

# 融合多粒度社团特征与快速并行矩阵分解的复杂网络个性化推荐算法

李贵平, 艾 均, 苏 湛

上海理工大学光电信息与计算机工程学院, 上海

收稿日期: 2025年3月7日; 录用日期: 2025年4月1日; 发布日期: 2025年4月10日

## 摘 要

针对复杂网络环境下个性化推荐系统面临的数据稀疏性、计算效率及用户偏好动态性等挑战, 本文提出了一种融合多粒度社团特征与快速并行矩阵分解的推荐算法。该算法首先构建基于用户相似度的复杂网络, 并引入改进的社团检测算法实现多粒度社团结构划分, 生成粗细粒度不同的用户群体; 随后, 设计了一种基于用户活跃度的自适应层次选择机制, 旨在根据用户行为特征动态选择合适的社团层次进行推荐, 以缓解数据稀疏性问题并提升推荐精度。在此基础上, 本文提出一种基于多层用户相似性网络的快速并行梯度下降矩阵分解模型, 有效捕捉用户在不同粒度上的偏好特征, 并结合用户全局偏好信息与局部社团特征。多个真实数据集上的实验结果表明, 相较于传统推荐算法, 融合多粒度社团特征与快速并行矩阵分解算法在准确性、精确率和召回率等指标上均有不同程度的提升, 验证了该算法在提高推荐质量和可扩展性方面的有效性。

## 关键词

用户相似性, 多粒度层次划分, 复杂网络, 推荐算法, 并行梯度下降矩阵分解, 社团检测

# Personalized Recommendation Algorithm in Complex Networks via Multi-Granularity Community Features and Fast Parallel Matrix Factorization

Guiping Li, Jun Ai, Zhan Su

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 7<sup>th</sup>, 2024; accepted: Apr. 1<sup>st</sup>, 2025; published: Apr. 10<sup>th</sup>, 2025

文章引用: 李贵平, 艾均, 苏湛. 融合多粒度社团特征与快速并行矩阵分解的复杂网络个性化推荐算法[J]. 运筹与模糊学, 2025, 15(2): 443-459. DOI: 10.12677/orf.2025.152096

## Abstract

To address the challenges of data sparsity, computational efficiency, and dynamic user preferences in personalized recommendation systems within complex networks, this paper proposes a recommendation algorithm that integrates multi-granularity community features with fast parallel matrix factorization. First, a complex network based on user similarity is constructed, and an improved community detection algorithm is introduced to achieve multi-granularity community structure partitioning, generating user groups at different levels of granularity. Then, an adaptive hierarchical selection mechanism based on user activity is designed to dynamically select the appropriate community level for recommendation according to user behavior characteristics, thereby alleviating data sparsity and improving recommendation accuracy. On this basis, we propose a fast parallel gradient descent matrix factorization model based on multi-layer user similarity networks, effectively capturing user preference features at different granularities while incorporating both global user preference information and local community characteristics. Experimental results on multiple real-world datasets demonstrate that, compared with traditional recommendation algorithms, the proposed approach achieves significant improvements in accuracy, precision, and recall by integrating multi-granularity community features with fast parallel matrix factorization, verifying its effectiveness in enhancing recommendation quality and scalability.

## Keywords

User Similarity, Multi-Granularity Hierarchical Partitioning, Complex Network, Recommendation Algorithm, Parallel Gradient Descent Matrix Factorization, Community Detection

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

推荐系统是一种能够解决信息过载问题并提供个性化服务的有效工具[1]。通过对用户个性、习惯、偏好的分析,建立用户与项目的动态关联模型,有效过滤无关信息,对不同的用户提供其感兴趣的信息和服务[2]。与传统搜索引擎相比,推荐系统输出结果更符合用户需求,降低二次筛选概率和时间成本[3]。

协同过滤(CF)是推荐系统中应用最广泛的方法,矩阵分解(MF)是其常用技术之一,在解决可扩展性问题上取得了显著进展[4]。矩阵分解算法将用户对物品的评分矩阵分解为用户的潜在因子矩阵和物品的潜在因子矩阵,用户对物品的评分预测值由用户潜在因子矩阵和物品潜在因子矩阵的内积求得[5]。

一方面,矩阵分解技术在推荐系统领域广受关注和应用,并在多个方面取得了显著进展。目前流行趋势主要体现在以下三个方面:

(1) 深度学习与矩阵分解结合。基于深度学习的矩阵分解(Deep Matrix Factorization, DMF)利用特征转换函数生成用户和物品的潜在因子,引入隐反馈嵌入(IFE)整合隐反馈数据,减少参数规模并提升训练效率[6][7]。混合深度概率矩阵分解通过捕捉用户偏好和基于注意力的卷积神经网络建模物品吸引力,提高推荐准确性[8]。

(2) 增量矩阵分解(Incremental Matrix Factorization, IMF)关注实时更新和动态推荐,减少计算开销并快速适应用户行为变化[9]。并行增量矩阵分解(Parallel Incremental Matrix Factorization, PIMF)通过并行化

计算和增量更新提升性能[10]。基于时间敏感更新的增量矩阵分解方法(Time-Sensitive Incremental Matrix Factorization, TS-IMF)考虑用户行为的时效性, 实时调整推荐结果[11] [12]。

(3) 因果推断与矩阵分解结合。因果矩阵分解(Causal Matrix Factorization, CMF) [13]通过识别因果关系减少偏差, 提升推荐的可解释性和公平性。Liu 等人提出因果推断模型(Causal Inference-Based Matrix Factorization, CIMF) [14], 通过调整传统矩阵分解方法来处理因果偏差问题, 在此基础上, Zhou 等研究者 [15]在传统矩阵分解算法中加入因果推断模型, 该方法能够识别和消除因果偏差, 从而优化推荐结果的准确性和可解释性。

尽管矩阵分解算法取得进展, 仍面临挑战: 1) 用户 - 物品交互矩阵稀疏, 难以捕捉有效偏好和特征; 2) 大规模数据实时更新时计算资源消耗大; 3) 现有算法假设用户偏好静态, 难以建模长期兴趣变化。

另一方面, 复杂网络在推荐系统中也起到了相当大的作用。当前研究主要集中在提升预测准确性、推荐多样性和效率。深度协同复杂网络框架(Deep Collaborative Clustering, DCC)采用深度神经网络自动提取用户和项目的高维特征, 并进行复杂网络划分, 从而在大规模推荐场景中有效减少计算开销, 并提升个性化推荐的精度[16]。动态用户复杂网络划分方法(Dynamic User Clustering, DUC)结合增量复杂网络技术, 实时跟踪用户兴趣变化, 提高推荐的响应速度和准确度[17]。跨领域学习框架结合多模态数据融合技术, 通过整合来自多个领域的用户和内容数据, 从多个视角对数据进行复杂网络划分。这种方法显著提高了模型在稀疏数据和噪音环境中的鲁棒性[18]-[22]。

尽管复杂网络与先进技术结合提升了推荐的个性化和准确性, 但仍面临挑战: 1) 冷启动场景下, 数据稀疏性导致难以捕捉潜在相似性; 2) 用户规模增长使传统集中式计算难以满足实时推荐需求; 3) 单一复杂网络划分导致推荐内容缺乏多样性。

因此, 本文对于基于矩阵分解的多层社团结构进行研究, 针对推荐可扩展性和推荐准确性方面的不足, 本文做出了如下贡献:

(1) 本文构建了一个基于用户相似性的复杂网络, 并采用模块度优化的社团检测算法将用户分组到社团中, 形成小的用户群落。这一分组机制有效降低了每个社团的用户处理规模, 使得用户群体能够在不同计算设备上实现分布式并行处理, 从而显著提升了算法在大规模数据集上的处理效率, 并增强了系统的可扩展性。

(2) 本文构建了一个基于用户相似性的多粒度社团结构, 将用户分为粗粒度到细粒度的多个层次社团。该结构使得系统能够根据用户特征选择合适的社团层次, 从而既保证推荐的精确性, 又提供灵活的推荐策略。

(3) 本文设计了一种基于用户活跃度的自适应层次选择机制。根据不同用户活跃度采用不同粒度社团层次以提供精准推荐, 提高算法可扩展性和推荐准确性, 从而缓解了数据稀疏性问题。

(4) 本文进一步提出了一种基于多层用户相似性网络的快速并行梯度下降矩阵分解推荐模型, 通过多层次社团结构捕捉用户的不同尺度特征, 利用用户的全局偏好信息和局部社团特征, 从而提升推荐系统的准确性。

本文第2节将介绍与本文研究相关的重要工作、原理及应用场景; 第3节提出模型; 第4节通过实验对模型进行分析与讨论, 并与现有领先算法进行比较; 第5节总结本文并提出未来展望。

## 2. 相关工作

本文采用复杂网络、Louvain 社团发现、多层复杂网络以及快速并行矩阵分解算法(Fast Parallel Stochastic Gradient Descent, FPSG)等技术, 研究采用多层用户相似性网络进行社团划分后, 如何在每个社团内进行快速分布式并行矩阵分解, 以提升模型在大规模数据下的可扩展性。

复杂网络的社团结构可以标识网络中节点之间的内在联系和相似性特征, 从而揭示节点之间的距离

特征。这些特性在推荐系统中尤为重要，因为推荐系统本身可以被视为一个复杂网络，其中节点代表用户或物品，边则代表用户与物品之间的交互关系。通过对推荐系统进行复杂网络建模，本文可以深入研究推荐系统中用户间的网络结构特征，从而提升推荐的准确性和个性化程度[23]。

文献[24]提出基于模块度优化的社团发现算法(Fast Newman-Girvan Algorithm, FN)。这是一种凝聚式算法，通过逐步合并社团以优化模块度函数，尽管降低了复杂度，但在处理大规模网络时仍耗时较多。为了解决这一问题，Blondel 等[25]提出了基于模块度优化的社团发现算法(Louvain)。该算法基于层次聚类和局部优化算法，显著提高了计算效率，尤其适用于大规模网络。本文采用 Louvain 算法对用户相似性网络进行划分，形成较小规模的用户群体，以提升模型的计算效率和推荐性能。

多层复杂网络[26]：多层复杂网络是由多个相互连接的层次组成的系统，每层代表特定关系或功能，节点通过边连接，跨层节点可能通过跨层边相连。Clauset 等通过层次随机图模型将网络划分为多个层次，并通过概率参数控制层间连接，有效提高了链接预测的准确性。本文将划分后的社团视为节点，构建层次树结构，不同层次代表不同尺度的社团结构。

快速并行矩阵分解算法[27]，这是一种专为共享内存系统设计的高效并行随机梯度下降算法，用于矩阵分解任务。FPSG 通过部分随机方法优化内存访问和收敛速度，采用动态任务调度机制避免线程锁定和负载不平衡问题，显著提升了计算效率。本文选用 FPSG 算法进行矩阵分解，以加速训练和预测任务。

总之，本文通过构建基于用户相似性网络，利用改进的 Louvain 算法进行多层次社团划分，将大规模网络分割为多个层次，并在每个层次内再次划分社团，确保社团内用户具有更高相似性。同时，将社团视为节点构建层次树结构，分析不同层次社团间的关系。最后，在每个用户群落中使用 FPSG 算法对评分矩阵进行训练与预测。

### 3. 模型

#### 3.1. 问题定义

Table 1. Key symbols and their descriptions

表 1. 关键符号及其说明

符号	说明
$D$	整个数据集
$D_1, D_2, \dots, D_n$	将数据集 $D$ 分割成的若干较小的子集
$C$	整个社团
$C_1, C_2, \dots, C_k$	将数据集 $C$ 分割成的若干较小的子集
$A_{uv}$	用户 $u$ 和用户 $v$ 之间的边权重
$T$	由相似性网络提取的最小生成树
$Q(u, v)$	模块度函数
$R$	评分矩阵
$U_{\text{active}}$	最活跃的用户群体
$p_u$	用户 $u$ 的潜在特征向量
$q_v$	用户 $v$ 的潜在特征向量
$w_{uv}$	社团 $C_u$ 与社团 $C_v$ 之间边的权重
$(u, v)$	用户对
$pcc_{uv}$	用户 $u$ 和用户 $v$ 之间的相似值
$k_u$	用户 $u$ 的度数
$m$	网络中的总权重数

推荐系统规模的不断扩大使得模型在处理大批量数据时面临深度学习包含多个隐藏层和大量参数导致训练时间较长和矩阵分解在评分矩阵分布不均时无法充分利用计算资源。在实际应用中,用户的评分行为存在显著差异:部分用户评分频繁且覆盖多个物品类别,而另一部分用户则评分稀疏且兴趣单一。传统矩阵分解算法在处理大规模评分数据时面临严重挑战:一方面,将所有用户-物品评分数据统一建模导致计算复杂度过高,难以充分利用活跃用户的丰富评分信息;另一方面,对于评分稀疏的用户,全局模型难以捕捉其个性化偏好特征,影响预测准确性。因此,如何基于用户的评分行为特征,将大规模用户-物品评分数据合理划分为多个层次的用户社团,并为不同活跃度的用户群体采用相应的预测策略,降低数据稀疏性的同时提升预测准确性,是本研究需要解决的关键问题。

本文所用到的符号及说明如表 1 所示。

### 3.2. 算法整体结构

本文所提出的算法整体结构如图 1 所示。算法从完整数据集出发,基于用户相似性构建复杂网络,并通过颗粒度计算将其划分为多个层次,每个层次包含多个社团。这些社团分布在多台服务器上,形成分布式计算环境,模拟实际处理流程。系统采用基于多粒度社团特征的快速并行矩阵分解算法,高效处理数据并提取隐因子特征。每个服务器训练不同模型,通过计算用户活跃度选择对应颗粒度的社团层次,生成多个预测结果并输出综合预测。所有计算在单台机器上模拟完成,近似多服务器分布式系统的工作效果,确保子集独立处理与实际情况一致。通过这种方式,算法得以验证和优化,同时保证分布式系统中各节点的计算一致性。其详细结构如图 2 所示。

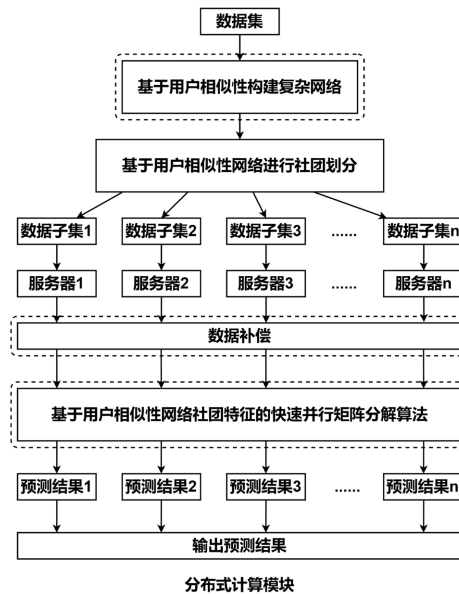


Figure 1. Overall algorithm structure

图 1. 整体算法结构图

首先,用户相似性建模模块通过计算用户相似性构建加权邻接矩阵,提取最小生成树简化网络结构,保留关键连接,形成用户相似性网络。

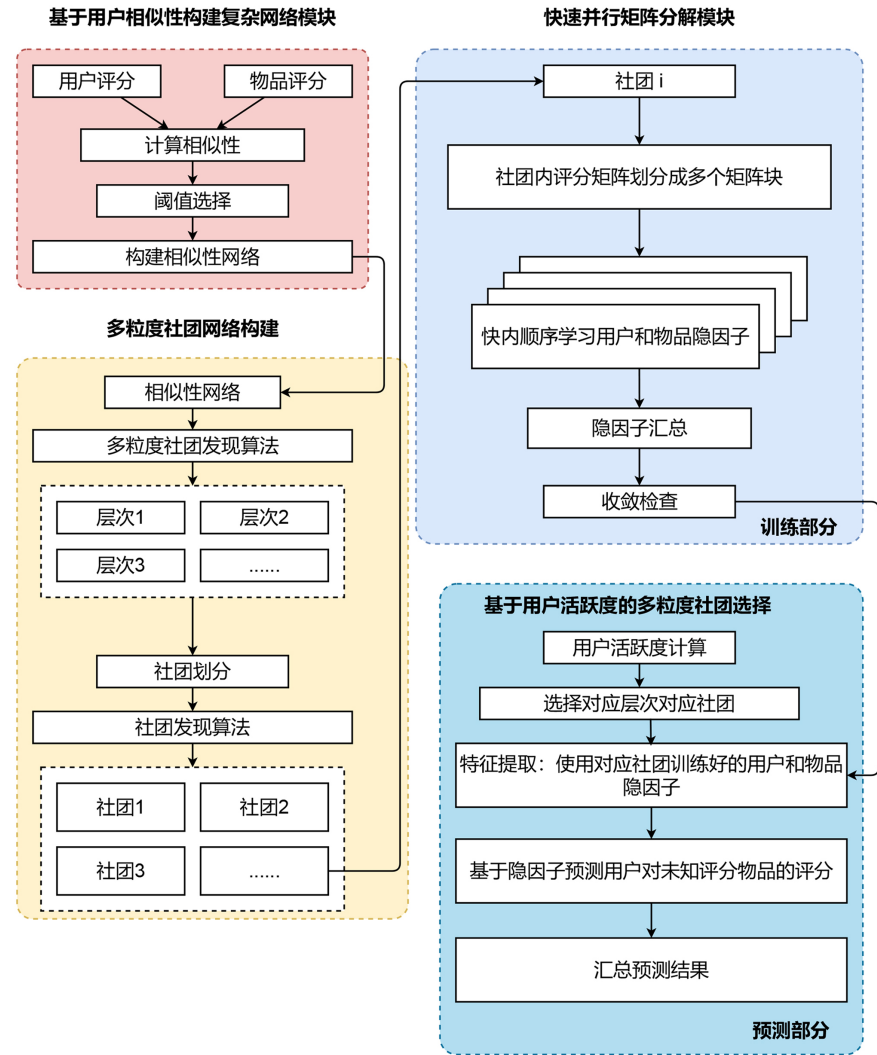
其次,通过递进的分辨率参数和改进的 Louvain 算法进行多粒度社团划分,自适应地发现从粗粒度到细粒度的层次化社团结构,并严格评估每个层次的划分质量,确保合理性和有效性。

然后,在对应粒度层的社团中使用优化的快速并行矩阵分解算法(FPSG)学习用户和物品的隐因子特



征，支持多服务器并行处理，理论上适用于任何矩阵分解类算法。

最后，在预测阶段引入基于用户活跃度的自适应机制，通过计算用户评分数量并结合活跃度阈值，动态分配用户到最适合的社团层次进行评分预测。该方法提升了推荐准确性、计算效率，并缓解了数据稀疏性和冷启动问题，增强了算法的可扩展性。



**Figure 2.** Structure of fast parallel matrix factorization based on user similarity network community features  
**图 2.** 基于用户相似性网络社团特征的快速并行矩阵分解结构图

### 3.3. 基于用户相似性的社团网络发现方法

#### 3.3.1. 构建用户相似性网络

不同用户对相同物品的行为反映了用户之间的相似性。通常，参与评分的物品越多，用户与其他用户之间的相似度就越高。一般来说，两个用户的相似性越大，他们的评分也越趋近一致，因此在预测过程中，具有高相似性的邻居用户对推荐的贡献越大。相反，低相似性的邻居用户对评分预测的影响较小，甚至可能干扰其他用户的评分，从而降低预测的准确性。

度量相似性的方法众多，本文选择了皮尔逊(Pearson)相似性度量方式。其公式如式(1):

$$PCC(u, v) = pcc_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (1)$$

其中  $I_{uv}$  集合代表用户  $u$  和用户  $v$  共同评过分的物品集合,  $r_{ui}$  和  $r_{vi}$  分别表示用户  $u$  和用户  $v$  对物品  $i$  的评分,  $\bar{r}_u$  和  $\bar{r}_v$  分别表示用户  $u$  和用户  $v$  对所有相关物品的平均评分。

根据计算得到的 Pearson 相关系数, 构建加权网络的邻接矩阵  $S$ , 每个元素  $S_{uv}$  表示用户  $u$  和用户  $v$  之间的边权重:

$$S_{uv} = \begin{cases} pcc_{uv}, & \text{if nodes } u \text{ and } v \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

其中  $pcc_{uv}$  是用户  $u$  和用户  $v$  之间的相似值。基于该矩阵, 可以构建反映用户相似性关系的复杂网络, 为后续的社团划分和矩阵分解提供基础。

本文选择过滤掉权重小于等于 0 的边, 仅保留权重大于 0 的边, 一方面为了简化网络结构, 另一方面负权重在用户相似性网络中无法有效反映正相关性和推荐价值。其公式如(3):

$$S_{uv} = \begin{cases} pcc_{uv} & \text{if } pcc_{uv} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

由于构建的相似性网络过于密集, 信息冗杂, 因此需要对其进行简化处理。在这种情况下, 最小生成树作为复杂网络模型的一种基本结构, 可以有效过滤冗余信息, 简化网络结构。因此, 本文通过采用 Prime 算法从相似性网络中提取最小生成树, 其步骤如下:

(1) 选择任意一个起始用户节点, 将其加入到最小生成树(MST)集合  $T$  中, 初始化一个空的边集合  $E_T$ , 用来存储 MST 边, 使用一个最小堆来存储当前可用的边, 按边的权重从大到小排。

(2) 从最小堆中取出权重最小的边  $(u, v)$  加入到 MST 集合中, 即  $T = T \cup \{v\}$  和  $E_T = E_T \cup \{(u, v)\}$ , 将用户节点  $v$  的所有的边加入最小堆中, 只保留连接到树外用户节点的边, 更新最小堆, 继续选择下一条边。

(3) 重复步骤 2, 知道所有用户节点都被包含在 MST 中, 即  $|T| = n$ , 其中  $n$  图中用户总数。

### 3.3.2. 多粒度用户社团层次化分解算法

为了能够更好地挖掘用户间的紧密度, 本文提出了一种基于 Louvain 算法的多粒度用户社团层次化分解算法。该算法分为两个阶段: 首先, 将每个节点分配到能使模块度增益最大的社团中; 其次, 将发现的社团作为新的节点构建新的网络, 重复第一阶段直到模块度不再显著增加。在此基础上, 通过动态调整分辨率参数  $\gamma$ , 实现了对社团尺度的精确控制, 从而实现在同一网络结构上获得不同粒度的社团划分。

Louvain 算法是一种基于模块度优化的社团发现算法, 其核心思想是通过迭代优化模块度  $Q(u, v)$  来发现网络中的社团结构[28]。模块度函数被用来定量描述网络中的社团结构, 衡量网络社团结构划分的质量, 其定义如(4):

$$Q(u, v) = \frac{1}{2m} \sum_{uv} \left( S_{uv} - \frac{k_u k_v}{2m} \right) \delta(C_u, C_v) \quad (4)$$

式中模块度函数  $Q(u, v)$  的值大小反应了社团结构的质量,  $Q(u, v)$  的值越接近 1 说明网络的社团结构越明显。  $S_{uv}$  是邻接矩阵元素, 表示用户节点  $u$  和用户节点  $v$  之间的边权重,  $k_u$  和  $k_v$  分别表示用户节点  $u$  和用户节点  $v$  的度数,  $m$  表示网络中的总边权重,  $C_u$  表示用户节点所在的社团, 若  $u$  和  $v$  在同一个社团中,

则  $\delta(C_u, C_v) = 1$ , 否则  $\delta(C_u, C_v) = 0$ 。其定义如(5):

$$\delta(C_u, C_v) = \begin{cases} 1, & \text{if } C_u = C_v \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

在优化过程中, 算法通过计算节点迁移带来的模块度增益来决定节点的社团归属, 其公式如(6):

$$\Delta Q = \left\{ \frac{1}{2m} \left( \sum_{\text{in}} + k_{u,\text{in}} \right) - \left[ \frac{1}{2m} \left( \sum_{\text{tot}} + k_u \right) \right]^2 \right\} - \left\{ \frac{1}{2m} \sum_{\text{in}} - \left[ \frac{1}{2m_{\text{tot}}} \right]^2 - \left( \frac{k_u}{2m} \right)^2 \right\} \quad (6)$$

其中  $\sum_{\text{in}}$  表示社团  $C_v$  内部边权重之和,  $k_{u,\text{in}}$  表示用户节点  $u$  与社团  $C_v$  中其他用户节点的边权重之和,  $\sum_{\text{tot}}$  表示社团  $C_v$  的所有边权重之和,  $k_u$  表示用户节点  $u$  的度数。

在此基础上, 设计一种适应多层分解策略。首先设定层次数量  $L$ , 然后通过动态调整分辨率参数使得不同层次具有不同的社团粒度, 较大的  $\gamma$  会产生更多的小型社团, 适合活跃度高的用户; 较小的  $\gamma$  值则会产生较少的大型社团, 适合活跃度低的用户。其公式如(7)所示:

$$\gamma_l = \gamma_{\text{base}} + \frac{\gamma_{\text{range}}(l-1)}{L-1}, \quad l = 1, 2, \dots, L \quad (7)$$

其中  $l$  表示当前层次的社团尺度,  $L$  为总层次数,  $\gamma_{\text{base}}$  表示基础分辨率值,  $\gamma_{\text{range}}$  表示分辨率范围大小。

为了确保社团划分的质量, 引入了社团合并约束条件, 确定不同的社团之间是否能够合并, 其公式如(8)所示。当  $\frac{w(C_u, C_v)}{\min(|C_u|, |C_v|)}$  大于阈值  $\theta$  时, 表示两个社团之间平均连接强度足够高, 合并后能够形成一个

个紧密的社团结构, 可以进行合并操作, 否则这两个社团之间联系相对松散, 合并后会降低社团的内部紧密度, 因此社团选择保持独立。

$$\text{Merge}(C_u, C_v) = \begin{cases} \text{True}, & \text{if } \frac{w(C_u, C_v)}{\min(|C_u|, |C_v|)} > \theta \\ \text{False}, & \text{otherwise} \end{cases} \quad (8)$$

其中  $w(C_u, C_v)$  表示社团  $C_u$  和  $C_v$  之间的连接权重总和, 其计算方式(9)所示, 反映的是社团间的连接强度。 $\min(|C_u|, |C_v|)$  表示两个社团规模的较小值,  $|C_u|$  和  $|C_v|$  分别表示社团  $u$  和  $v$  包含的用户数量。

$$w(C_u, C_v) = \sum_{u \in C_u, v \in C_v} A_{uv} \quad (9)$$

在完成社团合并之后, 需要对每个社团进行质量评估, 以确保社团划分符合实际需求, 其公式如(10)所示:

$$\text{Quality}(C) = \frac{\sum_{u,v \in C} A_{uv}}{\sum_{u \in C, v \notin C} A_{uv}} \quad (10)$$

其中  $\sum_{u,v \in C} A_{uv}$  表示社团中所有用户对之间的相似度总和,  $\sum_{u \in C, v \notin C} A_{uv}$  表示社团中用户与社团外用户之间的相

似度总和。设定质量阈值, 对于整体比值低于阈值的社团分析内部结构, 识别关键链接, 考虑是否需要重新划分, 与此同时, 持续监管社团质量变化, 记录质量指标的时间序列, 发现质量下降时及时调整。

其具体算法步骤如下:



(1) 初始化多层次结构。根据预设的层次数量，计算每个层次对应的分辨率参数。这些参数将控制不同层次社团的大小，较小的分辨率形成较大的社团，较大的分辨率形成较小的社团。

(2) 构建用户关系网络。基于用户间的相似度数据，建立初始的用户关系图，其中节点代表用户，边的权重代表用户间的相似度。

(3) 在每个层次上进行社团发现。首先将每个用户视为独立的社团，然后通过计算模块度增益来确定用户节点的最优归属。如果将用户移动到某个邻居社团能够提高整体模块度，则进行移动。

(4) 社团合并优化。检查相邻社团之间的连接强度，当两个社团之间的连接强度除以较小社团的规模超过预设阈值时，将这两个社团合并。这一步骤确保形成的社团具有足够的内部联系。

(5) 社团质量评估。对每个形成的社团进行质量评估，包括计算社团的内部连接强度与外部连接强度的比值，以及社团的密度。这些指标用于判断社团划分的合理性。

### 3.4. 多粒度网络社团划分的快速并行矩阵分解模型

在完成多层网络社团划分后接下来进入基于多层网络社团的快速并行矩阵分解阶段。这个阶段的核心思想是通过社团划分策略，将传统的大规模全局矩阵分解问题转化为多个规模可控、互相独立的社团内部矩阵分解子任务。具体而言，对于每个社团  $C$ ，模型首先构建该社团的局部评分矩阵  $C \in R_{|U_C| \times |L_C|}$ ，其中  $|U_C|$  表示社团  $C$  中的用户数量， $|L_C|$  表示这些用户评分过的物品数量。通过矩阵分解技术，模型将  $R_C$  分解为两个低维潜在特征矩阵：用户特征矩阵  $P \in R^{|U_C| \times k}$  和物品特征矩阵  $Q \in R^{|L_C| \times k}$ ，其中  $k$  表示潜在特征空间的维度  $k < \min(|U_C|, |L_C|)$ 。这种降维映射能够有效捕捉用户的潜在偏好特征和物品的潜在属性特征，为后续的评分预测提供基础。

在矩阵分解过程中，模型通过最小化如下目标函数来学习最优的特征矩阵如公式(11)所示：

$$\min_{P, Q} \sum_{(u, v) \in C_i} \left( (pcc_{uv} - p_u^\top q_v)^2 + \lambda_p \|p_u\|^2 + \lambda_q \|q_v\|^2 \right) \quad (11)$$

该公式由两部分组成，第一部分是预测评分与实际评分之间的平方误差，用于保证模型的预测准确性；第二部分是正则化项，通过控制特征向量的范数来防止模型过拟合。其中  $\lambda$  是正则化参数，用于平衡这两个目标。系统采用随机梯度下降法进行模型优化，在每次迭代中随机选择一个评分样本  $(u, v)$ ，并按照以下规则更新参数：

$$\begin{aligned} p_u &\leftarrow p_u + \alpha (e_{uv} q_v - \lambda_p p_u) \\ q_v &\leftarrow q_v + \alpha (e_{uv} p_u - \lambda_q q_v) \end{aligned} \quad (12)$$

其中  $e_{uv} = pcc_{uv} - p_u^\top q_v$  表示当前预测的误差， $\alpha$  是学习率，用于控制参数更新的步长。通过这种社团内并行矩阵分解的策略，系统实现了推荐计算的高效并行化，同时由于社团内用户具有相似的评分模式，也提高了预测的准确性。这种方法为大规模推荐系统提供了一个既保证质量又提升效率的可行解决方案。

最后，经过上述步骤，更新原来的社团集合，生成了包含补充数据社团集合  $C$ 。

### 3.5. 基于用户活跃度的多粒度社团选择机制

在推荐系统的个性化服务中，用户社团结构的选择直接影响推荐的精确性和覆盖率。完成社团层次化分解后，系统需要通过评估和匹配机制，将目标用户分配到最适合的社团层次，实现个性化适配。这一过程包括三个关键步骤：用户活跃度评估、自适应层次选择和基于多特征的社团匹配。在用户活跃度评估中，模型从用户的历史评分记录中提取评分总数，作为活跃度度量指标，其公式如(13)所示。

$$\text{Activity}(u) = |R_u| \quad (13)$$

其中  $\text{Activity}(u)$  表示用户  $u$  的活跃度评分,  $|R_u|$  表示用户  $u$  在历史数据中的评分总数,  $R_u$  表示用户  $u$  的评分记录集合。

在获得用户活跃度评估结果后, 通过科学合理的分层选择机制, 将用户分配到最适合的社团层次。基于用户的活跃度特征, 系统将用户活跃度划分为三个层次, 分别为低活跃度用户、中等活跃度用户和高等活跃度用户。层次选择的公式如(14)所示。

$$\text{Level}(u) = \begin{cases} L-1, & \text{if } \text{Activity}(u) > \theta_{\text{high}} \\ \lfloor \text{Activity}(u) \cdot (L-1) \rfloor, & \text{if } \theta_{\text{low}} \leq \text{Activity}(u) \leq \theta_{\text{high}} \\ 0, & \text{if } \text{Activity}(u) < \theta_{\text{low}} \end{cases} \quad (14)$$

其中  $\text{Level}(u)$  表示为用户  $u$  选择的社团层次,  $L$  表示社团总层数,  $\text{Activity}(u)$  是用户  $u$  的活跃度得分,  $\theta_{\text{high}}$  为高活跃度阈值,  $\theta_{\text{low}}$  表示低活跃度阈值。当  $\text{Activity}(u)$  超过  $\theta_{\text{high}}$  时模型选择最细粒度的社团层次  $L-1$ , 当用户活跃度  $\text{Activity}(u)$  在低阈值  $\theta_{\text{low}}$  和高阈值  $\theta_{\text{high}}$  之间, 模型根据具体活跃度按比例分配到相应层次, 当  $\text{Activity}(u)$  低于低阈值  $\theta_{\text{low}}$  时, 系统会选择最粗粒度的社团层次 0。

在确定用户所属的社团层次后, 在该层次中找到最匹配的社团。基于多特征的社团匹配通过评估用户与社团之间的相似度和评分行为的重叠程度, 来确定最适合的目标社团。其公式如(15)所示:

$$\text{Match}(u, C) = \beta \cdot pcc(u, C) + (1 - \beta) \cdot \frac{|R_u \cap R_C|}{|R_u \cup R_C|} \quad (15)$$

其中  $\text{Match}(u, C)$  用户  $u$  与社团  $C$  的综合匹配度得分, 一般考虑用户的评分偏好和社团的相似度和该用户对对应物品与社团的重叠度,  $pcc(u, C)$  是用户  $u$  与社团  $C$  的相似度值,  $R_u$  用户  $u$  的评分项目集合,  $R_C$  是社团  $C$  中用户的评分项目集合,  $\beta$  表示相似度权重系数,  $1 - \beta$  为评分重叠度权重系数。

在完成用户与各个社团的匹配度计算后, 模型需要为用户选择最终的目标社团。在用户所属层次的所有社团中, 计算用户与每个社团的匹配度, 并选择匹配度最高的那个社团作为用户的最终归属社团, 其公式如(16)所示:

$$C_{\text{assigned}}(u) = \arg \max_{C \in C_{\text{Level}(u)}} \text{Match}(u, C) \quad (16)$$

式中  $C_{\text{assigned}}(u)$  表示最终分配给用户  $u$  的目标社团,  $\arg \max_{C \in C_{\text{Level}(u)}}$  表示寻找  $\text{Match}(u, C)$  取得最大值时的参数。

## 4. 实验设计

### 4.1. 数据集与实验环境

本文实验采用了四个公开数据集: MovieLens 25M、Movies\_and\_TV、TripAdvisor 和 Yelp。这些数据集分别来自不同的领域, 具体数据如表 2 所示。具体信息如下:

(1) MovieLens 25M: 由 GroupLens Research 实验室提供, 是电影评分数据集。

(2) Movies\_and\_TV: 该数据集是 Amazon Product Review 数据集的一个子集, 包含 7,506 名用户对 7,360 部电影或电视节目的 441,783 条评分记录。

(3) TripAdvisor: 该数据集来源于旅游平台 TripAdvisor, 反映了用户在旅游场景中的偏好和行为模式。

(4) Yelp: 该数据集来自 Yelp 平台, 包含用户对各类商家的评分和评论。

为了提高实验效率, 本文从每个数据集中随机抽取了 3000 名用户的完整评分记录, 作为实验的抽样

数据集。这一抽样策略在降低计算复杂度的同时，确保了数据的代表性和实验结果的可靠性。此外，本文采用五折交叉验证方法，以验证所提算法的科学性和有效性。抽样数据集的统计信息如表 3 所示。

**Table 2.** Complete dataset statistics

**表 2.** 完整数据集统计信息

数据集	用户数量	物品数量	评分数量
MovieLens 25M	162,000	62,000	2500 万
Movies_and_TV	7506	7360	441,783
TripAdvisor	9765	6280	320,023
Yelp	27,147	20,266	1,293,247

**Table 3.** Sample dataset statistics

**表 3.** 抽样数据集统计信息

数据集	用户数量	物品数量	评分数量
MovieLens 25M	3000	36,672	484,422
Movies_and_TV	3000	7360	180,163
TripAdvisor	3000	6280	97,828
Yelp	3000	19,766	142,265

## 4.2. 对比实验

本文开展对比实验的目的包括以下几点：(1) 评估 MGFPMF 模型在适应不同活跃度用户需求方面的表现，特别是其在缓解数据稀疏性和提升推荐精准性方面的效果。(2) 评估 MGFPMF 模型在计算成本方面的表现，重点关注其在大规模数据集上的计算效率和资源利用率。为此，本文选择了推荐系统中广泛应用且研究深入的经典算法作为基准，包括 LIBMF、概率矩阵分解(Probabilistic Matrix Factorization, PMF)以及奇异值分解扩展版(Singular Value Decomposition Plus Plus, SVD++)。这些算法在推荐系统领域具有重要地位，并经过了广泛验证。以下是对这些基准算法的简要介绍：

(1) PMF [29]：一种概率矩阵分解方法。PMF 假设用户 - 物品评分矩阵可以近似表示为两个潜在因子矩阵的乘积，基于高斯噪声假设，通过最大化后验概率来求解潜在因子矩阵。

(2) SVD++ [30]：一种经典 SVD 方法的扩展算法。它在传统 SVD 的基础上增加了对隐式反馈信息的建模能力，从而改进预测性能。

(3) LIBMF [27]：一种优化的并行随机梯度下降算法，能够在多线程环境下高效训练模型。通过优化数据存储结构和计算流程，LIBMF 显著减少了内存访问的开销，提升模型的收敛速度。

## 4.3. 误差分析

平均绝对误差(mean absolute error, MAE)和均方根误差(root mean square error, RMSE)是评价推荐系统预测准确性的两个常用指标，用来衡量预测评级与实际评级的距离。

$$MAE = \frac{1}{|\text{test}|} \sum_{r_{u,i} \in N} |r_{u,i} - r| \quad (19)$$

$$RMSE = \sqrt{\frac{1}{|\text{test}|} \sum_{r_{u,i} \in N} (r_{u,i} - r)^2} \quad (20)$$

其中 $|\text{test}|$ 为测试集的长度； $r$ 为用户的真实评分。MAE 为预测值与实际值的真实误差，RMSE 则是误差平方的平均值的平方根，对大误差更敏感，能更好反映预测偏差大的情况。

根据实验数据分析，MGFPMF 算法在不同数据集上的表现各异。其分析结果如表 4 所示。在 Movielens25m 数据集上，MGFPMF 展现出最佳的推荐准确性，MAE 和 RMSE 均为最低(分别为 0.6356 和 0.8447)。在 Movies\_and\_TV 数据集上，虽然平均预测误差(MAE)最小，但 RMSE 略高于其他算法，说明可能存在较大预测偏差。而在 Yelp 和 TripAdvisor 数据集上，MGFPMF 与 LIBMF 算法表现接近，MAE 略有降低，但 RMSE 则略有升高，提示可能需要优化算法以减少较大误差。总体而言，MGFPMF 算法具备一定的优势，但仍需针对不同数据集的特性进行优化，尤其是在降低较大误差方面。

**Table 4.** MAE and RMSE evaluation results  
**表 4.** MAE、RMSE 评估结果

	Movielens25m		Yelp		TripAdvisor		Movies_and_TV	
	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
LIBMF	0.6403	0.8521	0.8840	<b>1.1364</b>	0.6454	0.8403	0.7433	0.9988
PMF	0.6628	0.8720	0.8870	<b>1.1364</b>	0.6377	0.8285	0.7556	<b>0.9876</b>
SVD++	0.6450	0.8562	0.9036	1.1569	<b>0.6269</b>	<b>0.8166</b>	0.7436	1.0096
<b>MGFPMF</b>	<b>0.6356</b>	<b>0.8447</b>	<b>0.8678</b>	1.1401	0.6290	0.8412	<b>0.7271</b>	1.0165

**4.4. 准确性分析**

在评估推荐系统的性能时，准确率(Accuracy)、精确率(Precision)和召回率(Recall)是常用的分类指标。针对物品推荐场景，混淆矩阵中的各项指标意义如下：真阳性(TP)表示系统成功推荐了用户感兴趣的物品；假阳性(FP)表示系统推荐了用户不感兴趣的物品；假阴性(FN)表示系统未能推荐给用户，但用户实际感兴趣的物品。这些指标共同用于衡量推荐系统的性能。表 5 显示了四种指标在混淆矩阵的情况。

**Table 5.** Classification of preference prediction results  
**表 5.** 分类结果混淆矩阵

用户喜好	系统推荐	系统不推荐
相关	True-Positive (TP)	False-Negative (FN)
不相关	False-Positive (FP)	True-Negative (TN)

准确性(Accuracy)是评估分类模型性能的指标，定义为正确分类的样本数与总样本数之比。在推荐系统领域，准确性体现了系统区分用户感兴趣物品与不感兴趣物品的能力。较高的准确性意味着系统能够有效识别用户偏好，从而提升推荐质量。其公式如(21)所示：

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{21}$$

实验结果如表 6 所示，MGFPMF 在不同数据集上的准确率表现存在差异。在 TripAdvisor 和 Movies\_and\_TV 数据集上，MGFPMF 分别比最差算法 LIBMF 提高 1.46%和 1.18%，比 PMF 提高 0.71%和 0.75%，表现最佳。在 Yelp 数据集上，MGFPMF 也略优于 LIBMF 和 PMF，但优势不明显(提升 1.41%和 0.60%)。然而，在 Movielens25m 数据集上，MGFPMF 的准确率比最佳算法 SVD++低 2.81%，表现不佳，可能需要针对该数据集的特性进一步优化。

**Table 6.** Accuracy evaluation metrics  
**表 6.** Accuracy 评价指标

	Movielens25m	Yelp	TripAdvisor	Movies_and_TV
	Accuracy ↑	Accuracy ↑	Accuracy ↑	Accuracy ↑
LIBMF	0.7050	0.6661	0.7698	0.7678
PMF	0.7107	0.6715	0.7755	0.7711
SVD++	<b>0.7171</b>	0.6542	0.7866	0.7676
<b>MGFPMF</b>	0.6970	<b>0.6755</b>	<b>0.7810</b>	<b>0.7769</b>

在推荐系统中,精确率(Precision)指的是推荐给用户的物品中,用户真正感兴趣的物品所占的比例。该指标用于评估推荐系统推荐结果的质量,体现了推荐相关性。较高的精确率表明系统所推荐的物品更符合用户偏好,从而有助于提升用户满意度。公式如(22)所示

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (22)$$

**Table 7.** Precision evaluation metrics  
**表 7.** Precision 评价指标

	Movielens25m	Yelp	TripAdvisor	Movies_and_TV
	Precision↑	Precision↑	Precision↑	Precision↑
LIBMF	0.8336	0.7504	<b>0.8850</b>	0.8850
PMF	0.8466	0.7568	0.8752	0.8752
SVD++	0.8472	0.7468	0.8830	0.8830
<b>MGFPMF</b>	<b>0.8723</b>	<b>0.7601</b>	0.8271	<b>0.8937</b>

实验结果如表 7 所示, MGFPMF 算法在不同数据集上的精确率表现各有千秋。在 Movielens25m 和 Movies\_and\_TV 数据集上,其精确率分别比最优对比算法提升 2.96% 和 1.21%,表现最佳。在 Yelp 数据集上, MGFPMF 仅略高于对比算法(提升 0.44%)。然而,在 TripAdvisor 数据集上,其精确率比最优算法低 6.54%,表现最差,需进一步优化以适应该数据集的特性。总体而言, MGFPMF 在电影和影视推荐任务中表现突出,但在旅游推荐(TripAdvisor)上的精确率仍有较大提升空间。

在推荐系统中,召回率(Recall)定义为用户感兴趣的物品中,被推荐系统成功推荐的比例。该指标用于评估推荐系统对用户兴趣的覆盖程度,即系统发现并推荐所有相关物品的能力。较高的召回率意味着系统能更全面地满足用户需求。公式如(23)所示:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (23)$$

**Table 8.** Recall evaluation metrics  
**表 8.** Recall 评价指标

	Movielens25m	Yelp	TripAdvisor	Movies_and_TV
	Recall↑	Recall↑	Recall↑	Recall↑
LIBMF	0.8343	0.7504	0.8850	0.8225
PMF	0.8475	0.7571	0.8753	0.8367
SVD++	<b>0.8478</b>	0.7469	0.8830	0.8204
<b>MGFPMF</b>	0.7223	<b>0.7583</b>	<b>0.8933</b>	<b>0.8591</b>



实验数据显示, MGFPMPF 算法在不同数据集上的召回率表现出差异。在 TripAdvisor 和 Movies\_and\_TV 数据集上, 其召回率分别比最优对比算法提升 0.94% 和 4.72%, 表现最佳。在 Yelp 数据集上, MGFPMPF 仅略高于对比算法(提升 0.16%), 优势不明显。然而, 在 Movielens25m 数据集上, 其召回率比最优算法低 14.78%, 表现最差, 未能充分覆盖用户兴趣, 需进一步优化以适应该数据集的特性。总体而言, MGFPMPF 在旅游(TripAdvisor)和影视推荐(Movies\_and\_TV)任务中召回能力较强, 但在电影推荐(Movielens25m)上的召回率仍有较大提升空间, 具体分析如表 8 所示。

#### 4.5. 参数分析

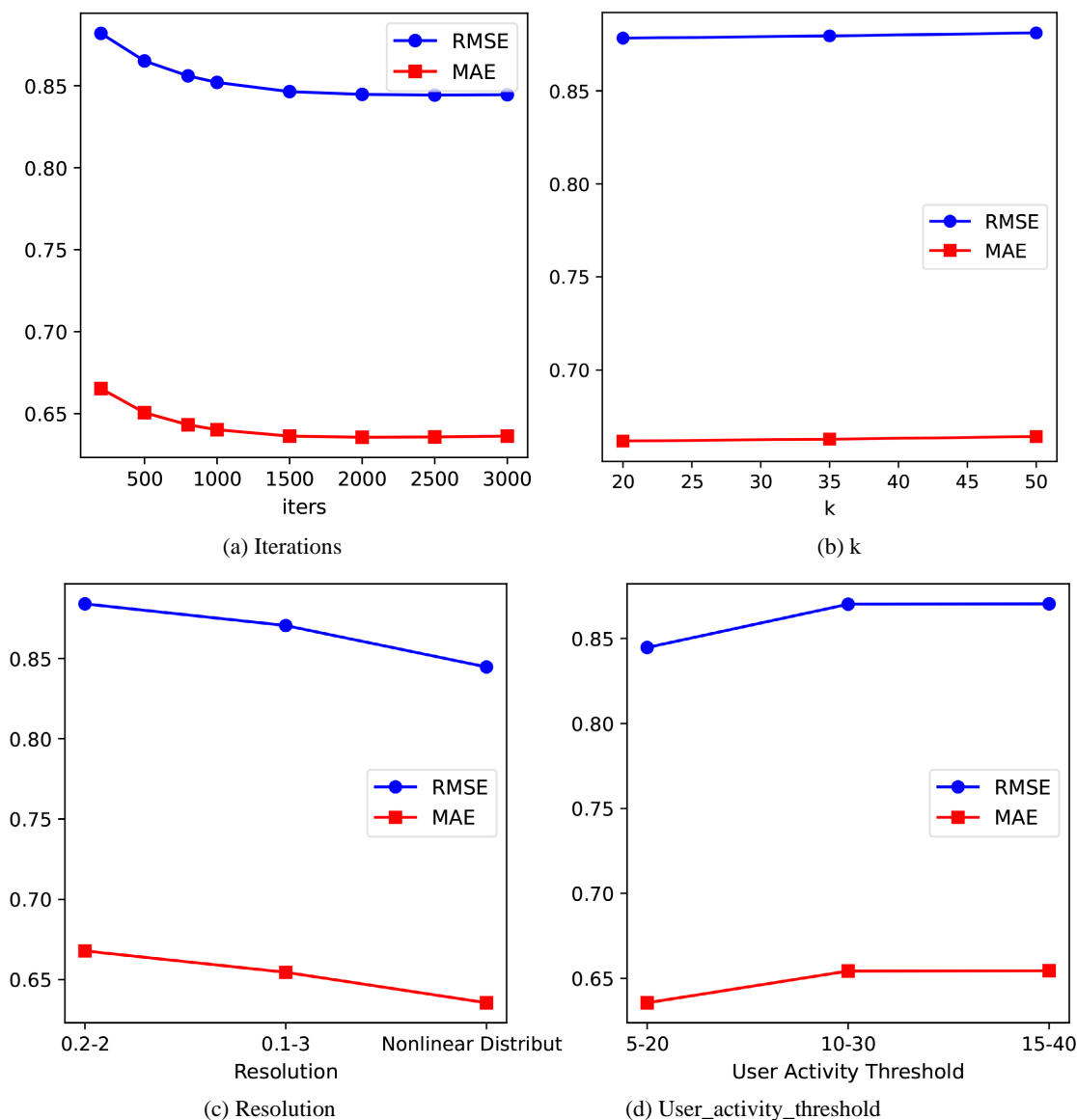


Figure 3. MAE and RMSE of MGFPMPF under different parameters

图 3. MGFPMPF 不同参数下的 MAE 和 RMSE

本文对 MGFPMPF 算法的关键参数进行了详细分析和优化, 所有实验在相同背景下进行, 采用相同编程语言, 并尽量保持与对比方法设置一致。图 3 展示了不同参数下 MGFPMPF 算法在 MovieLens25m 数据

集上的准确度表现。图(a)显示,随着迭代次数增加, RMSE 和 MAE 逐渐下降,模型预测能力提升,迭代 1500 次后误差趋于平稳,因此设置迭代次数为 1500。图(b)表明,潜在因子  $k$  值在 20 到 50 之间时, RMSE 和 MAE 保持稳定,故选择  $k = 35$ 。图(c)显示, Resolution 参数下限从 0.2 缩小到 0.1, 上限从 2 扩展到 3 时, RMSE 从 0.88 降至 0.86, MAE 从 0.67 降至 0.65; 采用非线性分布后, RMSE 进一步降至 0.85, MAE 降至 0.64, 因此采用非线性分布定义 Resolution 参数。图(d)显示, 用户活跃度阈值设为 5~20 时, RMSE 和 MAE 达到最优(RMSE = 0.84, MAE = 0.63), 调整区间后性能小幅下降或保持稳定, 表明 5~20 的阈值区间能更准确识别用户活跃度, 确保模型在可控误差范围内从高频用户中获取有价值信息。所有参数值通过网格搜索确定, 达到最佳实验效果。

#### 4.6. 实验总结

本次实验旨在评估 MGFPMPF 算法在实际推荐场景中的性能。选取 MovieLens 25M、Yelp、TripAdvisor 和 Movies\_and\_TV 四个数据集, 与 LIBMF、PMF 和 SVD++ 等经典算法对比, 验证了 MGFPMPF 在计算效率和推荐准确性上的优势。实验还对 MGFPMPF 的关键参数进行了分析和优化, 以提升算法性能。

实验结果显示, MGFPMPF 在以下方面表现突出:

推荐准确性: 在 MovieLens25M 数据集上, MGFPMPF 的 MAE 和 RMSE 均达到最低值, 预测误差最小; 在 Movies\_and\_TV 数据集上也取得了较好的平均预测误差。

参数优化: 通过网格搜索确定了最佳参数组合。例如, 用户活跃度阈值区间设为 5~20 时, 能更准确识别用户活跃度, 提升推荐效果。

综上, 实验验证了 MGFPMPF 在计算效率和推荐准确性上的优势, 通过合理调整参数, MGFPMPF 能适应不同数据集特点, 为实际推荐系统应用提供支持。

### 5. 总结

本研究旨在解决推荐系统在复杂网络环境下面临的挑战, 包括数据稀疏性导致的推荐质量下降、大规模数据实时更新带来的计算效率问题以及用户兴趣随时间变化难以建模等问题。

为应对这些挑战, 本文创新性地提出了一种融合多粒度社团特征与快速并行矩阵分解(MGFPMPF)的个性化推荐算法。该算法首先构建基于用户相似性的复杂网络, 并通过改进的 Louvain 算法进行多粒度社团划分, 将用户划分为不同层次、不同规模的社团结构。同时, 本文设计了一种基于用户活跃度的自适应选择机制, 能够根据用户行为特征动态地调整所使用的社团层次, 从而在提高推荐准确性的同时, 缓解数据稀疏性和冷启动问题。最终, 通过快速并行矩阵分解技术, 有效利用用户在不同尺度下的偏好特征, 进一步提升了推荐性能。

实验结果表明, MGFPMPF 算法在多个真实数据集上表现出良好的性能, 验证了其在提高推荐质量和可扩展性方面的优势。

展望未来, 我们将继续探索更精细化的社团结构建模方法, 研究用户偏好信息与社团特征的深度融合策略, 并尝试将该算法应用于更多的推荐场景。

### 基金项目

国家自然科学基金项目(61803264)。

### 参考文献

- [1] Li, Y., Liu, K., Satapathy, R., Wang, S. and Cambria, E. (2024) Recent Developments in Recommender Systems: A Survey [Review Article]. *IEEE Computational Intelligence Magazine*, **19**, 78-95.

- <https://doi.org/10.1109/mci.2024.3363984>
- [2] Zhu, J., Dai, Q., Su, L., Ma, R., Liu, J., Cai, G., *et al.* (2022) BARS. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Madrid, 11-15 July 2022, 2912-2923. <https://doi.org/10.1145/3477495.3531723>
  - [3] Zhu, Y., Chen, L., Xiong, D., Chen, S., Du, F., Hao, J., *et al.* (2023) C-AOI: Contour-Based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Long Beach, 6-10 August 2023, 5750-5759. <https://doi.org/10.1145/3580305.3599786>
  - [4] Yu, R., Liu, Q., Ye, Y., Cheng, M., Chen, E. and Ma, J. (2020) Collaborative List-and-Pairwise Filtering from Implicit Feedback. *IEEE Transactions on Knowledge and Data Engineering*, **34**, 2667-2680. <https://doi.org/10.1109/tkde.2020.3016732>
  - [5] Li, H., Li, K., An, J. and Li, K. (2022) An Online and Scalable Model for Generalized Sparse Nonnegative Matrix Factorization in Industrial Applications on Multi-GPU. *IEEE Transactions on Industrial Informatics*, **18**, 437-447. <https://doi.org/10.1109/tii.2019.2896634>
  - [6] Fang, C., Gu, Y., Zhang, W. and Zhang, T. (2022) Convex Formulation of Overparameterized Deep Neural Networks. *IEEE Transactions on Information Theory*, **68**, 5340-5352. <https://doi.org/10.1109/tit.2022.3163341>
  - [7] Yi, B., Shen, X., Liu, H., Zhang, Z., Zhang, W., Liu, S., *et al.* (2019) Deep Matrix Factorization with Implicit Feedback Embedding for Recommendation System. *IEEE Transactions on Industrial Informatics*, **15**, 4591-4601. <https://doi.org/10.1109/tii.2019.2893714>
  - [8] Zhang, X., Liu, H., Chen, X., Zhong, J. and Wang, D. (2020) A Novel Hybrid Deep Recommendation System to Differentiate User's Preference and Item's Attractiveness. *Information Sciences*, **519**, 306-316. <https://doi.org/10.1016/j.ins.2020.01.044>
  - [9] De Almeida, M.P. (2024) Incremental Matrix Factorization for Real-Time Recommender Systems. 1-7.
  - [10] Yang, X. and Zhao, L. (2023) Efficient Incremental Matrix Factorization with Parallel Updates for Large-Scale Recommender Systems. *Information Sciences*, **567**, 92-106.
  - [11] Xu, Z. and Ma, Z. (2023) Incremental Matrix Factorization Based on Time-Sensitive Updates for Personalized Recommendations. *Knowledge-Based Systems*, **263**, Article 107792.
  - [12] Lin, H. and Wei, Q. (2022) Real-Time Personalized Recommendations Using Incremental Matrix Factorization with User Feed-Back. *Neural Computing and Applications*, **34**, 7529-7543.
  - [13] Wang, Q. and Zhang, T. (2022) Causal Matrix Factorization for Personalized Recommendation Systems. *ACM Transactions on Recommender Systems*, **36**, 23-44.
  - [14] Liu, S. and Li, J. (2023) Causal Inference-Based Matrix Factorization for Fair Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, **35**, 1309-1321.
  - [15] Zhou, X. and Li, Y. (2023) Addressing Causal Inference in Matrix Factorization-Based Recommendation Systems. *Proceedings of the 16th International Conference on Web Search and Data Mining*, Singapore, 27 February-3 March 2023, 271-279.
  - [16] Binbusayyis, A. (2022) Deep Embedded Fuzzy Clustering Model for Collaborative Filtering Recommender System. *Intelligent Automation & Soft Computing*, **33**, 501-513. <https://doi.org/10.32604/iasc.2022.022239>
  - [17] Yang, W., Yang, J. and Liu, Y. (2023) Multimodal Optimal Transport Knowledge Distillation for Cross-Domain Recommendation. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, Birmingham, 21-25 October 2023, 2959-2968. <https://doi.org/10.1145/3583780.3614983>
  - [18] Tang, J. and Wang, X. (2023) Adaptive Clustering for Personalized Recommendations in Streaming Environments. *Proceedings of the 2023 IEEE International Conference on Data Mining (ICDM 2023)*, Shanghai, 1-4 December 2023, 283-292.
  - [19] Wang, L. and Zhang, L. (2023) Cross-Domain Personalized Recommendation with Multi-Modal Data Fusion. *Proceedings of the 2023 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2023)*, Vancouver, 17-24 June 2023, 6534-6542.
  - [20] Liu, B. and Sun, G. (2022) Multi-Modal Cross-Domain Collaborative Filtering with Attention-Based Graph Neural Networks. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022)*, Madrid, 11-15 July 2022, 1104-1113.
  - [21] Gupta, M. and Ghosh, R. (2023) Multi-Modal Cross-Domain Learning for Video Recommendation Systems. *Proceedings of the 31st International Conference on Neural Information Processing (ICONIP 2023)*, Changsha, 20-23 November 2023, 47-55.
  - [22] Zhou, X. and Li, Y. (2022) Multi-View Clustering for Multi-Modal Recommendation. *Proceedings of the 31th ACM*

- 
- International Conference on Information and Knowledge Management (CIKM 2022)*, Atlanta, 17-21 October 2022, 2074-2083.
- [23] Clauset, A., Moore, C. and Newman, M.E.J. (2008) Hierarchical Structure and the Prediction of Missing Links in Networks. *Nature*, **453**, 98-101. <https://doi.org/10.1038/nature06830>
  - [24] Newman, M.E.J. (2004) Fast Algorithm for Detecting Community Structure in Networks. *Physical Review E*, **69**, Article 066133. <https://doi.org/10.1103/physreve.69.066133>
  - [25] Blondel, V., Guillaume, J. and Lambiotte, R. (2024) Fast Unfolding of Communities in Large Networks: 15 Years Later. *Journal of Statistical Mechanics: Theory and Experiment*, **2024**, 10R001. <https://doi.org/10.1088/1742-5468/ad6139>
  - [26] Clauset, A., Moore, C. and Newman, M.E.J. (2008) Hierarchical Structure and the Prediction of Missing Links in Networks. *Nature*, **453**, 98-101. <https://doi.org/10.1038/nature06830>
  - [27] Chin, W., Zhuang, Y., Juan, Y. and Lin, C. (2015) A Fast Parallel Stochastic Gradient Method for Matrix Factorization in Shared Memory Systems. *ACM Transactions on Intelligent Systems and Technology*, **6**, 1-24. <https://doi.org/10.1145/2668133>
  - [28] 柯建坤, 许忠好. Louvain 算法与 K 均值聚类算法的比较研究[J]. 应用概率统计, 2022, 38(5): 780-790.
  - [29] Xu, S., Zhuang, H., Sun, F., Wang, S., Wu, T. and Dong, J. (2021) Recommendation Algorithm of Probabilistic Matrix Factorization Based on Directed Trust. *Computers & Electrical Engineering*, **93**, Article 107206. <https://doi.org/10.1016/j.compeleceng.2021.107206>
  - [30] Behera, G., Mohapatra, R.K. and Bhoi, A.K. (2023) A Mixed Collaborative Recommender System Using Singular Value Decomposition and Item Similarity. In: Udgata, S.K., Sethi, S. and Gao, X.Z., Eds., *Intelligent Systems*, Springer Nature, 259-267. [https://doi.org/10.1007/978-981-99-3932-9\\_23](https://doi.org/10.1007/978-981-99-3932-9_23)