

# 基于评论约束与Kolmogorov-Arnold网络的深度推荐算法

柏光耀, 艾均\*, 苏湛, 马菀言

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19日; 录用日期: 2025年4月15日; 发布日期: 2025年4月25日

## 摘要

基于深度学习的推荐算法逐渐成为推荐系统领域的主流研究方向。然而, 大多数现有工作仅基于单一的用户与物品交互数据, 并且缺乏可解释性。本文对用户评论进行充分挖掘, 并且额外引入物品信息来缓解冷启动问题并提高推荐算法的准确性。该算法利用BERT预训练模型来处理文本数据, 并将用户与物品的评论特征与矩阵分解得到的潜在特征相融合, 最后在评分预测任务中使用Kolmogorov-Arnold网络进行特征学习。通过本文算法与其他基线算法在亚马逊评论数据集上进行对比, 该算法与基线算法相比显著提高了评分预测的精度以及准确率和召回率。本研究通过深入挖掘用户评论文本和物品描述信息, 证明其在提升推荐算法准确性方面的显著效果, 为推荐系统的研究提供了新的思路。

## 关键词

推荐系统, 深度学习, 评论约束, Kolmogorov-Arnold网络, 物品描述

# Deep Recommendation Algorithm Based on Review Constraints and Kolmogorov-Arnold Network

Guangyao Bai, Jun Ai\*, Zhan Su, Wanyan Ma

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

Received: Mar. 19<sup>th</sup>, 2025; accepted: Apr. 15<sup>th</sup>, 2025; published: Apr. 25<sup>th</sup>, 2025

\*通讯作者。

文章引用: 柏光耀, 艾均, 苏湛, 马菀言. 基于评论约束与 Kolmogorov-Arnold 网络的深度推荐算法[J]. 软件工程与应用, 2025, 14(2): 359-370. DOI: 10.12677/sea.2025.142032

## Abstract

Recommendation algorithms based on deep learning have emerged as a prominent research in the field of recommender systems. However, most existing approaches rely solely on user-item interaction data and lack interpretability. This article thoroughly explores user reviews and incorporates additional item information to alleviate the cold-start problem and enhance the accuracy of recommendation algorithms. The proposed approach employs the BERT pre-trained model to process textual data and integrates review-based features of users and items with latent features obtained through matrix factorization. Finally, the Kolmogorov-Arnold network is utilized for feature learning in the rating prediction task. Comparative experiments on Amazon review datasets demonstrate that the proposed algorithm significantly outperforms baseline methods in terms of rating prediction accuracy and recall. By deeply mining user review texts and item descriptions, this study validates their substantial impact on improving recommendation accuracy and offers new insights for recommender system research.

## Keywords

Recommender System, Deep Learning, Review Constraints, Kolmogorov-Arnold Network, Item Description

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

推荐系统的核心目标是通过提供个性化推荐服务以帮助用户应对日益增长的信息过载问题[1]。推荐系统通过深度挖掘用户偏好特征、项目属性信息以及用户-项目交互行为数据,进而精准预测用户对目标项目的偏好概率,最终优化用户的体验并显著提升用户粘性[2]。推荐系统能够从海量数据中自动挖掘并分析用户潜在的兴趣偏好,在创造显著商业价值的同时也成为学术界与工业界共同聚焦的研究领域[3]。

推荐系统大致可分为基于协同过滤的推荐、基于内容的推荐、基于知识的推荐以及混合推荐算法[4]。

基于协同过滤的推荐算法最早由 Amazon 和 Netflix 等公司在推荐系统的实际应用中提出并使用。协同过滤推荐算法包括两种类型分别为基于邻域的推荐和基于模型的推荐[5]。基于邻域的推荐算法其核心是通过相似性计算方法获得用户或者物品间的相似度[6],随后利用相似的用户或者物品作为邻居,从而计算用户对物品的预测评分[7]。基于模型的推荐算法通常是利用矩阵分解[8]、神经网络[9]等方法,计算用户和物品之间的复杂特征表示,从而进行推荐或者预测。

基于内容的推荐算法通常根据物品的内容特征,结合用户的历史行为数据产生推荐结果[10]。例如在电影推荐领域,物品的内容特征可能包括电影的类型、导演、演员等信息,在新闻推荐领域,物品的内容特征可能包括新闻的摘要、关键词等等。TF-IDF (Term Frequency-Inverse Document Frequency)是广泛用于推荐系统、搜索引擎以及文本挖掘中提取本文特征的算法[11]。Word2Vec [12]是用于学习单词向量化表示的模型,其利用神经网络将单词映射到连续的向量空间中,使得具有相似语义的单词在空间中的分布距离较近。

基于知识的推荐算法是利用领域知识和用户需求信息进行推荐行为,它不依赖于用户的历史行为数

据而是通过显式地建模领域知识和用户偏好来生成推荐结果[13]。这类算法通常适用于冷启动问题或数据稀疏的场景。

混合推荐算法是结合多种推荐方法的优点，以克服单一推荐方法的局限性，提升推荐系统的准确性和多样性的一种策略[14]。

本文针对如何利用用户和物品的附加信息来提升推荐算法的准确性并缓解冷启动问题，深入研究了用户评论文本的编码方法，旨在提取用户和物品的评论约束特征，提出了基于评论约束与 Kolmogorov-Arnold 网络的深度推荐算法。

## 2. 相关工作

推荐系统面临的核心挑战在于海量的用户与物品间复杂的非线性交互关系，这些因素显著增加了特征学习的难度，在神经协同过滤中，通过引入神经网络模型，隐式地建模用户与物品之间的非线性交互特征，从而有效捕捉其潜在关联[15]。深度学习技术能够有效建模用户与物品之间复杂的非线性交互关系[16]，相比传统推荐算法在个性化推荐精度和效果上具有显著优势[17]。

MCFSAE [18]通过 One-Hot 编码对用户和物品的 ID 进行特征表示，并将其与用户-物品评分矩阵融合，构建高维输入向量，进而利用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将面对维度灾难问题。DualDeepMF [19]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。DELRCR [20]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。上述基于深度学习的推荐算法都仅依赖于用户对物品的评分数据，并且可解性不足。

随着电子商务的蓬勃发展，平台在为用户提供海量商品信息的同时，也积累了规模庞大的用户评论数据。用户的评论中包含了对物品的功能、质量以及体验等多维度的客观评价信息，为深入挖掘用户偏好提供了重要的数据支撑，对提升推荐系统的精准性和个性化水平具有重要的研究价值。DeepCoNN [21]以 CNN 为基础框架，构建了一个双通道并行神经网络结构，分别利用评论数据来学习用户和物品的特征，该实验证明了利用评论文本可以有效缓解稀疏性问题。NARRE [22]在 DeepCoNN 的基础上，引入注意力机制以动态评估每条评论的贡献权重，通过聚焦于更具信息量的评论内容来优化特征提取过程，从而显著提升模型的预测精度与可解释性。CAML [23]提出一种互注意力可解释性推荐模型，利用推荐和解释任务之间的相关性提升两者的性能。PETER [24]使用 Transformer [25]架构，对用户和物品的 ID 进行建模，设计了一个上下文预测的任务来连接 ID 和评论文本，根据用户和项目 ID 同时产生推荐和生成解释，以获得可解释的推荐。

尽管已有许多研究工作采用 CNN 来提取用户和物品评论中的特征，但这种方法在处理长序列文本时存在显著局限性。此外，现有方法在评分预测任务中主要依赖于 MLP，并且仅利用用户的评论文本信息而忽略了物品描述中的丰富内容。物品描述通常包含对物品属性的全面介绍，对于缓解物品冷启动问题具有重要的辅助作用。本文提出的 RCKAN (Review Constrained Kolmogorov-Arnold Network)利用预训练的 BERT [26]模型来处理文本数据，有效克服了 CNN 提取局部特征的局限性，引入了物品的描述文档，为解决物品冷启动问题提供了新的思路，并且在评分预测任务中使用 KAN [27]替代传统 MLP 进一步提升了评分预测任务的准确性。

## 3. 算法设计

本文提出的 RCKAN 是由用户以及物品的潜在特征、用户以及物品的评论约束以及基于 Kolmogorov-

Arnold 的评分预测这三个部分组成, 整体算法如图 1 所示。

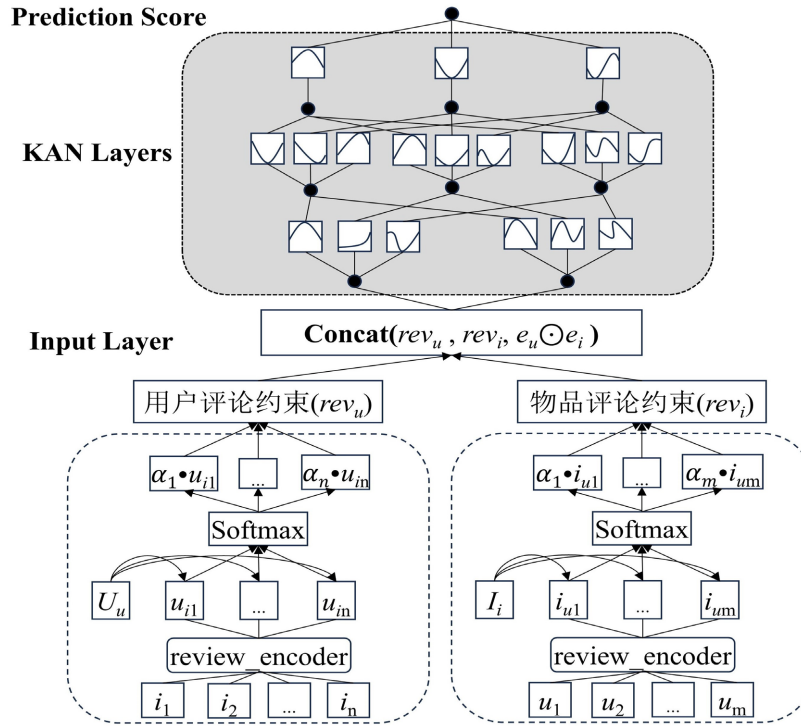


Figure 1. Schematic diagram of RCKAN algorithm design

图 1. RCKAN 算法设计示意图

### 3.1. 用户以及物品的潜在特征

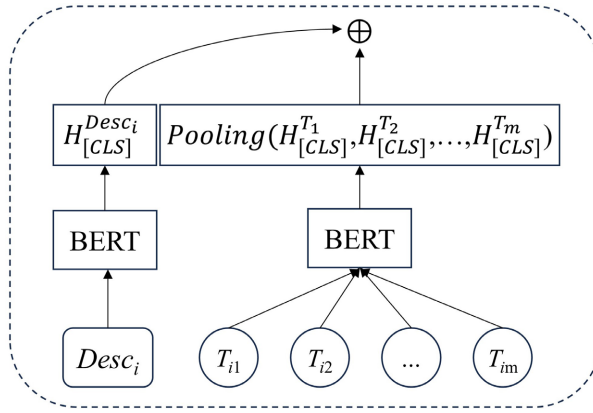
LIBMF [28]是一个高效且专门设计用于大规模数据的矩阵分解算法。它主要用于协同过滤任务, 特别是在推荐系统中处理用户-物品评分矩阵。LIBMF 的核心优势在于其能够高效处理大规模数据集, 同时保持较低的内存占用和较高的计算性能。LIBMF 的目标函数如公式(1)所示, 其中  $R_{ui}$  为真实评分,  $P_u$  和  $Q_i$  为用户与物品的潜在向量, 并使用 L2 正则化系数防止过拟合。

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

本文采用 LIBMF 对用户-物品评分矩阵进行分解处理, 旨在提取用户和物品的潜在特征向量, 并将这些预计算得到的潜在特征作为算法中用户和物品嵌入层的初始化参数, 在后续均以  $e_u$  和  $e_i$  来表示用户以及物品的潜在特征。

### 3.2. 用户以及物品的评论约束

在现实生活中, 用户可能会对浏览、点击、购买过的物品进行评论, 评论中包含了用户对物品的功能、质量等方面是否满意的信息。对于用户而言, 物品的描述性信息具有同等的重要性, 尤其是在面对不熟悉的物品时, 这类信息往往是吸引用户注意力的关键因素。此外, 当某些新型物品缺乏与用户的交互记录时, 详尽的物品描述能够补充提供必要的项目信息, 即便是在数据稀缺的情况下, 也能增强用户对物品的理解和兴趣, 从而提高用户的体验和满意度。因此本文将评论信息结合物品的描述作为一种用户对物品的偏好约束, 并利用预训练的 BERT 模型来处理文本数据。



**Figure 2.** Schematic diagram of review constraint feature encoding module

**图 2.** 评论约束特征编码模块示意图

假设某个用户  $u$ ，他的所有评论物品为  $I = (i_1, i_2, \dots, i_n)$ ，假设物品  $i$  收到的所有评论文本为  $T_i = (T_{i1}, T_{i2}, \dots, T_{im})$ ，即有  $m$  个用户对该物品有过评论。在 review\_encoder 模块中，如图 2 所示，其中  $\oplus$  表示向量拼接操作。将物品的所有的评论输入进 BERT 中，取 BERT 中的 [CLS] 标记位得到评论的隐藏表示  $H$ ，取所有评论的隐表示的平均值，并将该物品的描述信息  $Desc_i$  输入进 BERT 中，得到物品描述隐藏表示。最后，将二者进行拼接操作，得到物品  $i$  的特征  $u_i$ ，如公式(2)所示。

$$u_i = \text{Concat}[\text{BERT}(Desc_i), \text{Pooling}(\text{BERT}(T_{i1}, T_{i2}, \dots, T_{im}))] \quad (2)$$

随后利用嵌入技术将用户的 ID 嵌入到与物品的特征相同维度的空间中，得到用户的嵌入向量  $U_u$ ，并与物品特征进行点积计算，得到用户  $u$  对物品  $i$  的特征的相关度  $\alpha_i$ ， $\alpha_i$  越大说明用户对物品的关注程度越高，用户对物品的喜爱程度也越高。将  $\alpha_i$  ( $i = 1, 2, \dots, n$ ) 通过 Softmax 进行归一化操作，并与  $u_i$  相乘得到每个物品加权后的特征，最后取所有物品的特征的平均值，作为用户的评论约束表示  $rev_u$ 。如公式(3)和(4)所示。

$$\alpha_i = U_u \cdot u_i \quad (3)$$

$$rev_u = \text{Pooling}[\text{Softmax}(\alpha_1) \cdot u_{i1}, \dots, \text{Softmax}(\alpha_n) \cdot u_{in}] \quad (4)$$

通过相同的操作，对物品  $i$  取其所有的评论用户，在这一部分仅利用评论信息，并通过 review\_encoder 模块获得用户的评论特征，将物品 ID 嵌入到相同维度的空间中，并计算物品对于每个用户的吸引程度，最后获得物品的评论约束表示  $rev_i$ 。

### 3.3. 基于 Kolmogorov-Arnold 网络的评分预测

Kolmogorov-Arnold 网络，简称为 KAN，是一种基于 Kolmogorov-Arnold 表示定理设计的神经网络模型，和传统的 MLP 不同，KAN 通过可学习的函数替代固定的激活函数，消除了对线性权重矩阵的依赖。对于任意一个  $d$  变量的连续函数  $f(x_1, x_2, \dots, x_d)$  都可以表示为单变量连续函数的两层嵌套叠加。具体而言，对于任意连续函数  $f: [0, 1]^d \rightarrow \mathbb{R}$ ，存在单变量连续函数  $\phi_q$  和  $\varphi_{p,q}$ ，都有公式(5)，

$$f(x_1, x_2, \dots, x_d) = \sum_{q=1}^{2d+1} \phi_q \left[ \sum_{p=1}^d \varphi_{q,p}(x_p) \right] \quad (5)$$

其中， $\varphi_{p,q}$  为内部函数(架构中的第 0 层)， $\phi_q$  为外部函数(架构中的第 1 层)，输入层为  $d$  个节点，隐藏层



为  $2d+1$  个节点, 每个节点连接到所有的输入节点。每个隐藏节点应用一个单变量函数  $\phi_{p,q}$ , 输出层接收隐藏层所有节点的输出, 通过应用函数  $\phi_q$  并加权求和得到最终输出, 如图 3 所示。对于多层的 KAN 网络, 其向量化公式如(6)所示。

$$\text{KAN}(x) = (\Phi_3 \circ \Phi_2 \circ \Phi_1)(x) \quad (6)$$

与 MLP 相比, 除去了激活函数和参数矩阵的嵌套, 而直接是非线性函数  $\Phi$  的嵌套。非线性函数采用添加偏置函数控制的含样条(spline)函数, 如公式(7)所示。

$$\Phi(x) = \omega [b(x) + \text{spline}(x)] \quad (7)$$

其中,  $\omega$  起到对于激活函数的控制作用, 另外基础函数  $b(x)$  如公式(8)所示。

$$b(x) = \text{silu}(x) = \frac{x}{1 + e^{-x}} \quad (8)$$

对于样条激活函数而言, 其中 B 样条函数的权重参数  $c_i$  为可训练对象, 如公式(9)所示。

$$\text{spline}(x) = \sum_i c_i B_i(x) \quad (9)$$

本文将上述用户和物品的评论约束特征与用户以及物品的潜在特征拼接, 之后利用 KAN 网络预测用户对物品的评分, 如公式(10)所示。

$$\hat{r}_{ui} = \text{KAN}[\text{concat}(rev_u, rev_i, e_u \odot e_i)] \quad (10)$$

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(11)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[29]。

$$\text{Loss} = \sum_{u,i} (\hat{r}_{ui} - r_{ui})^2 \quad (11)$$

其中,  $\hat{r}_{ui}$  表示用户  $u$  对物品  $i$  的预测评分,  $r_{ui}$  为真实评分,  $u$  和  $i$  均属于训练集中的数据。

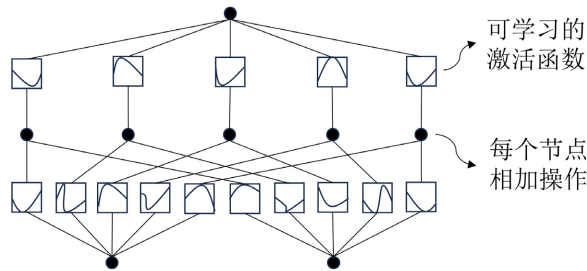


Figure 3. Calculation diagram of KAN  
图 3. KAN 计算示意图

## 4. 实验设计与分析

### 4.1. 数据集

Amazon Review Dataset 公开数据集记录了用户对亚马逊网站各类产品的评价, 本文选取其中 Movies and TV 数据集以验证所提出的算法。该数据集包含 12 万余用户对 5 万余物品的评分, 评分区间为 1 至 5, 间隔为 1, 以及包含 167 万余条评论, 为了节省计算资源和时间本文从该数据集中分别选取 5000 以及 10000 名用户的所有数据展开实验, 在后续标记为 Amazon-5000 和 Amazon-10000。本文将上述数据集按照 80%, 10%, 10% 的比例将数据集划分为训练集、验证集和测试集, 每个算法仅基于训练集预测每

个测试集中的所有评分数据，并重复随机选取五次，进行折五验证，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)、均方根误差(Root Mean Square Error, RMSE)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(12)和(13)所示。

$$MAE = \frac{\sum_{i \in Test} |r_{ui} - \hat{r}_{ui}|}{|Test|} \quad (12)$$

$$RMSE = \frac{\sqrt{\sum_{i \in Test} |r_{ui} - \hat{r}_{ui}|^2}}{|Test|} \quad (13)$$

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)和半衰期效用指标(Half Life Utility, HLU) [30]来评估推荐列表的排序性能，NDCG 如公式(14)和(15)所示，HLU 如公式(16)所示。

$$DCG = \sum_{i=1}^b R_i + \sum_{i=b+1}^N \frac{R_i}{\log_b r_i} \quad (14)$$

$$NDCG = \frac{\sum_{u=1}^m DCG}{m} \quad (15)$$

在 NDCG 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， $m$  表示推荐列表的数量。 $R_i$  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。

$$HLU = \frac{\sum_{i=1}^m \frac{\max(r_{ui} - d, 0)}{2^{i-1}(h-1)}}{m} \quad (16)$$

在 HLU 中， $h$  表示半衰期阈值，在本文的实验中设置为 2， $d$  是用户的中性评价，本文按照用户的平均评分进行计算。

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(17)、(18)和(19)所示。

$$Precision = \frac{FP}{TP + FP} \quad (17)$$

$$Recall = \frac{FP}{TP + FN} \quad (18)$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (19)$$

## 4.3. 基线算法

为了评估本文提出的 RCKAN 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) LIBMF 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) DSS [31]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) PETER 使用 Transformer 架构, 对用户和物品的 ID 进行建模, 设计了一个上下文预测的任务来连接 ID 和评论文本, 根据用户和项目 ID 同时产生推荐和生成解释。
- (3) DELCR 从嵌入技术中获得的稠密向量转为单独的深度神经网络, 从而形成用户和物品的双塔结构, 并且将用户与物品的内积集成到神经网络的输出层从而预测评分。
- (5) RCKAN 为本文提出的算法, 本文将潜在特征的大小设置为 300, 使用 Adam 优化器来调整模型的参数, 学习率设置为[0.001, 0.005, 0.001], 批次大小设置为[16, 32, 64, 128], KAN 网络的层数设置为[2, 3, 4], 隐藏层神经元个数设置为[900, 600, 300, 100], 在上述设置中寻找模型的最优参数。

4.4. 实验结果与分析

4.4.1. 各指标对比结果

各算法在 Amazon-5000 和 Amazon-10000 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。整体而言, RCKAN 的 MAE 值和 RMSE 值在两个数据集上相较于其他基线算法均达到了最优。其中, 在 Amazon-5000 中, RCKAN 的 MAE 值与 RMSE 值分别为 0.7934 和 1.0663, 相较于其他基线算法分别平均降低了 5.47%和 9.61%。在 Amazon-10000 中, RCKAN 的 MAE 值与 RMSE 值分别为 0.7612 和 1.0339, 相较于其他基线算法分别平均降低了 5.42%和 9.38%。

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	Amazon-5000		Amazon-10000	
	MAE	RMSE	MAE	RMSE
LIBMF	0.8207	1.0960	0.7888	1.0475
DSS	0.8606	1.2301	0.8337	1.2016
PETER	0.9022	1.2395	0.8635	1.2088
DELCR	0.8087	1.0842	0.7757	1.0531
RCKAN	<b>0.7934</b>	<b>1.0663</b>	<b>0.7612</b>	<b>1.0339</b>

表 2 显示了各算法的 NDCG 和 HLU 在两个数据集上的结果对比, NDCG 和 HLU 均用于衡量推荐列表的排序性能。在两个数据集中, PETER 在 NDCG 指标上均取得了最好成绩, 表明其在排序相关性方面表现卓越, DSS 在 HLU 指标上表现最佳, 体现了其在推荐列表准确性方面的优势。RCKAN 的 NDCG 值分别为 0.7171 和 0.7244, 在 NDCG 指标中排名第二, 略低于 PETER, 与排名第一的 PETER 平均相差 0.74%。HLU 值分别为 0.7478 和 0.7641, 在 HLU 指标中排名第三低于 DSS 与 PETER, 其中 DSS 要显著优于其他算法, 说明其能够将用户真正感兴趣的项目排在推荐列表的较前位置, 提升了用户发现这些项目的概率。

表 3 显示了各算法在分类指标上的对比结果, 其中 LIBMF、DSS 以及 PETER 在召回指标上表现较差, RCKAN 在两个数据集上的 Recall 值均排名第一, DECLR 的召回指标与 RCKAN 接近, 比 RCKAN 平均低 2.87%, 说明本文提出的算法能够为用户识别并推荐更多的目标项目。在 Accuracy 指标中, RCKAN 仍排名第一, 略高于 DELCR, 在两个数据集上的整体提升分别为 2.11%和 2.18%。Precision 指标中排名



第一的是 LIBMF，比 RCKAN 平均高出 2.89%，说明 RCKAN 在精准识别用户真实偏好的物品方面，仍存在一定的局限性。

**Table 2.** Comparison results of NDCG and HLU for each algorithm

**表 2.** 各算法在两个数据集中的 NDCG 和 HLU 的对比结果

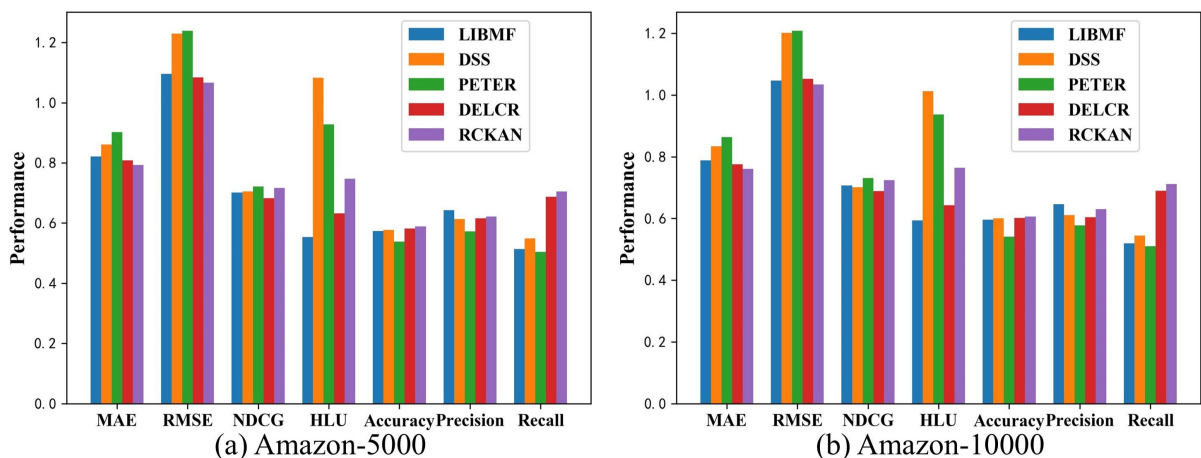
Methods	Amazon-5000		Amazon-10000	
	NDCG	HLU	NDCG	HLU
LIBMF	0.7018	0.5530	0.7069	0.5939
DSS	0.7052	<b>1.0829</b>	0.7008	<b>1.0127</b>
PETER	<b>0.7214</b>	0.9283	<b>0.7309</b>	0.9367
DELCR	0.6829	0.6319	0.6884	0.6434
RCKAN	0.7171	0.7478	0.7244	0.7641

**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	Amazon-5000			Amazon-10000		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
LIBMF	0.5734	<b>0.6425</b>	0.5140	0.5956	<b>0.6458</b>	0.5196
DSS	0.5767	0.6138	0.5491	0.6008	0.6112	0.5442
PETER	0.5383	0.5726	0.5041	0.5414	0.5781	0.5096
DELCR	0.5814	0.5959	0.6869	0.6022	0.6041	0.6902
DPFLE	<b>0.5886</b>	0.6216	<b>0.7049</b>	<b>0.6068</b>	0.6307	<b>0.7118</b>

整体实验结果如图 4 所示，本文提出的 RCKAN 在 MAE、RMSE、Accuracy 以及 Recall 这四个指标上均取得了最好的效果。实验结果显示，RCKAN 相比于其他基线算法拥有更好的预测精度，说明用户的评论中蕴含了丰富的隐式反馈和细致的个人情感信息，这些信息能够揭示用户对特定物品的具体偏好，通过充分利用用户对于物品的评论信息，不仅有利于增强算法的预测精准度，而且在准确率和召回率指标上也有可观的提升。然而，在推荐列表的排序性能方面 RCKAN 表现并不出色，对项目之间的相对重要性区分能力较弱，导致高价值项目的排名可能不够突出。



**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

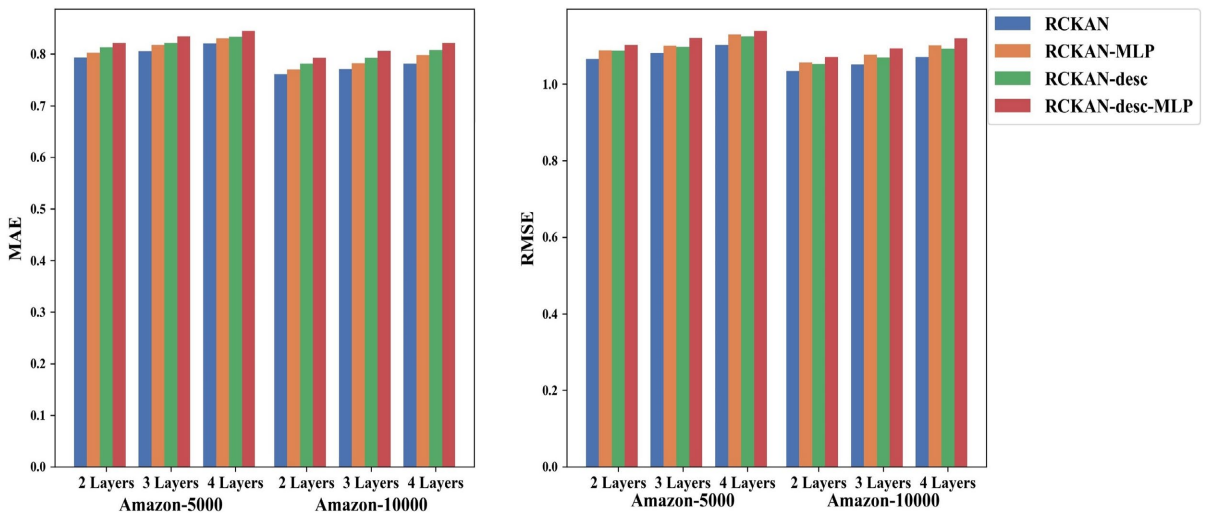
4.4.2. 实验分析

表 4 展示了在 RCKAN 中是否利用物品的描述信息的 MAE 与 RMSE 对比结果，其中 RCKAN-desc 表示在计算用户评论约束时，不添加额外的物品描述。通过实验结果可知，添加了物品描述的 RCKAN 算法比仅利用评论信息的 RCKAN-desc 的 MAE 与 RMSE 均要更低，其中 MAE 平均降低了 2.51%，RMSE 平均降低了 1.97%，说明利用物品描述信息来充实物品属性，可以更准确地匹配用户的偏好，从而有效提高推荐算法的准确度。

**Table 4.** The impact of adding item descriptions on experimental results  
**表 4.** 添加物品描述对实验结果的影响

Methods	Amazon-5000		Amazon-10000	
	MAE	RMSE	MAE	RMSE
RCKAN-desc	0.8132	1.0871	0.7814	1.0533
RCKAN	<b>0.7934</b>	<b>1.0663</b>	<b>0.7612</b>	<b>1.0339</b>

图 5 显示了利用 KAN 神经网络与传统的 MLP 的评分预测误差对比结果，其中 RCKAN-MLP 表示在评分预测阶段使用 MLP 替代 KAN，RCKAN-desc-MLP 表示计算用户评论约束时，不添加额外的物品描述并且在预测阶段使用 MLP。由实验结果可知，RCKAN-MLP 在两个数据集中的 MAE 值分别为 0.8028 和 0.7701，RMSE 值分别为 1.0885，1.0564，利用 KAN 网络进行评分预测和 MLP 相较，MAE 平均降低了 1.17%，RMSE 平均降低了 2.09%，说明 KAN 网络通过可学习的函数替代固定的激活函数，将网络边缘的每个权重参数通过样条函数形式替换，通过局部结构组合表示任意复杂的非线性函数使其具备更强的表示复杂关系的能力，在预测用户对物品的评分任务上拥有更优的表现，并且增强了模型的可解释。本文还探索了不同隐藏层个数对实验结果的影响，结果显示当隐藏层为 2 时，KAN 与 MLP 均达到最优效果，当隐藏层个数增加时，二者的 MAE 和 RMSE 均有不同幅度的上升，其中当 KAN 的隐藏层为 2 时比其他隐藏层个数的 MAE 和 RMSE 在两个数据集中分别平均低 2.21%和 2.45%，说明当网络层数过深时，层间的变换数量增多可能导致学习到的表示变得过于复杂。过多的非线性变换可能会过拟合训练数据，从而削弱模型的泛化能力。



**Figure 5.** Comparison results of KAN and MLP in rating prediction tasks  
**图 5.** KAN 与 MLP 在评分预测任务中的对比结果

## 5. 总结与展望

针对推荐系统准确性不足以及冷启动问题, 本文利用用户评论数据的同时还引入物品的描述文档, 利用 BERT 对本文数据进行编码, 将用户与物品的评论约束特征与通过矩阵分解得到的潜在特征相融合, 并在评分预测任务中利用 Kolmogorov-Arnold 神经网络进行特征学习。实验结果表明, 在预测精度以及准确率、召回率等方面, 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, 本文考虑引入更先进的排序学习算法来优化推荐项目的排序, 从而进一步提升用户体验, 同时探索如何有效融合其他多模态数据如图像、视频等, 以更全面地捕捉用户偏好和物品特征, 从而提高推荐的精准性和多样性。

## 参考文献

- [1] Roy, D. and Dutta, M. (2022) A Systematic Review and Research Perspective on Recommender Systems. *Journal of Big Data*, **9**, Article No. 59. <https://doi.org/10.1186/s40537-022-00592-5>
- [2] Saifudin, I. and Widiyaningtyas, T. (2024) Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets. *IEEE Access*, **12**, 19827-19847. <https://doi.org/10.1109/access.2024.3359274>
- [3] 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>
- [4] 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>
- [5] Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E. and Fragopoulou, P. (2022) Collaborative Filtering Recommender Systems Taxonomy. *Knowledge and Information Systems*, **64**, 35-74. <https://doi.org/10.1007/s10115-021-01628-7>
- [6] Fkih, F. (2022) Similarity Measures for Collaborative Filtering-Based Recommender Systems: Review and Experimental Comparison. *Journal of King Saud University—Computer and Information Sciences*, **34**, 7645-7669. <https://doi.org/10.1016/j.jksuci.2021.09.014>
- [7] Jena, K.K., Bhoi, S.K., Malik, T.K., Sahoo, K.S., Jhanjhi, N.Z., Bhatia, S., et al. (2022) E-learning Course Recommender System Using Collaborative Filtering Models. *Electronics*, **12**, Article 157. <https://doi.org/10.3390/electronics12010157>
- [8] 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>
- [9] Zhang, S., Yao, L., Sun, A., et al. (2019) Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys (CSUR)*, **52**, 1-38.
- [10] Papadakis, H., Papagrigoriou, A., Kosmas, E., Panagiotakis, C., Markaki, S. and Fragopoulou, P. (2023) Content-Based Recommender Systems Taxonomy. *Foundations of Computing and Decision Sciences*, **48**, 211-241. <https://doi.org/10.2478/fcds-2023-0009>
- [11] Liang, M. and Niu, T. (2022) Research on Text Classification Techniques Based on Improved TF-IDF Algorithm and LSTM Inputs. *Procedia Computer Science*, **208**, 460-470. <https://doi.org/10.1016/j.procs.2022.10.064>
- [12] Church, K.W. (2016) Word2Vec. *Natural Language Engineering*, **23**, 155-162. <https://doi.org/10.1017/s1351324916000334>
- [13] Tarus, J.K., Niu, Z. and Mustafa, G. (2017) Knowledge-Based Recommendation: A Review of Ontology-Based Recommender Systems for E-learning. *Artificial Intelligence Review*, **50**, 21-48. <https://doi.org/10.1007/s10462-017-9539-5>
- [14] Çano, E. and Morisio, M. (2017) Hybrid Recommender Systems: A Systematic Literature Review. *Intelligent Data Analysis*, **21**, 1487-1524. <https://doi.org/10.3233/ida-163209>
- [15] 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>
- [16] Sharifani, K. and Amini, M. (2023) Machine Learning and Deep Learning: A Review of Methods and Applications. *World Information Technology and Engineering Journal*, **10**, 3897-3904.
- [17] Li, C., Ishak, I., Ibrahim, H., et al. (2023) Deep Learning-Based Recommendation System: Systematic Review and Classification. *IEEE Access*, **11**, 113790-113835. <https://doi.org/10.1109/ACCESS.2023.3323353>
- [18] Yu, M., Quan, T., Peng, Q., Yu, X. and Liu, L. (2021) A Model-Based Collaborate Filtering Algorithm Based on Stacked Autoencoder. *Neural Computing and Applications*, **34**, 2503-2511. <https://doi.org/10.1007/s00521-021-05933-8>

- 
- [19] 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>
- [20] 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>
- [21] Zheng, L., Noroozi, V. and Yu, P.S. (2017) Joint Deep Modeling of Users and Items Using Reviews for Recommendation. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, Cambridge, 6-10 February 2017, 425-434. <https://doi.org/10.1145/3018661.3018665>
- [22] Chen, C., Zhang, M., Liu, Y. and Ma, S. (2018) Neural Attentional Rating Regression with Review-Level Explanations. *Proceedings of the 2018 World Wide Web Conference on World Wide Web—WWW'18*, Lyon, 23-27 April 2018, 1583-1592. <https://doi.org/10.1145/3178876.3186070>
- [23] Chen, Z., Wang, X., Xie, X., Wu, T., Bu, G., Wang, Y., et al. (2019) Co-Attentive Multi-Task Learning for Explainable Recommendation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, Macao, 10-16 August 2019, 2137-2143. <https://doi.org/10.24963/ijcai.2019/296>
- [24] Li, L., Zhang, Y. and Chen, L. (2021) Personalized Transformer for Explainable Recommendation. arXiv: 2105.11601.
- [25] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017) Attention Is All You Need. arXiv: 1706.03762.
- [26] Devlin, J., Chang, M.W., Lee, K., et al. (2018) Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805.
- [27] Liu, Z., Wang, Y., Vaidya, S., et al. (2024) Kan: Kolmogorov-Arnold Networks. arXiv: 2404.19756.
- [28] 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.
- [29] 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>
- [30] Jalili, M., Ahmadian, S., Izadi, M., Moradi, P. and Salehi, M. (2018) Evaluating Collaborative Filtering Recommender Algorithms: A Survey. *IEEE Access*, **6**, 74003-74024. <https://doi.org/10.1109/access.2018.2883742>
- [31] 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>