

融合画像约束和潜在特征的深度推荐算法

艾 均, 柏光耀, 苏 湛, 马莞言

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15日; 录用日期: 2025年4月8日; 发布日期: 2025年4月17日

摘要

基于深度学习的推荐算法已成为推荐系统领域的研究趋势。然而, 大多数现有工作仅考虑单一的用户与物品交互数据, 限制了算法的预测性能。本文提出一种画像约束的编码方式, 并融合隐因子模型中的潜在特征, 丰富了推荐算法的输入以提升评分预测的准确性。该算法利用矩阵分解得到潜在特征初始化用户与物品的嵌入, 然后通过线性注意力机制增强模型对画像特征的敏感度, 最后结合深度神经网络进行评分预测。通过本文算法与其他基线算法在MovieLens与Netflix数据集上进行对比, 该算法与基线算法相比显著提高了评分预测的精度, 并在推荐列表排序性能等方面表现出色。本文的研究揭示了加入用户与物品的画像约束和潜在特征, 可以有效提升推荐系统的性能。

关键词

推荐系统, 深度学习, 画像约束, 隐因子模型, 线性注意力

A Deep Recommendation Algorithm Integrating Portrait Constraint and Latent Feature

Jun Ai, Guangyao Bai, Zhan Su, Wanyan Ma

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

Received: Mar. 15th, 2025; accepted: Apr. 8th, 2025; published: Apr. 17th, 2025

Abstract

Recommendation algorithms based on deep learning have emerged as a prominent research direction in the field of recommender systems. However, most existing studies focus primarily on single user-item interaction data, which limits the prediction performance of these models. This article

proposes a method for encoding portrait constraints and integrates potential features from hidden factor models, increasing the input of recommendation algorithms to improve the accuracy of rating prediction. This algorithm utilizes matrix decomposition to initialize the embedding of latent features between users and items, and then enhances the sensitivity of the model to portrait features through linear attention mechanism. Finally, it combines deep neural networks for rating prediction. By comparing the algorithm proposed in this article with other baseline algorithms on the MovieLens and Netflix datasets, it was found that this algorithm significantly improved the accuracy of rating prediction compared to the baseline algorithm, and performed well in terms of recommendation list ranking performance. This study reveals that incorporating portrait constraints and latent features of users and items can effectively improve the performance of recommendation systems.

Keywords

Recommender System, Deep Learning, Portrait Constraint, Latent Model, Linear Attention Mechanism

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

推荐系统旨在为用户提供个性化的产品或服务推荐，以解决日益严重的信息过载问题[1]。推荐系统通过项目、用户以及项目与用户之间的交互的相关信息预测用户对项目的兴趣来向特定用户或企业推荐最合适的产品或服务，并提高用户的满意度和忠诚度[2]。推荐系统能在海量数据中自动帮助用户找到可能偏好的信息，产生了难以估量的商业价值，也获得了学术界和工业界的广泛关注[3]。

推荐算法的发展呈现出以下几个主要趋势：基于隐因子模型的推荐算法、基于用户画像的推荐算法、基于深度学习的推荐算法。

基于隐因子模型的推荐算法。矩阵分解是隐因子模型中常用且知名的方法之一[4]。它能够有效地将用户和物品的高维交互数据分解成低维的用户和物品的特征表示，从而提高推荐的效果和效率。常见的矩阵分解算法如 SVD++ [5]在 SVD 基础上，增加了对用户行为的考虑，特别是用户与物品之间的隐式反馈(如用户浏览过某物品但未评分的影响)。Funk-SVD [6]旨在通过低秩矩阵分解来预测用户对未评分物品的评分，从而实现个性化推荐。

基于用户画像的推荐算法。用户画像基于用户的行为数据、人口统计信息、兴趣偏好等多种维度的数据，综合分析和描绘出的一个用户的特征[7]。周[8]等融合多源数据构建用户画像，作为外部信息融入链路预测中，提高了算法的精度。Triyanna [9]等将用户的个人信息(年龄、性别、职业、地理位置)作为用户的基本画像，使用 K-Means 方法对用户进行聚类，实验结果显示，每个类中的 MAE 和 RMSE 都比聚类处理前要低。

基于深度学习的推荐算法。近年来，深度学习迅速发展，并在自然语言处理、图像处理等多个领域取得了显著成果[10]。Tegene [11]等将现有的深度学习方法与隐因子模型结合，通过嵌入技术生成的稠密用户和物品向量映射到两个神经网络架构中，从而预测评分。Lee [12]等通过嵌入层将用户和物品的内容信息转化为稠密特征向量，利用交叉卷积滤波器捕捉用户和物品特征之间的复杂交互关系，并通过全局池化技术减轻过拟合，从而实现有效的评分预测和个性化推荐。Dien 等[13]提出了基于深度矩阵分解的

学习资源和课程推荐方法，在 DMF 中，用户和学习资源两个嵌入特征被连接在一起，作为神经网络的输入，通过一个输出层来预测评分。

推荐系统发展至今仍面临一些问题：1) 大多数推荐算法仅依赖用户对物品的显式反馈，如用户 - 物品评分数据，往往忽视了用户的兴趣或隐式偏好，导致推荐效果难以达到最佳。2) 如何从复杂的用户 - 物品交互数据中有效提取并利用用户与物品的特征，是提升个性化推荐算法准确性的重要挑战。3) 当遇到新用户或新物品时，由于缺乏历史数据，推荐系统难以提供准确的推荐，进而面临冷启动问题。针对如何利用用户的兴趣或隐式偏好以及如何从复杂的交互数据中提取并利用用户和物品的特征，本文提出一种融合画像约束和潜在特征的深度推荐算法(Deep Portrait Feature and Latent Embedding Algorithm, DPFLE)。

2. 相关工作

LIBMF [14]是专门用于大规模数据的矩阵分解算法，它可以利用磁盘来缓存训练矩阵，并且能够在多核机器上并行计算，能够在处理非常大的用户 - 物品评分矩阵时保持高效的计算性能。因此，本文使用 LIBMF 对用户 - 物品评分矩阵进行处理，以提取用户以及物品的潜在特征，使用预先计算的潜在特征初始化算法中用户与物品的嵌入层。

注意力机制使模型能够根据输入数据中各部分的重要性动态调整关注点，从而更有效地捕捉关键信息[15]。DeepICF [16]通过神经网络学习物品之间的特征，Cheng [17]等提出了特征级别的注意力方法并应用在 DeepICF 上，以提高其性能。Zhai [18]等使用点乘的方式来实现自注意力模块的计算，包括 AFT-Full、AFT-Simple、AFT-Local，其中 AFT-Simple 的时间和空间复杂度均为线性级别。本文利用线性注意力来捕捉输入数据中不同特征之间的相关性，减少计算复杂度的同时提升模型的性能。

推荐系统中的主要问题在于用户和物品之间的非线性交互以及大量的数据，这使得特征学习变得困难，在神经协同过滤(Neural Collaborative Filtering, NCF)中，通过神经网络隐式地学习了用户和物品之间的非线性特征表示[19]。利用神经网络能够捕捉用户和物品之间复杂的非线性关系，相较于传统的推荐算法能够实现更加个性化和精准的推荐[20]。MCFSAE [21]采用 One-Hot 编码对用户和物品的 ID 进行特征表示，并将其与用户 - 物品评分矩阵相结合，构建输入向量，进而通过神经网络模型实现评分预测。然而，该方法在处理大规模用户和物品数据时，由于用户和物品数量的增加会导致输入向量的维度急剧膨胀，从而面临维度灾难问题。DualDeepMF [22]利用双正则化矩阵分解获得用户和物品的潜在表示，之后利用神经网络的非线性建模能力来缓解矩阵分解中的线性点积操作而导致的模型预测能力受限问题。DELCR [11]通过嵌入技术提取用户与物品的潜在特征表示，并采用双塔结构对用户和物品分别进行特征学习，最后利用点积进行评分预测。

大多数基于深度学习的推荐算法仅依赖于用户 - 物品评分数据，且普遍存在可解释性不足的问题，本文提出的 DPFLE 使用矩阵分解初始化用户和物品的嵌入向量，利用物品标签信息构建细粒度的画像向量，并通过线性注意力机制增强模型对画像特征的敏感度，丰富算法输入的同时还改善了算法的可解释性。

3. 算法设计

本文提出的 DPFLE 是由用户以及物品的潜在特征、用户以及物品的画像约束、基于深度学习的评分预测这三个部分组成，整体算法如图 1 所示。首先对用户 - 物品评分矩阵，使用矩阵分解技术，得到用户和物品的潜在特征，利用预先计算的潜在特征来初始化用户嵌入层以及物品嵌入层。随后，利用物品的标签信息并结合评分信息计算出用户和物品的画像向量，经过 `portrait_encoder` 层编码得到用户画像约

束以及物品画像约束，对这个四个特征进行融合，之后利用深度神经网络进行评分预测。

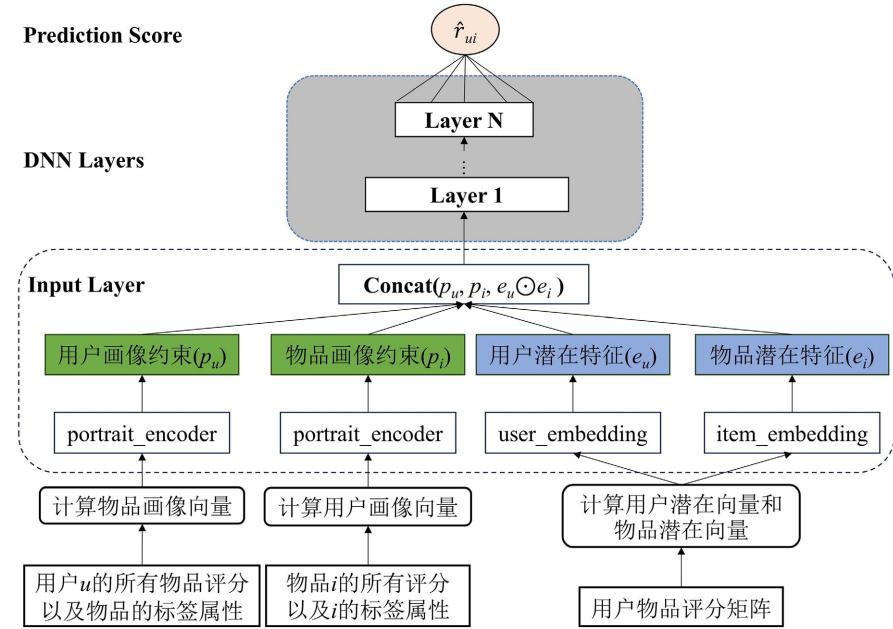


Figure 1. Schematic diagram of DPFLE algorithm design

图 1. DPFLE 算法设计示意图

3.1. 用户以及物品的潜在特征

LIBMF 的目标函数如公式(1)所示，其中 R_{ui} 为真实评分， P_u 和 Q_i 为用户与物品的潜在向量，并使用 L2 正则化系数防止过拟合。

$$\min_{P, Q} \sum_{(u, i) \in \Omega} (R_{ui} - P_u^T Q_i)^2 + \lambda (\|P_u\|^2 + \|Q_i\|^2) \quad (1)$$

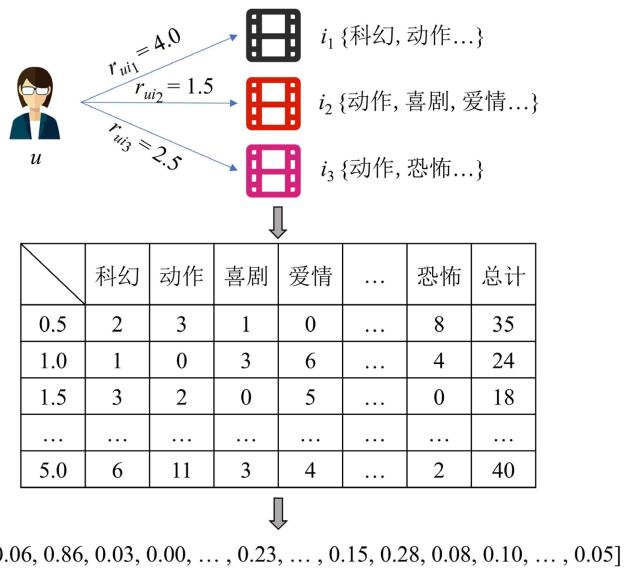
本文利用 LIBMF 对用户物品评分矩阵进行分解，先最小化 LIBMF 的目标函数，以提取用户以及物品的潜在向量。随后，将用户和物品按照 ID 嵌入到向量空间中，利用预先计算的用户与物品的潜在向量初始化用户与物品的嵌入层。在后续中，本文均以 e_u 和 e_i 来表示用户以及物品的潜在特征。

3.2. 用户以及物品的画像约束

在电影推荐领域，电影标签作为电影的抽象表示，蕴含了丰富的用户隐式信息，合理利用电影标签数据，可以有效提升推荐算法的性能，具有重要的应用价值[23]。

假设用户评分区间为 $S = \{0.5, 1, 1.5, \dots, 4.5, 5\}$ ，评分的间隔为 0.5，假设物品的标签集合为 L ，并且一共有 10 个标签，即 $L = \{l_1, l_2, \dots, l_9, l_{10}\}$ 。在本文的实验中，物品的标签即为电影的类型，例如动作片、爱情片、恐怖片、科幻片等等。通过统计用户对不同标签的物品在不同评分下的评分次数，并除以相同评分下的评分总次数最后进行拼接操作得到用户的画像向量。具体过程如图 2 所示。通过相同的方式，统计物品的各标签属性在不同评分下被不同用户评分的次数，并除以该物品在该评分下被用户打分的总次数，可以得到物品的画像向量。

本文使用线性级别的注意力机制 AFT-Simple 作为注意力层来对输入数据进行加权计算。对于输入 X ，通过线性变换得到 $Q = XW_Q$ ， $K = XW_K$ ， $V = XW_V$ ，之后如公式所示，



$$p_u = [0.06, 0.86, 0.03, 0.00, \dots, 0.23, \dots, 0.15, 0.28, 0.08, 0.10, \dots, 0.05]$$

Figure 2. Calculate user's portrait diagram

图 2. 计算用户画像示意图

$$Y_t = \sigma_q(Q_t) \odot \frac{\sum_{t=1}^T \exp(K_t) \odot V_t}{\sum_{t=1}^T \exp(K_t)} \quad (2)$$

其中 Y_t 为经过线性注意力计算的输出结果, σ_q 表示激活函数, 通常是 Sigmoid 函数。通过这个线性注意力层的计算, 模型能够自动分配权重, 捕捉对预测结果影响较大的特征组合, 提升模型对重要特征的敏感度。

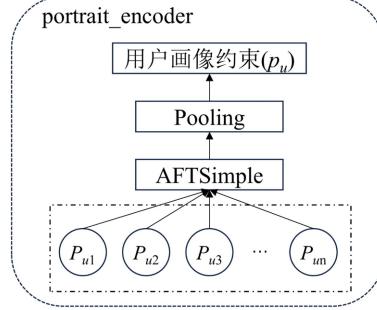


Figure 3. Schematic diagram of user's

portrait constraint encoding process

图 3. 用户画像约束编码过程示意图

假设用户 u 的所有评分物品为 I_{ui} ($i=1, 2, \dots, n$), 计算各物品的画像向量, 记为 $P_{u1}, P_{u2}, \dots, P_{un}$ 并输入到 AFT-Simple 注意力模块中, 得到加权后的中间特征表示, 最后通过全局平均池化操作(Pooling)将加权后的特征表示压缩为全局特征, 得到用户的画像约束表示 p_u , 如图 3 所示。同样地, 对于物品 i 可取其所有的用户评分, 并计算各用户的画像向量, 通过 portrait_encoder 得到物品的画像约束表示 p_i 。

3.3. 基于深度学习的评分预测

本文利用深度神经网络, 将上述已得到画像约束与潜在特征作为神经网络的输入数据, 从而预测用

户对物品的评分。将用户与物品的潜在特征进行逐元素点乘，再与用户和物品的画像约束进行拼接，最后得到深度神经网络的输入 x_0 ，具体如公式(3)所示，

$$\begin{cases} x_0 = \text{Concat}(p_u, p_i, e_u \odot e_i) \\ x_1 = \text{ReLU}(W_1 x_0 + b_1) \\ x_2 = \text{ReLU}(W_2 x_1 + b_2) \\ \dots \\ x_N = \text{ReLU}(W_N x_{N-1} + b_N) \end{cases} \quad (3)$$

其中， \odot 表示逐元素点乘， W_1, W_2, \dots, W_N 和 b_1, b_2, \dots, b_N 分别表示每个隐藏层的权重矩阵和偏置向量。

x_1, x_2, \dots, x_N 表示每个隐藏层的输出，在本文的实验中使用 ReLU 作为每个隐藏层之间的激活函数。

最后，本文使用一个简单的线性回归模型将 x_N 直接映射到预测评分上，如公式(4)所示，

$$\hat{r}_{ui} = z^T x_N + b_z \quad (4)$$

其中 \hat{r}_{ui} 表示用户 u 对物品 i 的预测评分， z 和 b 分别表示预测层的权重向量和偏置。

为了优化模型参数以预测显示评分，本文选择了均方误差损失函数如公式(5)所示，因为它能够有效衡量预测评分与真实评分之间的差异，并适用于这种点对点的回归任务[24]。

$$\text{Loss} = \min \left[\sum_{u,i} (\hat{r}_{ui} - r_{ui})^2 + \lambda W^2 + \alpha b^2 \right] \quad (5)$$

其中， λ 和 α 分别表示隐藏层中权重 W 和偏置 b 的 L2 正则化系数。

4. 实验设计与分析

4.1. 数据集

本文使用经典数据集 MovieLens 以及 Netflix 来验证提出的方法。MovieLens25M 记录了 16 万位用户对 6 万部电影的 2500 万余条评分数据，评分的范围为 0.5 到 5，间隔为 5，电影的类型有动作片、恐怖片、喜剧片等等，一共有 19 种类型。Netflix 记录了 48 万位用户对 1 万余部电影的 10000 万余条评分数据，评分的范围为 1 到 5，间隔为 1。本文分别从两个数据集中随机选取 1500 位用户的所有数据进行实验，以节省计算资源和时间。需要注意的是，由于 Netflix 数据集中不包含电影类型的信息，因此本文通过电影标题在 MovieLens 数据集中查找其相应的类型。本文将数据集按照 90%，10% 的比例将数据集划分为训练集和测试集，每个算法仅基于训练集预测每个测试集中的所有评分，并进行折十验证，最终结果取所有实验的平均值。

4.2. 评价指标

本文使用平均绝对误差(Mean Absolute Error, MAE) [25]、均方根误差(Root Mean Square Error, RMSE) [26] 来评估评分预测的误差。MAE 反映了算法预测评分与用户实际评分之间的偏差，RMSE 表示预测用户评分与实际评分之间偏差的均方根值。它们的定义如公式(6)和(7)所示，

$$\text{MAE} = \frac{\sum_{i \in \text{Test}} |r_{ui} - \hat{r}_{ui}|}{|\text{Test}|} \quad (6)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i \in \text{Test}} (r_{ui} - \hat{r}_{ui})^2}{|\text{Test}|}} \quad (7)$$

在进行 Top-K 推荐时, 仅预测未知的评分是不够的, 推荐系统还需要根据预测的结果过滤出用户喜欢的项目并进行排序, 从而生成推荐列表。在本文的工作中使用归一化折损累计增益(Normalized Discounted Cumulative Gain, NDCG) [27] 来评估推荐列表的排序性能, 其公式如(8)和(9)所示。

$$DCG = \sum_{i=1}^b R_i + \sum_{i=b+1}^N \frac{R_i}{\log_b r_i} \quad (8)$$

$$NDCG = \frac{\sum_{u=1}^m \frac{DCG}{\max(DCG_u)}}{m} \quad (9)$$

其中, b 为影响力因子, 在本实验中设置为 2。 N 表示推荐列表中项目个数。 R_i 为 1 表示用户喜欢某个项目, 为 0 则表示用户不喜欢某个项目, 该指标越大说明排序性能越好。

推荐列表中项目度值的标准差与项目度值的均值的比率可以用来衡量推荐列表的多样性, 即推荐列表中包含了哪些受欢迎的项目和小众的项目。多样性指标(Diversity) [28]如公式(10)所示。

$$Diversity = \frac{\sum_{l \in L} \left(\frac{\sigma(l)}{Ave(l)} \right)}{|L|} \quad (10)$$

精确率(Precision)、准确率(Accuracy)和召回率(Recall)通常用来衡量推荐算法的分类性能, 其公式如(11)、(12)和(13)所示。

$$Precision = \frac{FP}{TP + FP} \quad (11)$$

$$Recall = \frac{FP}{TP + FN} \quad (12)$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (13)$$

4.3. 基线算法

为了评估本文提出的 DPFLE 的性能, 本文将其与其他基线算法进行了上述评价指标的比较。所有算法均取其论文中的最优指标。

- 1) LIBM 通过矩阵分解技术, 将用户 - 物品评分矩阵分解为两个低维矩阵, 从而预测未知评分。
- 2) ScoR [29]把物品和用户当成节点, 将合成坐标分配给节点以便用户和物品间的距离可以准确预测用户对某个物品偏好。
- 3) DSS [30]通过计算两个项目之间的评分差异来表示用户间的距离, 并通过皮尔逊系数计算用户间的相似度, 通过相似度大于设定的阈值来选取用户的邻居从而预测评分。
- 4) MCFSAE [21]对用户以及物品的 id 进行独热编码, 结合用户 - 物品评分矩阵, 利用堆叠自编码器(Stacked Autoencoder, SAE)获取输入数据的低维特征表示, 最后使用 Softmax 对未知评分进行分类预测。
- 5) DPFLE 为本文提出的方法, 本文将潜在特征的大小 k 设置为 300, 模型的优化器为 Adam, 正则化系数设置为 0.01, 学习率设置为 0.0001, 批次大小设置为 128。在深度神经网络中, 文本设置了两个隐藏层, 隐藏层中的因子分别为 300、100。

4.4. 实验结果与分析

4.4.1. 各指标对比结果

在 MovieLens 数据集和 Netflix 数据集中的 MAE 和 RMSE 结果对比如表 1 所示, 其中表现最佳的值

以黑体加粗标出, 后续也用此形式表示最优值。DPFLE 的 MAE 值和 RMSE 值在两个数据集上相较于其他对比算法均达到了最优。在 MovieLens 数据集上, DPFLE 的 MAE 值和 RMSE 值分别为 0.6387 和 0.8418, 相较于其他对比算法分别平均降低了 6.26% 和 7.97%。在 Netflix 数据集上, DPFLE 的 MAE 值和 RMSE 值分别为 0.6845 和 0.8827, 相较于其他对比算法分别平均降低了 5.91% 和 7.90%。实验结果显示, DPFLE 相比于其他基线算法拥有更好的预测精度, 预测出的评分与真实评分之间的误差更小。

Table 1. Comparison results of MAE and RMSE for each algorithm
表 1. 各算法在两个数据集中的 MAE 和 RMSE 的对比结果

| Methods | MovieLens | | Netflix | |
|---------|---------------|---------------|---------------|---------------|
| | MAE | RMSE | MAE | RMSE |
| LIBMF | 0.6678 | 0.8614 | 0.7109 | 0.9062 |
| SCoR | 0.7391 | 0.9606 | 0.8014 | 0.9911 |
| DSS | 0.6639 | 0.8685 | 0.7149 | 0.9155 |
| MCFSAE | 0.6604 | 0.9807 | 0.6920 | 1.0320 |
| DPFLE | 0.6387 | 0.8418 | 0.6845 | 0.8827 |

表 2 显示了 NDCG 和 Diversity 在两个数据集上的结果对比。NDCG 和 Diversity 分别用来评估推荐列表的排序质量和推荐列表的多样性。其中, NDCG 在两个数据集上都取得了最好的效果, 分别为 0.8198 和 0.8154, 相较于其他算法分别平均提升了 3.41% 和 1.96%。结果显示了 DPFLE 在推荐列表的排序性能上表现更优。但是 Diversity 表现相对不佳, 在两个数据集中均排名第三, 与排名第一的 SCoR 平均相差 9.31%, 与排名第二的 MCFSAE 平均相差 4.33%。

Table 2. Comparison results of NDCG and Diversity for each algorithm
表 2. 各算法在两个数据集中的 NDCG 和 Diversity 的对比结果

| Methods | MovieLens | | Netflix | |
|---------|---------------|---------------|---------------|---------------|
| | NDCG | Diversity | NDCG | Diversity |
| LIBMF | 0.7821 | 0.5594 | 0.7889 | 0.5259 |
| SCoR | 0.8083 | 0.6961 | 0.8102 | 0.6191 |
| DSS | 0.7926 | 0.6065 | 0.8016 | 0.5646 |
| MCFSAE | 0.7887 | 0.6532 | 0.7988 | 0.5925 |
| DPFLE | 0.8198 | 0.6254 | 0.8154 | 0.5667 |

表 3 显示了各个算法的 Accuracy, Precision 以及 Recall 在两个数据集上的表现。其中, DPFLE 在两个数据集上的 Accuracy 值均排名第一, 排名第二均为 MCFSAE, 相较其他算法在两个数据集中分别提升了 3.54% 和 4.69%, 这表示 DPFLE 能够更好地识别用户感兴趣和不感兴趣的物品。DPFLE 在两个数据集中的 Precision 均排名第二, 在两个数据集中分别低于排名第一的 LIBMF 3.52% 和 3.10%, 说明 DPFLE 为用户推荐其感兴趣的物品时, 通常有着不错的表现。MCFSAE 在两个数据集中的 Recall 表现最好, 均排名第一, DPFLE 次之, 说明 DPFLE 能够成功识别出大部分用户感兴趣的物品。

Table 3. Comparison results of classification metrics for each algorithm
表 3. 各算法在两个数据集中分类指标的对比结果

| Methods | MovieLens | | | Netflix | | |
|---------|---------------|---------------|---------------|---------------|---------------|---------------|
| | Accuracy | Precision | Recall | Accuracy | Precision | Recall |
| LIBMF | 0.6572 | 0.7178 | 0.6027 | 0.6499 | 0.7056 | 0.5741 |
| SCoR | 0.6153 | 0.6843 | 0.6735 | 0.5906 | 0.6756 | 0.6528 |
| DSS | 0.6551 | 0.6841 | 0.6727 | 0.6408 | 0.6611 | 0.6511 |
| MCFSAE | 0.6660 | 0.6663 | 0.7665 | 0.6630 | 0.6492 | 0.7213 |
| DPFLE | 0.6707 | 0.6925 | 0.7030 | 0.6646 | 0.6837 | 0.6748 |

在 MovieLens 与 Netflix 数据集中, 各算法的总体实验结果表现如图 4, 其中本文提出的 DPFLE 在 MAE、RMSE、NDCG、Accuracy 这四个指标上均取得了最好的效果。由上述实验结果可知, 将画像约束与潜在特征相融合并利用深度学习进行评分预测, 能够有效降低推荐系统中的误差, 并且在推荐列表的排序性能方面也有显著提升。然而, 在推荐列表的多样性方面, 本文提出的 DPFLE 表现并不出色。由于将用户与物品的画像约束加入算法中, 推荐结果往往集中于用户已经喜欢的物品, 而忽视了其他潜在的、有趣的选项。

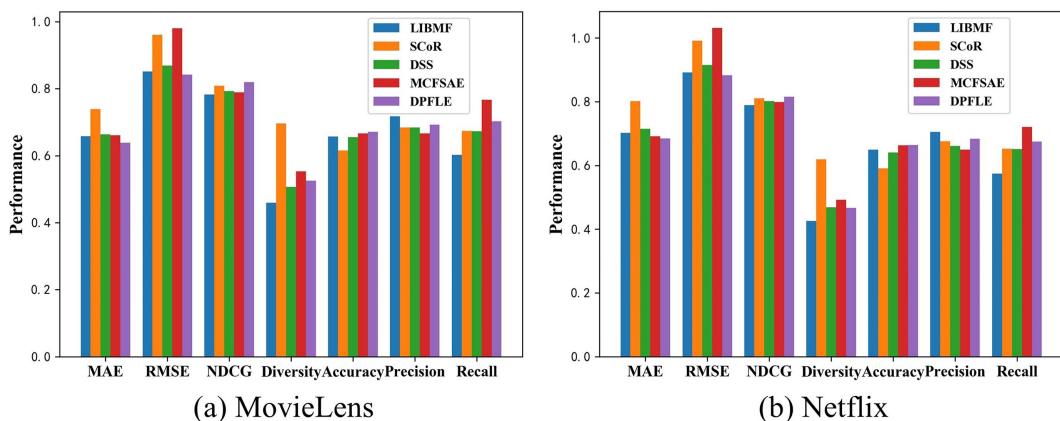


Figure 4. Overall experimental comparison of algorithms
图 4. 各算法在两个数据集中的总体实验对比图

4.4.2. 实验分析

为了探究潜在特征的大小对实验的影响, 本文将 k 分别设置为 50、100、200、300, 并在两个数据集上进行实验。表 4 显示了在不同的潜在特征大小下, DPFLE 在两个数据集中的 MAE 和 RMSE 的表现。当 k 大于 300 时, MAE 和 RMSE 的实验结果变化非常小, 但是算法的执行时间会更长。为了节省计算资源和时间, 本文没有过多探究更大的 k 对实验的影响。

图 5 显示了用户侧和物品侧的注意力权重矩阵示意图, 在 AFTSimple 中, 对输入进行线性变化后得到 K 矩阵, 再进行 Softmax 操作得到每个维度上的注意力分布, 不同维度的注意力分数互相独立计算。全局注意力信息揭示了用户对不同物品的关注程度, 反应出用户在偏好选择上的显著差异。通过动态地为不同用户及物品特征分配自适应权重, 注意力机制能够有效捕捉模型对不同特征的敏感程度, 从而提升个性化推荐算法的准确性。

Table 4. Comparison of MAE and RMSE with different hyperparameters**表 4.** 不同超参数在两个数据集中的 MAE 和 RMSE 对比

| Numbers of k | MovieLens | | | Netflix | |
|----------------|---------------|---------------|---------------|---------------|--|
| | MAE | RMSE | MAE | RMSE | |
| 50 | 0.6524 | 0.8583 | 0.6986 | 0.9010 | |
| 100 | 0.6461 | 0.8522 | 0.6949 | 0.8941 | |
| 200 | 0.6412 | 0.8482 | 0.6881 | 0.8854 | |
| 300 | 0.6387 | 0.8418 | 0.6845 | 0.8827 | |

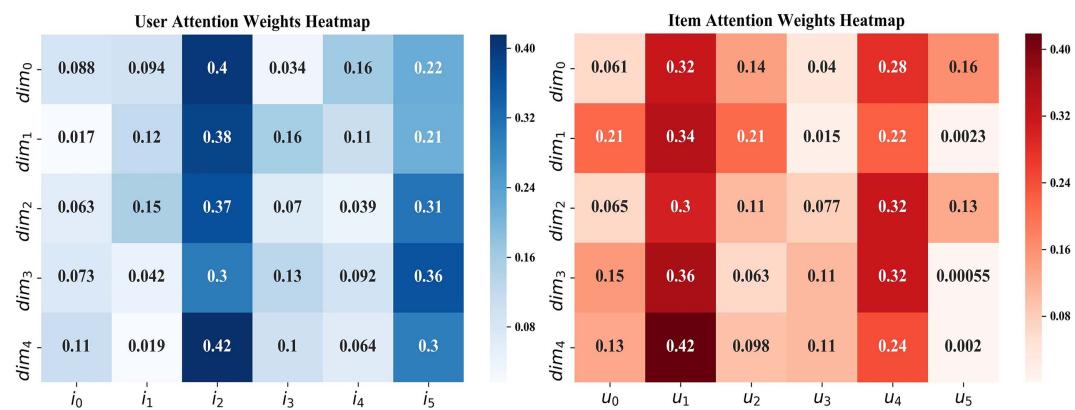
**Figure 5.** Heatmap of attention weight matrices on the user and item**图 5.** 用户侧与物品侧注意力权重矩阵热力图

表 5 显示了去掉 AFTSimple 注意力模块后与原始算法的比较，其中 DPFLE-AFT 表示去除注意力模块后的模型，经比较可以得出 DPFLE 在两个数据集中的 MAE 平均降低了 2.11%，RMSE 平均降低了 1.16%，说明通过线性注意力机制对画像特征进行加权计算能够更好地捕捉用户和物品之间复杂的关系并提升模型的预测精度。

Table 5. Comparison results of attention module ablation**表 5.** 注意力模块消融实验对比结果

| Methods | MovieLens | | Netflix | |
|-----------|---------------|---------------|---------------|---------------|
| | MAE | RMSE | MAE | RMSE |
| DPFLE-AFT | 0.6513 | 0.8506 | 0.7004 | 0.8941 |
| DPFLE | 0.6387 | 0.8418 | 0.6845 | 0.8827 |

5. 总结与展望

针对如何利用用户的兴趣或隐式偏好以及如何从复杂的交互数据中提取并利用用户和物品的特征，本文提出了一种画像约束的编码方式，并将画像约束与隐因子模型中的潜在特征相融合，设计了一种基于深度学习的个性化推荐算法，揭示了两种特征存在的差异可以有效提升推荐系统的性能。本文在两个真实世界的数据集中进行了验证，实验结果显示，在预测精度与推荐列表的排序方面，本文所提出的方法相较于一些先进的算法有明显的提升。

在未来的工作中，本文将考虑图神经网络替代传统的多层感知机，利用用户与物品之间的复杂关系构建交互图，挖掘更深层次的特征关联，以提高推荐系统的性能，为用户提供更加个性化和精准的推荐服务。

基金项目

国家自然科学基金项目(61803264)资助。

参考文献

- [1] Raza, S. and Ding, C. (2021) News Recommender System: A Review of Recent Progress, Challenges, and Opportunities. *Artificial Intelligence Review*, **55**, 749-800. <https://doi.org/10.1007/s10462-021-10043-x>
- [2] Ko, H., Lee, S., Park, Y. and Choi, A. (2022) A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields. *Electronics*, **11**, Article 141. <https://doi.org/10.3390/electronics11010141>
- [3] 张玉洁, 董政, 孟祥武. 个性化广告推荐系统及其应用研究[J]. 计算机学报, 2021, 44(3): 531-563.
- [4] Isinkaye, F.O. (2021) Matrix Factorization in Recommender Systems: Algorithms, Applications, and Peculiar Challenges. *IETE Journal of Research*, **69**, 6087-6100. <https://doi.org/10.1080/03772063.2021.1997357>
- [5] Koren, Y. (2008) Factorization Meets the Neighborhood. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, 24-27 August 2008, 426-434. <https://doi.org/10.1145/1401890.1401944>
- [6] Koren, Y., Bell, R. and Volinsky, C. (2009) Matrix Factorization Techniques for Recommender Systems. *Computer*, **42**, 30-37. <https://doi.org/10.1109/mc.2009.263>
- [7] 徐芳, 应洁茹. 国内外用户画像研究综述[J]. 图书馆学研究, 2020(12): 7-16.
- [8] 周晨曦, 赵天驰, 张玲玲. 基于用户画像和链路预测的电影个性化推荐研究[J]. 系统科学与数学, 2024, 44(6): 1586-1607.
- [9] Widyaningtyas, T., Hidayah, I. and Adji, T.B. (2021) Recommendation Algorithm Using Clustering-Based UP-CSim (CB-UP-CSim). *Computers*, **10**, Article 123. <https://doi.org/10.3390/computers10100123>
- [10] Ahmed, S.F., Alam, M.S.B., Hassan, M., Rozbu, M.R., Ishtiak, T., Rafa, N., et al. (2023) Deep Learning Modelling Techniques: Current Progress, Applications, Advantages, and Challenges. *Artificial Intelligence Review*, **56**, 13521-13617. <https://doi.org/10.1007/s10462-023-10466-8>
- [11] Tegene, A., Liu, Q., Gan, Y., Dai, T., Leka, H. and Ayenew, M. (2023) Deep Learning and Embedding Based Latent Factor Model for Collaborative Recommender Systems. *Applied Sciences*, **13**, Article 726. <https://doi.org/10.3390/app13020726>
- [12] Lee, S. and Kim, D. (2022) Deep Learning Based Recommender System Using Cross Convolutional Filters. *Information Sciences*, **592**, 112-122. <https://doi.org/10.1016/j.ins.2022.01.033>
- [13] Dien, T.T., Thanh-Hai, N. and Thai-Nghe, N. (2022) An Approach for Learning Resource Recommendation Using Deep Matrix Factorization. *Journal of Information and Telecommunication*, **6**, 381-398. <https://doi.org/10.1080/24751839.2022.2058250>
- [14] Chin, W.S., Yuan, B.W., Yang, M.Y., et al. (2016) LIBMF: A Library for Parallel Matrix Factorization in Shared-Memory Systems. *Journal of Machine Learning Research*, **17**, 1-5.
- [15] Lu, S., Liu, M., Yin, L., Yin, Z., Liu, X. and Zheng, W. (2023) The Multi-Modal Fusion in Visual Question Answering: A Review of Attention Mechanisms. *Peer J Computer Science*, **9**, e1400. <https://doi.org/10.7717/peerj-cs.1400>
- [16] Xue, F., He, X., Wang, X., Xu, J., Liu, K. and Hong, R. (2019) Deep Item-Based Collaborative Filtering for Top-N Recommendation. *ACM Transactions on Information Systems*, **37**, 1-25. <https://doi.org/10.1145/3314578>
- [17] Cheng, Z., Liu, F., Mei, S., Guo, Y., Zhu, L. and Nie, L. (2022) Feature-Level Attentive ICF for Recommendation. *ACM Transactions on Information Systems*, **40**, 1-24. <https://doi.org/10.1145/3490477>
- [18] Zhai, S., Talbott, W., Srivastava, N., et al. (2021) An Attention Free Transformer.
- [19] He, X., Liao, L., Zhang, H., Nie, L., Hu, X. and Chua, T. (2017) Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, Perth, 3-7 April 2017, 173-182. <https://doi.org/10.1145/3038912.3052569>
- [20] Fu, Z., Niu, X. and Maher, M.L. (2023) Deep Learning Models for Serendipity Recommendations: A Survey and New Perspectives. *ACM Computing Surveys*, **56**, 1-26. <https://doi.org/10.1145/3605145>
- [21] Yu, M., Quan, T., Peng, Q., Yu, X. and Liu, L. (2021) A Model-Based Collaborative Filtering Algorithm Based on Stacked

- Autoencoder. *Neural Computing and Applications*, **34**, 2503-2511. <https://doi.org/10.1007/s00521-021-05933-8>
- [22] Noulapeu Ngaffo, A. and Choukair, Z. (2022) A Deep Neural Network-Based Collaborative Filtering Using a Matrix Factorization with a Twofold Regularization. *Neural Computing and Applications*, **34**, 6991-7003. <https://doi.org/10.1007/s00521-021-06831-9>
- [23] Su, Z., Lin, Z., Ai, J. and Li, H. (2021) Rating Prediction in Recommender Systems Based on User Behavior Probability and Complex Network Modeling. *IEEE Access*, **9**, 30739-30749. <https://doi.org/10.1109/access.2021.3060016>
- [24] Safarov, F., Kutlimuratov, A., Abdusalomov, A.B., Nasimov, R. and Cho, Y. (2023) Deep Learning Recommendations of E-Education Based on Clustering and Sequence. *Electronics*, **12**, Article 809. <https://doi.org/10.3390/electronics12040809>
- [25] Deldjoo, Y., Nazary, F., Ramisa, A., McAuley, J., Pellegrini, G., Bellogin, A., et al. (2023) A Review of Modern Fashion Recommender Systems. *ACM Computing Surveys*, **56**, 1-37. <https://doi.org/10.1145/3624733>
- [26] Zangerle, E. and Bauer, C. (2022) Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, **55**, 1-38. <https://doi.org/10.1145/3556536>
- [27] Moffat, A. and Zobel, J. (2008) Rank-Biased Precision for Measurement of Retrieval Effectiveness. *ACM Transactions on Information Systems*, **27**, 1-27. <https://doi.org/10.1145/1416950.1416952>
- [28] Ai, J., Cai, Y., Su, Z., Zhang, K., Peng, D. and Chen, Q. (2022) Predicting User-Item Links in Recommender Systems Based on Similarity-Network Resource Allocation. *Chaos, Solitons & Fractals*, **158**, Article 112032. <https://doi.org/10.1016/j.chaos.2022.112032>
- [29] Papadakis, H., Panagiotakis, C. and Fragopoulou, P. (2017) Scor: A Synthetic Coordinate Based Recommender System. *Expert Systems with Applications*, **79**, 8-19. <https://doi.org/10.1016/j.eswa.2017.02.025>
- [30] Su, Z., Huang, Z., Ai, J., Zhang, X., Shang, L. and Zhao, F. (2022) Enhancing the Scalability of Distance-Based Link Prediction Algorithms in Recommender Systems through Similarity Selection. *PLOS ONE*, **17**, e0271891. <https://doi.org/10.1371/journal.pone.0271891>