交关系能够提供强有力的推荐信息。在 4 个变体模型中 AFS-GAN_1 的推荐指标在 FilmTrust 和 Epinions 数据集中的表现较好, *Precision*@15 分别为 0.114 5, 0.106 7, 0.108 9, 0.094 5。这是因为显式的社交数据通常更结构化、更清晰, 处理起来相对简单, 易于模型理解和利用, 能够有效减少噪声和不相关信息的干扰。

用户显式社交不够显著的较大规模稀疏数据集, 如 FilmTrust 和 Epinions 通常包含更广泛的用户行为和更复杂的交互类型, 单一的显式社交特征可能不足以完全捕获用户间的隐式联系, 而 AFS-GAN_2 在 LastFM 和 Yelp 数据集中表现较好, *Precision*@15 分别为 0.099 8, 0.100 4, 0.081 7, 0.093 8。这是因为: ①隐式社交特征能够利用用户间的间接交互(如共

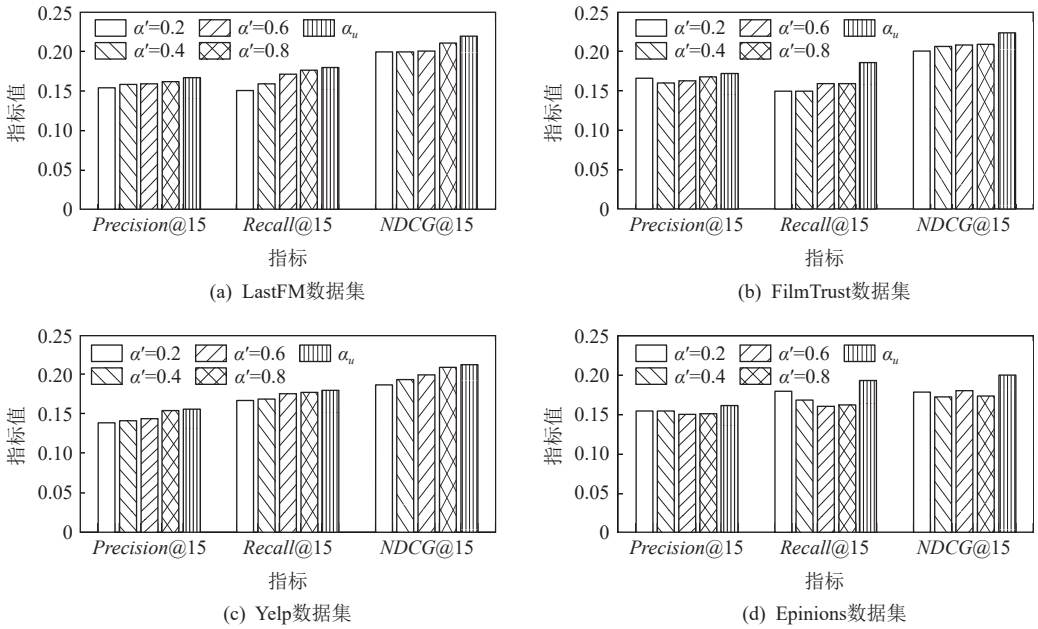


Fig. 4 The extensibility experiment of α_u

图 4 α_u可扩展性实验

同关注的项目或相似的评价行为)来推断潜在的社交联系和相似偏好，这在显式联系不明显的环境中尤其有价值。②隐式特征能够揭示用户不易观察到的深层次行为模式，比如通过用户对相同或相似项目的评分和浏览来推测其潜在的兴趣和偏好。在数据稀疏的情况下，这种方法可有效加深模型对用户行为的理解，从而提高推荐系统的准确性和个性化程度。

对问题 2 的回答：

1 阶显式社交特征(如互加好友、关注、点赞等)通常提供直接的社交联系信息，有助于捕获用户间明确的相似性和联系。2 阶隐式社交特征(如相同兴趣或行为模式)能够揭示用户间的间接联系，这在数据稀疏或社交联系不显著的环境中尤为重要。AFS-GAN 模型通过自编码器和自适应因子精细地融合显式、隐式 2 组特征，不仅考虑了用户的直接社交连接，也兼顾了通过共同兴趣或活动连接的用户。这些组件的协同显著提升了模型性能，组件的设计目的明确，其存在是必要的。

3.4 用户社交特征自适应融合方法适应性分析(问题 3)

这里采用的用户社交特征自适应融合方法有 2 个主要特点。第一，每个用户 u 拥有独特的与其显式、隐式社交关系密切相关的自适应因子 α_u，它不是一个全局的超参数，而是一个个性化的参数。在式(7)中，α_u为重构后的用户显式和隐式社交特征赋予了不同的权重。第二，加权后的重构社交特征通过哈达玛积将特征向量逐元素相乘，使得融合后的 $\boldsymbol{E}_{\text{joint},u}$综合反映了用户的个性化社交行为。针对以上 2 个特点，下面通过 2 组实验分析融合方法的适应性。

3.4.1 采用α_u融合社交特征提升模型的可扩展性

我们采用不同水平的常数α'替换自适应因子α_u，$\alpha' \in \{0.2, 0.4, 0.6, 0.8\}$，观察不同水平$\alpha'$与$\alpha_u$在推荐效果上的变化情况，如图 4 所示。随着$\alpha'$值的增加，推荐指标在 4 个数据集上均有不同程度的变化。

1)随着α'由小到大变化，LastFM 和 Yelp 数据集的各项指标均有提升。其中，*Precision*@15 分别从 0.154 5, 0.138 9 提高到 0.162 2, 0.154 7。

与此同时，采用自适应因子α_u得到的指标值分别为 0.167 1, 0.156 7，比α'更佳。从表 2 可知，这 2 个数据集的社交密度较高，且是交互密度的 2 倍以上，表明用户社交行为的频繁程度要高于项目交互行为。此时，社交关系对用户的推荐结果影响更大。因此随着α'逐步增大，模型的推荐质量也随之提升，但依然没有达到采用自适应因子α_u的推荐水平。从实验结果可知，尽管在社交相对丰富的应用场景中提升显式社交特征的融合权重可以优化推荐质量，却忽视单一用户的个性化偏好，也会影响模型性能，导致推荐结果次优。

2)随着α'由小到大变化，Epinions 数据集的各项指标有所下降。其中，*Recall*@15 下降比较明显，从

0.178 9 下降到了 0.162 3.与之相对应地，采用自适应因子α_u的 *Recall*@15 为 0.192 5，仍比α'更佳。从表 2 可知，尽管该数据集的社交密度是交互密度的 3 倍以上，但其社交密度却是 4 个数据集中最低的，而且相当稀疏，只有 0.030%，这说明过于依赖显式社交特征不足以产生高质量的推荐结果，甚至随着α'的增加，还反而会导致推荐质量下降。

3）随着α'由小到大变化，FilmTrust 数据集的各项指标变化平缓，尤其是 *NDCG*@15 的值仅从 0.200 8 增长到 0.210 1，变化不明显。此时，以自适应因子α_u为融合权重得到的指标值是 0.224 0，依旧比α'更佳。从表 2 可知，该数据集的交互密度（1.137%）远高于社交密度（0.081%），这说明当用户与项目的交互数据足够丰富时，协同信息能提供较丰富的信息用于推荐。在这种情况下，单纯改变α'的值不会为提升推荐结果提供较大的帮助，而在自适应因子α_u的协助下才能充分挖掘每个用户的社交特征，为其提供高质量的个性化推荐。

3.4.2 采用哈达玛积融合社交特征增强模型普适性

用式（18）（19）分别替换式（7），用于比较不同融合方式所产生的$\boldsymbol{E}_{\text{joint},u}$对于推荐质量的影响。

$$\boldsymbol{E}_{\text{joint},u}=\left(\alpha_u\cdot\hat{\boldsymbol{E}}_{\text{UTU},u}\right)+\left((1-\alpha_u)\cdot\hat{\boldsymbol{E}}_{\text{UIU},u}\right)\text{，}\tag{18}$$

$$\boldsymbol{E}_{\text{joint},u}=concat\left(\alpha_u\cdot\hat{\boldsymbol{E}}_{\text{UTU},u},(1-\alpha_u)\cdot\hat{\boldsymbol{E}}_{\text{UIU},u}\right)\text{，}\tag{19}$$

其中式（18）表示 2 个特征向量逐元素相加，式（19）表示 2 个特征向量拼接成 1 个长向量。

从图 5 可看出，采用式（7）中哈达玛积的方式融合用户社交特征所得到的推荐效果最好。式（19）的拼接方式在 4 个数据集上的 *Precision*@15 分别为 0.166 2，0.169 2，0.155 7，0.160 3，与式（7）最接近。式（18）的逐元素相加方式的效果最差，在 4 个数据集上的 *Precision*@15 分别为 0.150 2，0.160 3，0.149 5，0.159 9。

我们对各数据集用户按照显式社交数量、交互项目数量、评分这 3 个维度做 *K*-means 聚类分析，发现用户的社交行为与交互行为之间有明显差异，可知此实验结果受到 3 方面因素的影响。

1）相较于其他 2 种特征融合方法，AFS-GAN 的哈达玛积特有的信息过滤特性能在不同特征之间存在较大差异时凸显特征中重要的元素，增强特征之间的交互。

图 6 中黄色圆点构成的簇表示用户社交行为频繁，而项目交互和评分不活跃，即$\boldsymbol{E}_{\text{UTU},u}$携带的信息量丰富；紫色圆点构成的簇表示用户交互行为频繁，但缺乏社交行为，即$\boldsymbol{E}_{\text{UIU},u}$能提供更多的信息。由此发现，在社交推荐的应用场景中，用户的显式和隐式社交特征携带不同信息，AFS-GAN 能够有效融合它们，帮助推荐模型有效捕捉每个用户的社交特征，从而提升推荐的准确性和个性化水平。

2）由于用户显式与隐式社交特征差异明显，采用式（18）的逐元素相加融合得到的社交特征会抵消 2 种特征之间的差异，使得特征间的差异性得以降低，而这种差异却是理解用户社交行为的关键。反之，AFS-GAN 通过哈达玛积的融合方式能够保留并利用这种差异，使得特征融合后的结果对于输入特征的

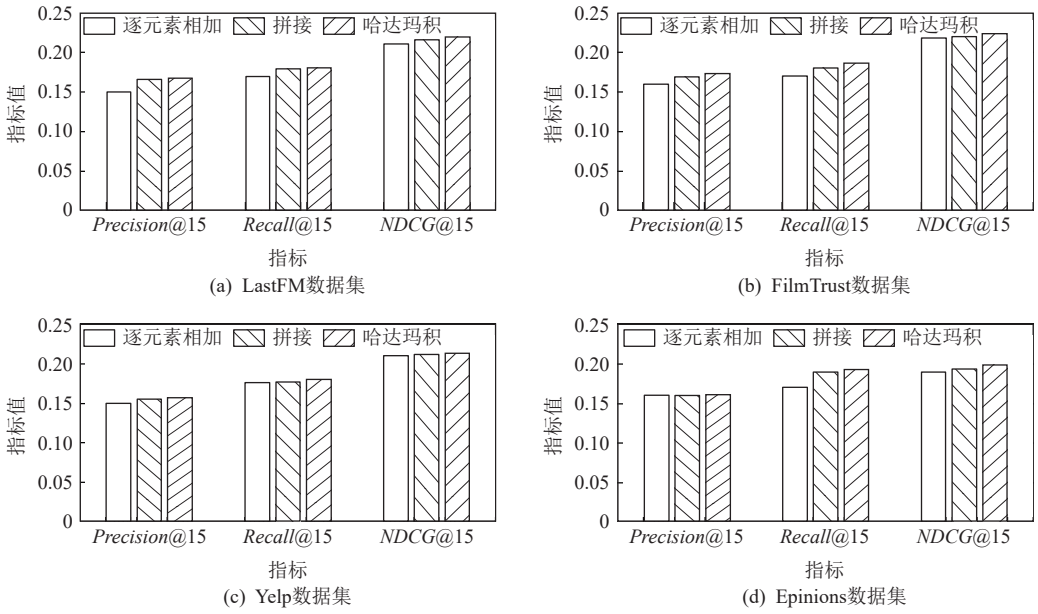


Fig. 5 The effectiveness experiments of Hadama product

图 5 哈达玛积有效性实验

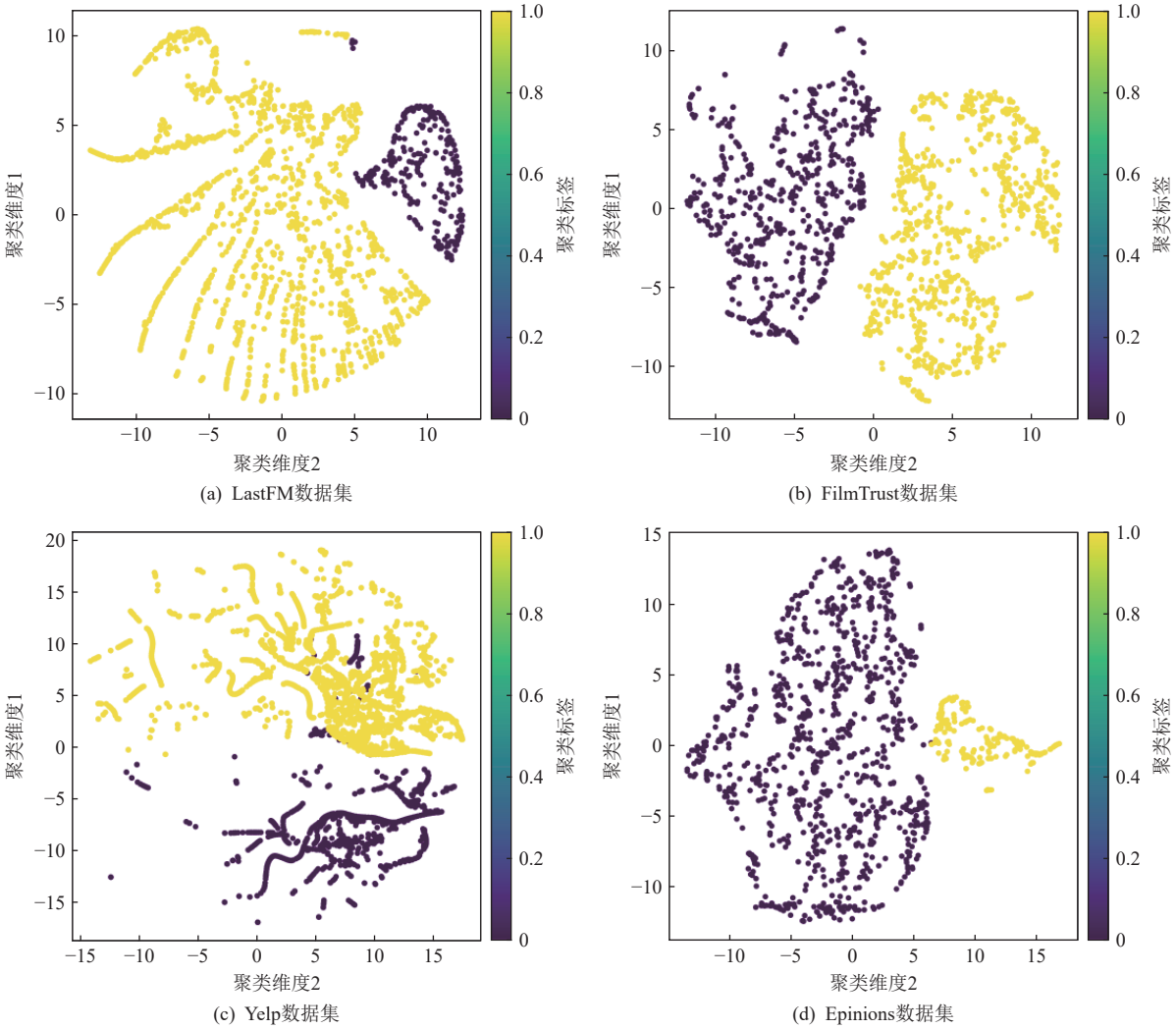


Fig. 6 Significant differences between user social behavior and interaction behavior

图 6 用户社交行为与交互行为间的显著差异性

微小变化更加敏感，这在动态环境下尤为重要，可使模型更灵敏地反映用户行为的微小变动，达到良好的个性化推荐效果。

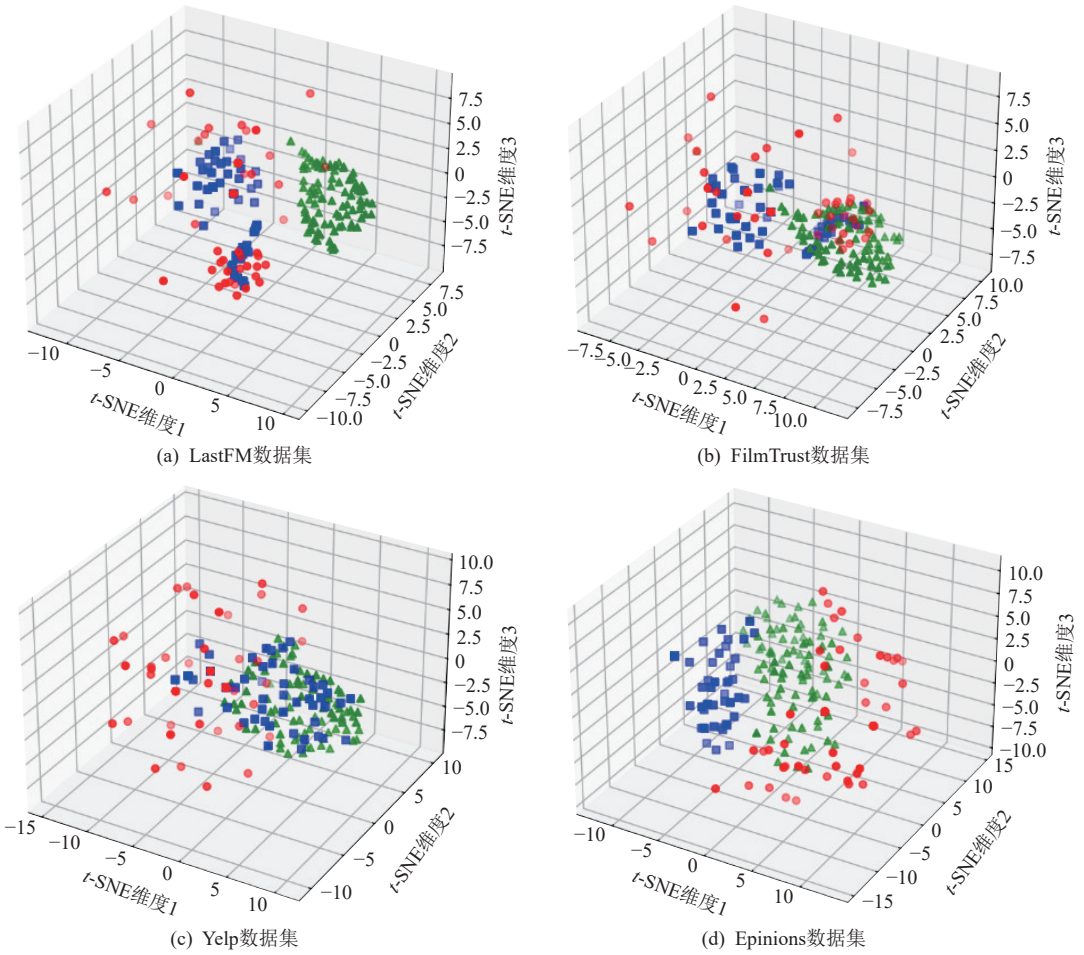
3)采用式(19)的拼接方式叠加并保留了原始特征的所有信息，没有任何信息压缩或丢失，由该方法得到的推荐效果与 AFS-GAN 的哈达玛积方式基本持平。但是，拼接方式处理信息时需要从更大的特征空间中学习，增加了模型的参数和复杂度。相反地，哈达玛积方式不增加特征向量的维度，并且能更有效整合与利用已有信息，突出 2 个特征向量中同时强烈的信号，而抑制不一致的信号，从而在特征融合时可更好地保留与目标相关性高的信息。这种方法可减少无关信息的干扰，增强模型在面对稀疏数据时的普适性。同时，由于哈达玛积不扩展特征空间的维度，模型的学习过程更加高效，缓解了因维数灾难带来的过拟合问题，从而表现出更好的推荐效果。

3.4.3 融合后的社交特征增强用户社交行为

为直观地说明自适应融合后的用户社交特征能充分反映用户的社交行为，我们从 4 个数据集中分别随机抽取 100 个用户，通过 3 维 *t*-SNE(*t*-distributed stochastic neighbor embedding)技术可视化了这些用户所对应的$\boldsymbol{E}_{\mathrm{UTU},u}$，$\boldsymbol{E}_{\mathrm{UIU},u}$，$\boldsymbol{E}_{\mathrm{joint},w}$，在 3 维空间中反映这 3 种嵌入向量的分布情况，如图 7 所示。

图 7 中不同颜色的点代表不同类型的特征集，其中红色圆点为显式特征，绿色三角为隐式特征，蓝色方块为融合特征。从图 7 中我们可观察到 2 个特点。

1)每种特征(显式、隐式、融合)都形成了明显的聚类。这表明每种特征类型均具有一定程度的内聚性，能反映不同用户社交模式的特定方面。而且，$\boldsymbol{E}_{\mathrm{joint},u}$在空间中与其他 2 组有明显的区别，说明自适应融合的用户社交特征整合了显式和隐式社交特征



(a) LastFM数据集 (b) FilmTrust数据集

(c) Yelp数据集 (d) Epinions数据集

Fig. 7 The social feature distribution of adaptive fusion

图 7 自适应融合社交特征分布

特有的优势，带来了更准确的推荐效果。同时，$\boldsymbol{E}_{\mathrm{UTU},u}$和$\boldsymbol{E}_{\mathrm{UIU},u}$在空间中的一侧聚集，$\boldsymbol{E}_{\mathrm{joint},u}$处于相对中部的位置，这表明融合特征在整合其他 2 种特征时不倾向于任何一方，能有效区分不同的用户个体或群体，这种明显的区分是提升推荐质量的基础。

2）$\boldsymbol{E}_{\mathrm{joint},u}$相较于其他 2 组特征（显式和隐式）展现出更高的聚类密度，结构也更有序。这一现象不仅表明融合特征在降低数据噪声和提升特征表达的有效性方面具有明显优势，还说明其在保持数据一致性方面更强。这说明，融合特征能够更加精确地捕捉到用户的社交行为模式，对于构建更准确的用户社交特征和提高推荐的相关性至关重要。此外，$\boldsymbol{E}_{\mathrm{joint},u}$的紧密分布也表明其在多样性和覆盖面上具有较好的表现，可有效地提升推荐模型在面对不同用户和场景时的适应性，增强其对新颖项目的探索能力。通过整合显式和隐式社交特征，融合特征不仅提高了推荐结果的准确性，也使推荐模型对动态社交有新的理解，这在复杂的用户交互网络推荐中尤为重要。

对问题 3 的回答：

自适应因子α_u允许推荐模型自适应地调整每个用户的显式和隐式社交特征的融合比例，在处理不同社交和交互密度的数据集时，具有显著优势。采用α_u的社交推荐模型能够灵活地应对新的应用场景，提高其在多样化应用场景下的可扩展性。

因哈达玛积能增强特征间的交互与对比，并保持模型简单，使得所提特征融合方法在处理社交推荐时具有较高的普适性。

3.5 超参数敏感度分析（问题 4）

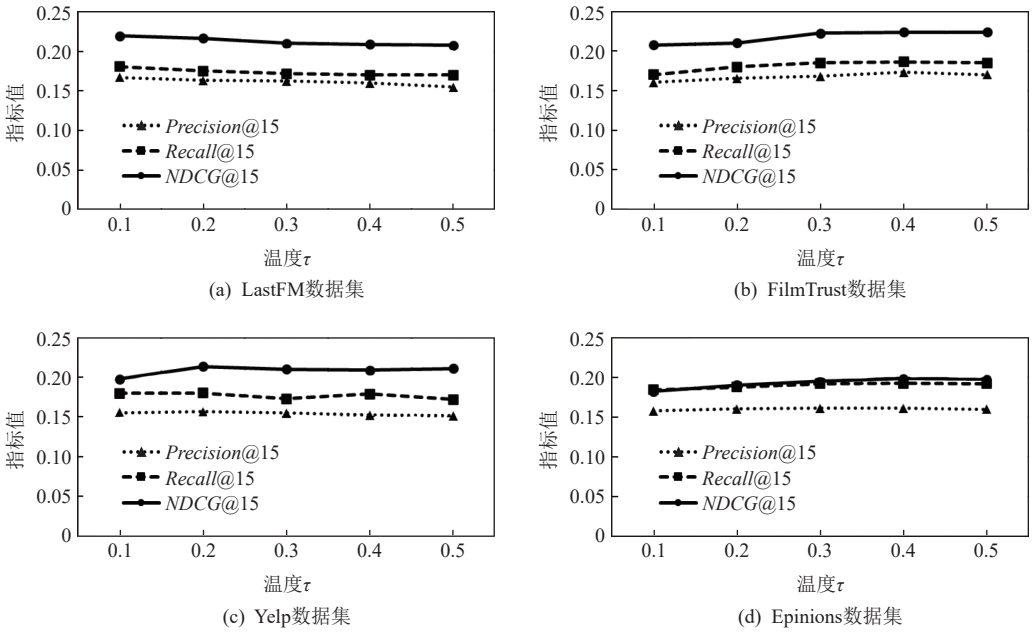
AFS-GAN 模型中的伪项目生成器与判别器是执行推荐任务的核心组件，为这些组件选择恰当的超参数是保障推荐模型效率和质量的关键。我们选择生成器 ST-GS 层的温度τ、生成器正则化系数λ_2和判别器正则化系数λ_3作为研究对象。

3.5.1 温度τ的影响

ST-GS 是一种在神经网络模型中处理离散变量

的技术，其核心在于使用可微分的方式近似离散选择的过程。这种方法通过引入 Gumbel 分布与 Softmax 函数相结合的方式，允许模型在反向传播过程中优化离散输出，使得梯度在原本是离散的节点中得以传播。ST-GS 层的核心参数是温度 τ，该参数对离散决策的随机程度起到关键作用。当温度低时，Softmax 函数的输出接近于硬最大值(hard max)，即输出向量中的最大值接近 1，其余值接近 0，模拟了更确定性的离散决策。当温度高时，Softmax 函数的输出趋向于更均匀的分布，每个元素都有较高的概率被选中，增加了输出的熵，使决策过程过于随机。

考虑到温度参数 τ 不同取值对调节输出的确定性有不同程度的影响，帮助 AFS-GAN 模型探索并获得最优生成策略，我们在实验中设置 $\tau \in \{0.1,0.2,0.3,0.4,0.5\}$，不同取值对应的推荐指标变化情况，如图 8 所示。当 τ 分别为 0.1, 0.4, 0.2, 0.4 时，LastFM, FilmTrust, Yelp, Epinions 数据集的各项推荐指标最优。在这 4 个数据集中，LastFM 的社交密度最高，其次是 Yelp，这说明温度参数 τ 取值较低时，模型能锁定最有效的用户社交特征。FilmTrust 和 Epinions 数据集的社交密度较低，τ 取值较高可促使模型充分探索和利用用户的社交特征。



(a) LastFM数据集　(b) FilmTrust数据集

(c) Yelp数据集　(d) Epinions数据集

Fig. 8 The effect of temperature τ in ST-GS layer

图 8 ST-GS 层温度 τ 的影响

3.5.2 正则化系数 λ_2 和 λ_3 的影响

鉴于 GAN 训练的动态性和复杂性，我们对正则化系数 λ_2 和 λ_3 进行分析，这样不仅可评估单个超参数的影响，还能探索 2 个超参数如何共同影响模型的学习过程和最终性能。一方面，在保证算法正则化有效的前提下，为避免过度抑制生成器的创造能力，设置 $\lambda_2 \in \{0.001,0.01\}$，当 λ_2=0.001 时可确保生成器在学习过程中有足够的灵活性来适应数据的多样性，而稍大的值(λ_2=0.01)则用来探索正则化效果更明显的情况。另一方面，判别器直接决定了生成样本的质量，其相较于生成器在训练过程中更为重要，需对 λ_3 作更细致的探索，设置 $\lambda_3 \in \{0.001,0.005,0.01,0.02,0.05\}$，期望在更广泛的参数范围内探寻模型调节伪项目与真实项目间差异性的最佳平衡点。λ_2 和 λ_3 的不同取值的推荐指标结果如图 9 所示，可以看出当 λ_2 = 0.001时，3 项指标的结果普遍优于 λ_2=0.01 的结果。

对于 LastFM 和 Yelp 数据集，当判别器的正则化系数 λ_3 增加时，判别器的约束变强，超过一定约束值后会导致生成器过度拟合至判别器的特定偏好上，使得各项推荐指标呈现先高后低的变化趋势，当 λ_3 = 0.005时，各项推荐指标处于峰值。这是由于，这 2 个数据集社交密度均高于交互密度，社交信息的影响比较大，丰富的社交信息可帮助生成器产生更准确的社交特征，需要一个更新较快的判别器来有效学习这些丰富的社交特征。

对于 FilmTrust 和 Epinions 数据集，当判别器的正则化系数 λ_3 增加时，各项推荐指标的变化幅度较为平缓，当 λ_3 = 0.02时，各项推荐指标达到最优。相较于 LastFM 和 Yelp 数据集，FilmTrust 和 Epinions 数据集的社交密度低，较高的正则化系数可放缓判别

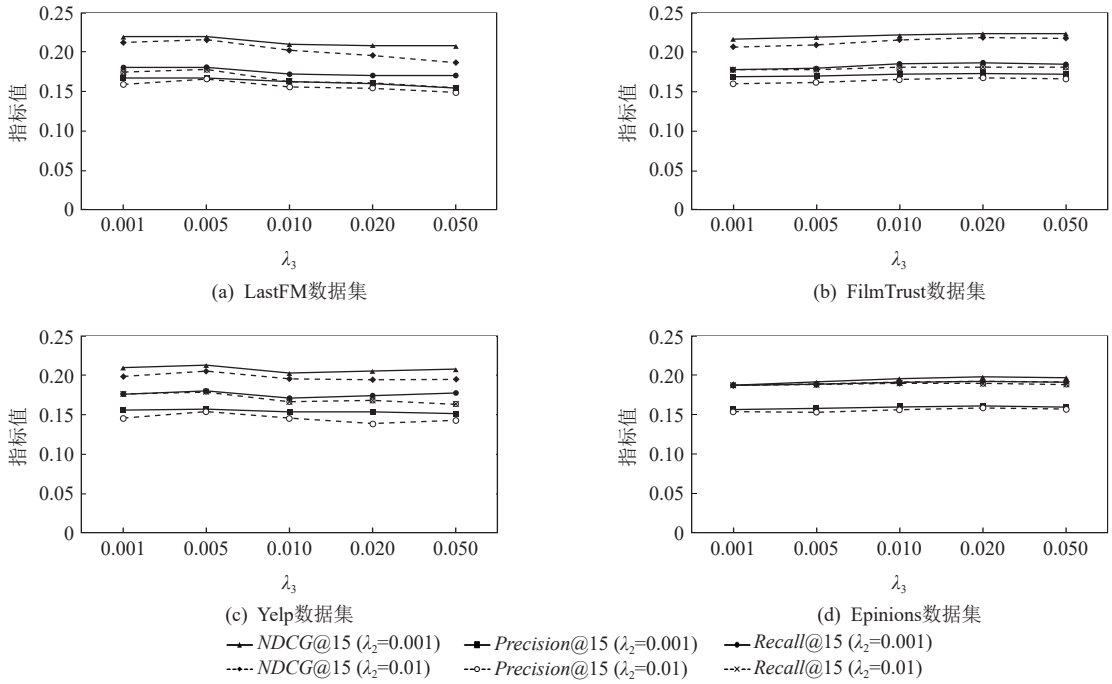


Fig. 9 Effect of the regularization coefficient λ_2, λ_3

图 9 正则化系数λ_2,λ_3的影响

器的学习速度，给生成器更多的时间来学习用户社交特征，并充分利用丰富的交互信息。

对问题 4 的回答：

当数据集社交密度较高时，温度 τ 应取较低的值，LastFM 数据集 τ 取值为 0.1，Yelp 数据集 τ 取值为 0.2。此时，λ_2和λ_3分别设置为 0.001, 0.005。

反之，当社交密度较低时，温度 τ 应取较高的值，FilmTrust 和 Epinions 数据集均取 $\tau = 0.4$，λ_2和λ_3设置分别为 0.001, 0.02。

3.6 推荐场景案例分析

在实验的 4 个数据集中，LastFM 数据规模小且社交密度与交互密度最高，Epinions 数据规模大且社交密度与交互密度最低，故我们选用这 2 个数据集作为推荐应用场景的分析对象。这里从 LastFM 数据集中随机选择项目编号 ID 为 2 的用户，其 1 阶显式社交关系有 13 条，2 阶隐式社交关系有 32 条，项目交互数为 50。在 Epinions 数据集中随机选择 ID 为 8040 的用户，其 1 阶显式社交关系有 6 条，2 阶隐式社交关系有 73 条，项目交互数为 116。

AFS-GAN 模型先向目标用户输出一个含有 15 个项目的推荐列表，再将该目标用户真实交互过的评分最高的 15 个项目挑选出来，按照预测评分从高到低排序。

图 10 是这 2 组项目的热力图，展示了预测排名与实际排名之间的匹配程度，色彩的深度表示匹配的准确程度。图中黑色表示完全匹配(预测项目 ID 与实际排名项目的 ID 相同且排名位置完全一致)，灰色表示部分匹配(项目 ID 相同但排名位置不一致)，白色表示无匹配(项目 ID 与排名位置均不匹配)。理想情况下，所有的匹配(黑色块)应该位于图中的对角线上，表示预测结果与实际情况完全一致。LastFM 数据集的热力图显示 AFS-GAN 模型的推荐项目与目标用户的实际项目交互结果较为一致，仅出现了个别的部分匹配和无匹配的情况。Epinions 数据集的热力图反映模型的推荐结果弱于 LastFM，特别是排名靠后的项目出现了无匹配的情况。

LastFM 数据集的社交关系和交互数据相对丰富，有利于 AFS-GAN 模型通过元路径更有效地捕捉用户的社交特征，从而能更准确地预测用户偏好。而且，模型有效利用自适应特征融合机制学习和强化用户的社交特征，得其能更好地理解用户的交互偏好，从而表现出优秀的推荐能力，并在热力图上生动地表现出来(黑色块集中在对角线上)。相比之下，尽管面对 Epinions 数据集规模大、数据稀疏的挑战，AFS-GAN 模型依然展示出其在提高模型实用性和适应性方面的潜力。即使在社交关系较少的情况下，通过精心设计的元路径、自适应特征融合机制，模型依然能有效捕捉 1 阶显式和 2 阶隐式社交关系，推动其丰富社交信息。并且，伪项目生成器和判别器进一步增强了模型在数据稀疏环境下的性能，通过不

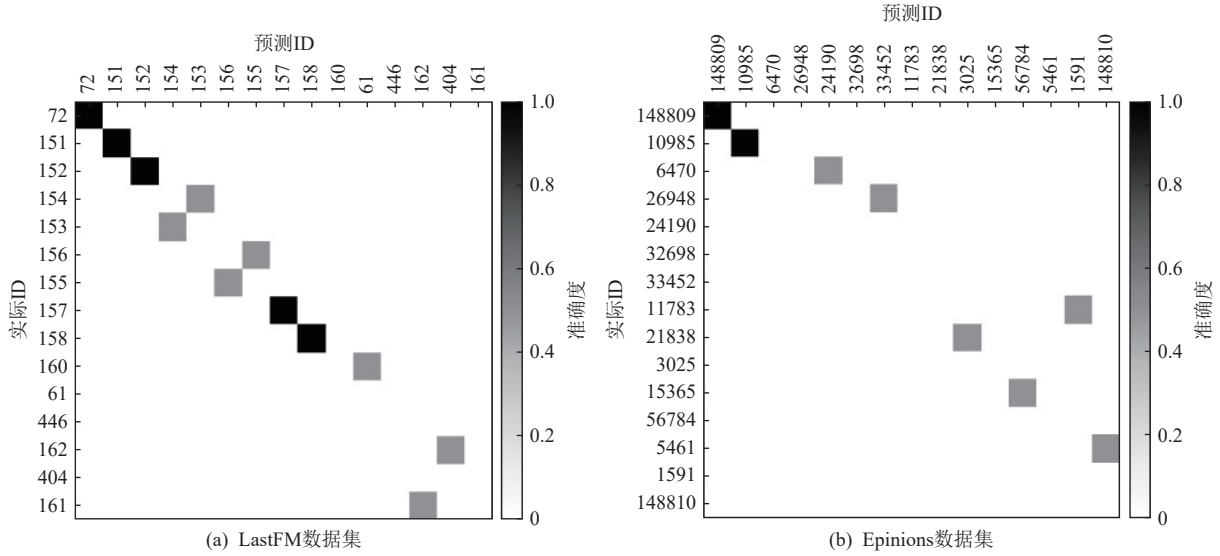


Fig. 10 Scenario case analysis of recommended application

图 10 推荐应用场景案例分析

断迭代优化以生成更加符合用户实际偏好的推荐列表。这种情况与 3.2 节的对比实验一致, 进一步阐明了 AFS-GAN 具有良好的实用性和适应性。

4 时间复杂度分析

AFS-GAN 模型的计算主要由用户社交特征构建与自适应融合模块、伪项目生成器模块、伪项目判别器模块、模型训练与推荐模块 4 个模块构成, 我们分析每个模块的时间复杂度, 以评估模型的整体复杂度。

1)用户社交特征构建与自适应融合模块的时间复杂度

在这个模块中, 模型首先按照 UTU 和 UIU 元路径搜索每个用户的 1 阶显式和 2 阶隐式社交用户序列, 搜索时间复杂度取决于用户节点数和节点的度。假设平均每个用户的 1 阶和 2 阶用户数分别为 M 和 N, 则总的搜索复杂度约为 $O(u(M+N))$, 其中 u 是用户总数。

对于每个用户得到的社交关系序列, 通过 Skip-gram 模块分别生成 $\boldsymbol{E}_{\mathrm{UTU},u}$ 和 $\boldsymbol{E}_{\mathrm{UIU},u}$ 特征向量。假设用户序列平均长度为 p, 嵌入维度为 d, 窗口大小为 w, 迭代次数为 t, 则 Skip-gram 模块的时间复杂度为 $O(updwt)$。

自编码器生成重构向量 $\hat{\boldsymbol{E}}_{\mathrm{UTU},u}$ 和 $\hat{\boldsymbol{E}}_{\mathrm{UIU},u}$ 的维度与输入相同, 中间隐层维度为 d_{h}, 共有 3 层。自编码器的时间复杂度约为 $O(udd_{\mathrm{h}})$。

将 $\hat{\boldsymbol{E}}_{\mathrm{UTU},u}$ 和 $\hat{\boldsymbol{E}}_{\mathrm{UIU},u}$ 通过哈达玛积融合为 $\boldsymbol{E}_{\mathrm{joint},u}$。哈达玛积的操作是逐元素相乘, 时间复杂度约为 $O(ud)$。

以上时间复杂度总和为 $O(u(M+N)+updwt+udd_{\mathrm{h}}+ud)$, 由于 Skip-gram 的时间复杂度 $O(updwt)$ 在本模块的计算中占主导地位, 即该项的增长速率远超其他项。另外, 数据集和模型结构确定后, M, N, d_{h} 均可视为常数, 则用户社交特征构建与自适应融合模块的时间复杂度约为 $O(updwt)$。

2)伪项目生成器模块的时间复杂度

本模块涉及 3 个主要步骤: ST-GS 重参数化、查询项目偏好矩阵 $\boldsymbol{H}$ 和计算 BPR 损失函数。每个维度为 d 的用户的融合社交特征 $\boldsymbol{E}_{\mathrm{joint},u}$ 在 ST-GS 层(包括添加 Gumbel 噪声和 Softmax 计算)的时间复杂度主要由 Softmax 运算决定, 为 $O(d)$。项目偏好矩阵 $\boldsymbol{H}$ 的维度为 $d\times d$, 但因 ST-GS 需输出一个与 $\boldsymbol{H}$ 的行数相匹配的向量, 故实际复杂度为 $O(d)$。在计算 BPR 损失函数时, 若每个用户 u 与 K 个伪项目进行比较, 则该步时间复杂度为 $O(uK)$。因此, 伪项目生成器的整体时间复杂度约为 $O(u(d+d+K))$, 约简为 $O(ud+uK)$。

3)伪项目判别器模块的时间复杂度

本模块涉及 2 个主要步骤, 通过 Skip-gram 模块分别生成用户 $\boldsymbol{e}_u$、正项目 $\boldsymbol{e}_{i^+}$ 和负项目 $\boldsymbol{e}_{i^-}$ 来计算 BPR 损失函数。Skip-gram 模块和用户社交特征构建与自适应融合模块中的相同, 其时间复杂度为 $O(updwt)$。在计算 BPR 损失函数时, 模型只需计算用户 $\boldsymbol{e}_u$ 与正项目 $\boldsymbol{e}_{i^+}$、伪项目 $\boldsymbol{e}_{\hat{i}}$ 和负项目 $\boldsymbol{e}_{i^-}$ 的概率损失。此时, 伪项目判别器的时间复杂度是 $O(u(i^+ +K+i^-))$, 这里的 i^+ 和 i^- 分别代表正项目和负项目的数量。因每个用户对应的正、伪、负项目数大致相等, 且伪项目 K 占主导地位, 故复杂度可简化为 $O(uK)$。因此, 项目判别

的整体时间复杂度约为 $O(updwt+uK)$。

4）模型时间复杂度

AFS-GAN 模型单次训练和推荐的时间复杂度由上面 3 个模块构成，即 $O(utpwd)+O(ud+uK))+O(updwt+uK)$，约简为 $O(u(tpwd+d+K))$。考虑到模型训练与推荐是 GAN 不断迭代优化的过程，则 AFS-GAN 模型的最终时间复杂度为 $O(\xi u(tpwd+d+K))$，ξ 为迭代次数。表 6 是 AFS-GAN 与对比模型在时间复杂度上的对比结果。

由表 6 中的对比结果可知，AFS-GAN 在时间复杂度上具有一定的优势。①AFS-GAN 对用户、项目和嵌入维度的依赖均为线性，避免了传统模型中常见的 2 次增长问题。这种线性特性使得 AFS-GAN 在处理大规模数据时能够保持高效的计算性能，尤其适合用户和项目数量庞大的推荐场景。②通过窗口滑动，AFS-GAN 将计算范围限制在用户行为序列的局部窗口内，避免了传统方法中对全局图的遍历，降低了计算复杂度。③由伪项目替代全量项目，AFS-GAN 大幅减少了生成器和判别器的计算负载。虽然 AFS-GAN 在训练过程中需多次迭代，导致训练时间有所增长，但这种时间消耗所带来的推荐性能提升是值得的。

Table 6 Complexity Comparison of Model Time

表 6 模型时间复杂度对比

模型	时间复杂度	参数说明
ESRF	$O(L(\lvert\boldsymbol{A}^+\rvert+\lvert\boldsymbol{Y}^+\rvert+Ku)d+Kud^2+u^2t)$	L 为图卷积层数，$\lvert\boldsymbol{A}^+\rvert$和$\lvert\boldsymbol{Y}^+\rvert$分别为社交矩阵、交互矩阵的非零元素数，$K$ 为平均邻居数，d 为嵌入维度，u 为用户总数，t 为生成器中隐层的维度
GNNRec	$O(d^2(TN+uLK))$	T 和 L 为图注意力和图神经网络层数，N 为会话长度，K 为平均邻居数，d 为嵌入维度，u 为用户总数
GBSR	$O(L(u+v)d^2+Sd^2+\xi B^2d)$	L 为图卷积层数，u 和 v 分别为用户、项目数，S 为社交图节点数，ξ 为迭代次数，B 为批次大小，d 为嵌入维度
CDRSB	$O((uK_u+vK_v)d+(u+v)d^2+B^2d+umd)$	u 和 v 分别为用户、项目数，K_u 和 K_v 分别为用户、项目平均邻居数，m 为历史交互数，B 为批次大小，d 为嵌入维度
DGNN	$O(\lvert\boldsymbol{M}\rvert+\lvert E\rvert+d^2)$	$\lvert\boldsymbol{M}\rvert$为异构图节点嵌入向量大小，$\lvert E\rvert$为异构图边数，$d$ 为嵌入维度
DASO	$O(\xi d(uv+u^2))$	u 和 v 分别为用户、项目数，ξ 为迭代次数，d 为嵌入维度
GANMF	$O(\xi Bv(d+H))$	v 为项目数，ξ 为迭代次数，B 为批次大小，d 为嵌入维度，H 为自编码器隐层维度
DAEGAN	$O(\xi Bv(G+D))$	v 为项目数，ξ 为迭代次数，B 为批次大小，d 为嵌入维度，G 和 D 为生成器和判别器隐层维度
AFS-GAN	$O(\xi u(tpwd+d+K))$	ξ 为迭代次数，u 为用户数，p 为用户序列平均长度，w 为窗口大小，d 为嵌入维度，t 为迭代次数，K 为伪项目数

5 结论与下一步工作

AFS-GAN 模型使用简单元路径捕获用户的 1 阶显式和 2 阶隐式社交特征，显著降低了复杂度，加强了其在真实世界推荐系统中的实用性。此外，AFS-GAN 模型通过自适应融合社交特征的方法精确提取用户社交行为，增强了对稀疏数据的处理能力，从而适应不同规模和密度的社交网络数据，提高了适用性。AFS-GAN 模型通过引入 ST-GS 技术，以及使用 BPR 损失函数作为判别器简化了学习过程，优化了 GAN 的训练效率，显著提升了其处理复杂数据时的整体效率。

AFS-GAN 模型为社交推荐领域提供了一种新的研究视角，可激励研究者进一步探索社交网络数据提升推荐效果的方法。下一步，我们将深入研究其他生成式模型结构和算法，利用其生成缺失数据和增强现有数据的能力，借助丰富的辅助信息（包括用户画像、项目说明、评价文本等）强化用户的复杂偏好和动态行为，进而在提升推荐准确性的同时，兼顾推荐的多样性和新颖性。

作者贡献声明：钱忠胜提出主体思想、整体规划、写作指导；朱辉负责算法设计、数据处理、论文写作；俞情媛和李玉龙负责主要实验及结果分析；万子珑负责协助实验数据采集、程序设计、文献整理。

参 考 文 献

[1] Lazarsfeld F P, Merton K R. Friendship as a social process: A substantive and methodological analysis[J]. Freedom and Control in Modern Society, 1954, 18(1): 18−66

[2] Sousa M A, Almeida M J, Figueiredo F. Analyzing and modeling user curiosity in online content consumption: A LastFM case study[C]// Pro of the 2019 IEEE/ACM Int Conf on Advances in Social Networks Analysis and Mining. New York: ACM, 2019: 426−431

[3] Schedl M, Brandl S, Lesota O, et al. LFM−2b: A dataset of enriched

music listening events for recommender systems research and fairness analysis[C]//Proc of the 2022 Conf on Human Information Interaction and Retrieval (CHIIR'22). New York: ACM, 2022: 337−341

[4] Chang Yaomin, Chen Chuan, Hu Weibo, et al. MEGNN: Meta-path extracted graph neural network for heterogeneous graph representation learning[J]. Knowledge-Based Systems, 2022, 235: 107611

[5] Sun Yizhou, Han Jiawei, Yan Xifeng, et al. PathSim: Meta path-based top-*k* similarity search in heterogeneous information networks[J]. VLDB Endowment, 2011, 4(11): 992−1003

[6] Song Yuqi, Gao Min, Yu Junliang, et al. Social recommendation based on implicit friends discovering via meta-path[C]//Proc of the 30th IEEE Int Conf on Tools with Artificial Intelligence (ICTAI). Piscataway, NJ: IEEE, 2018: 197−204

[7] Yu Junliang, Gao Min, Li Jundong, et al. Adaptive implicit friends identification over heterogeneous network for social recommendation[C]//Proc of the 27th ACM Int Conf on Information and Knowledge Management (CIKM'18). New York: ACM, 2018: 357−366

[8] Li Munan, Liu Kai, Liu Hongbo, et al. Heterogeneous meta-path graph learning for higher-order social recommendation[J]. ACM Transactions on Knowledge Discovery from Data, 2024, 18(8): 1−25

[9] Jiang Yanbin, Ma Huifang, Zhang Xiaohui, et al. Incorporating metapath interaction on heterogeneous information network for social recommendation[J]. Frontiers of Computer Science, 2024, 18: 181302

[10] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139−144

[11] Wang Jun, Yu Lantao, Zhang Weinan, et al. IRGAN: A minimax game for unifying generative and discriminative information retrieval models[C]//Proc of the 40th Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR'17). New York: ACM, 2017: 515−524

[12] Chae D K, Kang J S, Kim S W, et al. CFGAN: A generic collaborative filtering framework based on generative adversarial networks[C]// Proc of the 27th ACM Int Conf on Information and Knowledge Management (CIKM'18). New York: ACM, 2018: 137−146

[13] Wang Hongwei, Wang Jia, Wang Jialin, et al. GraphGAN: Graph representation learning with generative adversarial nets[C]//Proc of the 32nd AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2018, 32(1)

[14] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]//Proc of the 31st Int Conf on Machine Learning. New York: PMLR, 2014: 387−395

[15] Fan Wenqi, Derr T, Ma Yao, et al. Deep adversarial social recommendation[C]//Proc of the 28th Int Joint Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2019: 1351−1357

[16] Dervishaj E, Cremonesi P. GAN-based matrix factorization for recommender systems[C]//Proc of the 37th ACM/SIGAPP Symp on Applied Computing (SAC'22). New York: ACM, 2022: 1373−1381

[17] Yu Junliang, Gao Min, Yin Hongzhi, et al. Generating reliable friends via adversarial training to improve social recommendation[C]//Proc of the 2019 IEEE Int Conf on Data Mining (ICDM). Piscataway, NJ: IEEE, 2019: 768−777

[18] Jia Xiaosheng, Zhao Zhongying, Li Chao, et al. Heterogeneous network representation learning method fusing mutual information and multiple meta-paths[J]. Journal of Software, 2023, 34(7): 3256−3271 (in Chinese)
(贾霄生, 赵中英, 李超, 等. 互信息与多条元路径融合的异质网络表示学习方法[J]. 软件学报, 2023, 34(7): 3256−3271)

[19] Shi Chuan, Hu Binbin, Zhao Wayne Xin, et al. Heterogeneous information network embedding for recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(2): 357−370

[20] Dhelim S, Ning Huansheng, Aung N, et al. Personality-aware product recommendation system based on user interests mining and metapath discovery[J]. IEEE Transactions on Computational Social Systems, 2021, 8(1): 86−98

[21] Miao Hang, Li Anchen, Yang Bo. Meta-path enhanced lightweight graph neural network for social recommendation[C]//Proc of the 27th Database Systems for Advanced Applications(DASFAA' 2022). Berlin: Springer, 2022: 134−149

[22] Li Ruirui, Wu Xian, Wang Wei. Adversarial learning to compare: Self-attentive prospective customer recommendation in location based social networks[C]//Proc of the 13th Int Conf on Web Search and Data Mining. New York: ACM, 2020: 349−357

[23] Chen Jiawei, Wang Can, Zhou Sheng, et al. Samwalker: Social recommendation with informative sampling strategy[C]//Proc of the World Wide Web Conf. New York: ACM, 2019: 228−239

[24] Wu Le, Sun Peijie, Fu Yanjie, et al. A neural influence diffusion model for social recommendation[C]//Proc of the 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2019: 235−244

[25] Wu Le, Li Junwei, Sun Peijie, et al. Diffnet++: A neural influence and interest diffusion network for social recommendation[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(10): 4753−4766

[26] Yu Junliang, Yin Hongzhi, Li Jundong, et al. Enhancing social recommendation with adversarial graph convolutional networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(8): 3727−3739

[27] Yu Junliang, Yin Hongzhi, Li Jundong, et al. Self-supervised multi-channel hypergraph convolutional network for social recommendation[C]//Proc of the Web Conf 2021. New York: ACM, 2021: 413−424

[28] Yu Junliang, Yin Hongzhi, Gao Min, et al. Socially-aware self-supervised tri-training for recommendation[C]//Proc of the 27th ACM SIGKDD Conf on Knowledge Discovery & Data Mining. New York: ACM, 2021: 2084−2092

[29] Liu Chun, Li Yuxiang, Lin Hong, et al. GNNRec: Gated graph neural network for session-based social recommendation model[J]. Journal of Intelligent Information Systems, 2023, 60(1): 137−156

[30] Yang Yonghui, Wu Le, Wang Zihan, et al. Graph bottlenecked social recommendation[C]//Proc of the 30th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2024: 3853−3862

[31] Wang Bohao, Chen Jiawei, Li Changdong, et al. Distributionally robust graph-based recommendation system[C]//Proc of the ACM Web Conf 2024. New York: ACM, 2024: 3777−3788

[32] Chen Sirui, Chen Jiawei, Zhou Sheng, et al. SIGformer: Sign-aware

graph transformer for recommendation[C]//Proc of the 47th Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2024: 1274−1284

[33] Chen Sirui, Han Shen, Chen Jiawei, et al. Rankformer: A graph transformer for recommendation based on ranking objective[J]. arXiv preprint, arXiv: 250316927, 2025

[34] Guthrie D, Allison B, Liu Wei, et al. A closer look at skip-gram modelling[C]//Proc of the 5th Int Conf on Language Resources and Evaluation (LREC'06). Paris: ELRA, 2006: 1222−1225

[35] Wang Yaqing, Piao Hongming, Dong Daxiang, et al. Warming up cold-start CTR prediction by learning item-specific feature interactions[C]//Proc of the 30th ACM SIGKDD Conf on Knowledge Discovery and Data Mining. New York: ACM, 2024: 3233−3244

[36] Xiao Bowen, Chen Deng. Explicitly exploiting implicit user and item relations in graph convolutional network (GCN) for recommendation[J]. Electronics, 2024, 13(14): 2811

[37] Zhang Qi, Cao Longbing, Shi Chongyang, et al. Neural time-aware sequential recommendation by jointly modeling preference dynamics and explicit feature couplings[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(10): 5125−5137

[38] Mikolov T, Chen Kai, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint, arXiv: 13013781, 2013

[39] Caselles-Dupré H, Lesaint F, Royo-Letelier J. Word2vec applied to recommendation: Hyperparameters matter[C]//Proc of the 12th ACM Conf on Recommender Systems. New York: ACM, 2018: 352−356

[40] Lai Jie, Wang Xiaodan, Xiang Qian, et al. Review on autoencoder and its application[J]. Journal on Communications, 2021, 42(9): 218−230 (in Chinese)
(来杰, 王晓丹, 向前, 等. 自编码器及其应用综述[J]. 通信学报, 2021, 42(9): 218−230)

[41] Li Pengzhi, Pei Yan, Li Jianqiang. A comprehensive survey on design and application of autoencoder in deep learning[J]. Applied Soft Computing, 2023, 138: 110176

[42] Yuan Feiniu, Zhang Lin, Shi Jinting, et al. Theories and applications of auto-encoder neural networks: A literature survey[J]. Chinese Journal of Computers, 2019, 42(1): 203−230 (in Chinese)
(袁非牛, 章琳, 史劲亭, 等. 自编码神经网络理论及应用综述[J]. 计算机学报, 2019, 42(1): 203−230)

[43] Zhao Haiyan, Ma Quanyi, Cao Jian, et al. Dynamic neural network for incremental learning with task extended: Research progress and prospect[J]. Acta Electronic Sinica, 2023, 51(6): 1710−1724 (in Chinese)
(赵海燕, 马权益, 曹健, 等. 面向任务扩展的增量学习动态神经网络: 研究进展与展望[J]. 电子学报, 2023, 51(6): 1710−1724)

[44] Jiao Licheng, Sun Qigong, Yang Yuting, et al. Development, implementation and prospect of FPGA-based deep neural networks[J]. Chinese Journal of Computers, 2022, 45(3): 441−471 (in Chinese)
(焦李成, 孙其功, 杨育婷, 等. 深度神经网络 FPGA 设计进展、实现与展望[J]. 计算机学报, 2022, 45(3): 441−471)

[45] Li Xinchun, Zhan Dechuan, Yang Jiaqi, et al. Deep multiple instance selection[J]. Science China Information Sciences, 2021, 64: 130102

[46] Yu Yonghong, Gao Yang, Wang Hao. A ranking rased poisson matrix factorization model for point-of-interest recommendation[J]. Journal of Computer Research and Development, 2016, 53(8): 1651−1663 (in Chinese)
(余永红, 高阳, 王皓. 基于 Ranking 的泊松矩阵分解兴趣点推荐算法[J]. 计算机研究与发展, 2016, 53(8): 1651−1663)

[47] Wang Li, Xu Min, Zhang Quangui, et al. Causal disentanglement for regulating social influence bias in social recommendation[J]. Neurocomputing, 2025, 618: 129133

[48] Xia Lianghao, Shao Yizhen, Huang Chao, et al. Disentangled graph social recommendation[C]//Proc of the 39th IEEE Int Conf on Data Engineering (ICDE). Piscataway, NJ: IEEE, 2023: 2332−2344

[49] Li Mingjie, Liu Yunhan, Jiang Weiwei, et al. Improved negative sampling method in collaborative filtering recommendation based on generative adversarial network[J]. Electronic Commerce Research and Applications, 2024, 66: 101412



Qian Zhongsheng, born in 1977. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include machine learning, software engineering, and intelligent software.
钱忠胜, 1977 年生。博士, 教授, 博士生导师。CCF 高级会员。主要研究方向为机器学习、软件工程、智能化软件。



Zhu Hui, born in 1976. PhD candidate. Student member of CCF. His main research interests include intelligent recommender system, deep learning, large language models, and software engineering.
朱　辉, 1976 年生。博士研究生。CCF 学生会员。主要研究方向为智能推荐系统、深度学习、大语言模型、软件工程。



Yu Qingyuan, born in 1997. PhD candidate. Her research interests include machine learning, software engineering, and intelligent software.
俞情媛, 1997 年生。博士研究生。主要研究方向为机器学习、软件工程、智能化软件。



Li Yulong, born in 1997. PhD candidate. His main research interests include intelligent recommender system and software engineering.
李玉龙, 1997 年生。博士研究生。主要研究方向为智能推荐系统、软件工程。



Wan Zilong, born in 1999. Master candidate. His main research interests include software engineering and machine learning.
万子珑, 1999 年生。硕士研究生。主要研究方向为软件工程、机器学习。