

基于NCF框架的CNMF模型在推荐领域的研究

崔一航

天津工业大学计算机科学与技术学院, 天津

收稿日期: 2023年9月16日; 录用日期: 2023年10月16日; 发布日期: 2023年10月23日

摘要

深度学习在推荐系统领域取得了显著的成功, 然而, 现有的一些模型仍存在一些局限性, 如不能捕捉用户和物品之间的非线性交互信息。为了解决这个问题, 我们对NeuMF模型进行了改进, 提出CNMF模型。首先, 我们将NeuMF模型中的多层感知机(MLP)层替换为卷积神经网络(CNN)。CNN能够有效地提取用户和物品之间的时空特征, 从而更好地捕捉它们之间的关系。其次, 我们引入了注意力机制来进一步增强模型的性能。注意力机制可以自动学习用户和物品之间的重要关系, 从而更好地建模推荐过程。我们通过计算用户和物品之间的注意力权重来加权池化其交互特征, 从而更准确地预测用户的喜好。最后, 我们在经典的推荐数据集上进行了大量的实验。实验结果表明, 我们提出的改进算法在准确性和效率方面显著优于传统的NeuMF模型。特别是在用户和物品数量较大的情况下, 我们的算法展现出更好的稳定性和可扩展性。

关键词

深度学习, 推荐算法, 神经网络, 特征

Research on CNMF Model Based on NCF Framework in the Field of Recommendation

Yihang Cui

College of Computer Science and Technology, Tiangong University, Tianjin

Received: Sep. 16th, 2023; accepted: Oct. 16th, 2023; published: Oct. 23rd, 2023

Abstract

Deep learning has achieved remarkable success in the field of recommendation systems, however, some existing models still have some limitations, such as the inability to capture non-linear interaction information between users and items. To solve this problem, we improved NeuMF model

and proposed CNMF model. First, we replace the multi-layer perceptron (MLP) layer in the NeuMF model with a convolutional neural network (CNN). CNN can effectively extract the spatio-temporal characteristics between users and items, so as to better capture the relationship between them. Secondly, we introduce the attention mechanism to further enhance the performance of the model. Attention mechanisms can automatically learn important relationships between users and items to better model the recommendation process. We can more accurately predict user preferences by calculating the weight of attention between users and items to pool their interaction characteristics. Finally, we conducted a large number of experiments on classical recommendation datasets. Experimental results show that our improved algorithm is significantly superior to the traditional NeuMF model in terms of accuracy and efficiency. Especially in the case of a large number of users and items, our algorithm shows better stability and scalability.

Keywords

Deep Learning, Recommendation Algorithms, Neural Networks, Features

Copyright © 2023 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

推荐算法是信息检索和个性化推荐领域的重要研究方向，它对于帮助用户发现和获取感兴趣的内容具有重要意义。随着互联网的迅速发展和信息爆炸式增长，用户面临着海量的信息选择，如何从中快速准确地找到符合个人需求的内容成为了一个紧迫的问题。推荐算法的作用就在于通过分析和挖掘用户的行为信息、个人兴趣和社交网络等数据，提供个性化的推荐服务，为用户提供更有针对性和满意度的用户体验[1] [2] [3] [4]。

传统的推荐算法主要基于协同过滤和内容过滤两种思路进行研究，但随着互联网技术和数据挖掘技术的不断进步，新的推荐算法也不断涌现。比如基于机器学习的推荐算法利用用户行为数据和物品特征，通过构建模型来预测用户对未知物品的偏好，实现精准推荐。而基于深度学习的推荐算法则通过多层神经网络模型，对海量数据进行学习和表示，能够更好地挖掘用户和物品之间的复杂关系，提高推荐的准确性和效果。

尽管现有的推荐算法已经取得了一定的成果，但仍然存在一些挑战和问题。例如，数据稀疏性、冷启动问题、可解释性等方面的难题对推荐算法的发展提出了新的要求。因此，本文旨在探讨和研究针对这些问题的创新算法和方法。通过对现有推荐算法的综述和分析，本文将提出一种基于 NCF (Neural Collaborative Filtering) 框架所提出的 CNMF (CNN Neural Matrix Factorization) 推荐算法，该算法能够克服传统算法的局限性，并在实验中验证其效果。希望通过本研究的努力，能够为信息检索和个性化推荐领域的进一步发展提供一定的理论和实践支持，并改进用户体验，推动互联网应用的智能化发展。

2. 算法描述

2.1. NCF 框架

CNMF 模型的整体架构是基于 NCF 通用框架。所以，CNMF 模型与 NCF 保持有相同的框架结构。如图 1 所示，NCF 框架[5]主要具有四层结构：

Input Layer: 输入层的 User (用户)信息和 Item (物品)信息通过 one-hot 编码转化成了稀疏向量。

Embedding Layer: 在这之后, 把 User 向量和 Item 向量分别嵌入到一个较小维数的空间, 假设为 K 维。嵌入层的具体实现是把输入的向量 u_i 和嵌入矩阵 P 相乘, 得到该向量的嵌入向量 p_i 。如果一共 M 个用户, 嵌入维数为 K 维, 那么对于用户的嵌入矩阵的大小为 $M \times K$, 其中嵌入矩阵的第 i 行代表第 i 个用户的嵌入向量。(如果一共 N 个物品, 那么对于物品的嵌入矩阵大小为 $N \times K$)。

Neural CF Layers: 这一层为神经协同过滤层。将用户的嵌入向量和物品的嵌入向量送入 NCF layers 后得到一个新向量。本文所提出的算法思想也是重点对这一层进行了改进。

Output Layer: 通过全连接层把 NCF layers 输出的向量映射到最终的预测分数 \hat{y}_{ui} , 最后再通过损失函数 $l(\hat{y}_{ui}, y_{ui})$ 进行梯度下降更新参数。

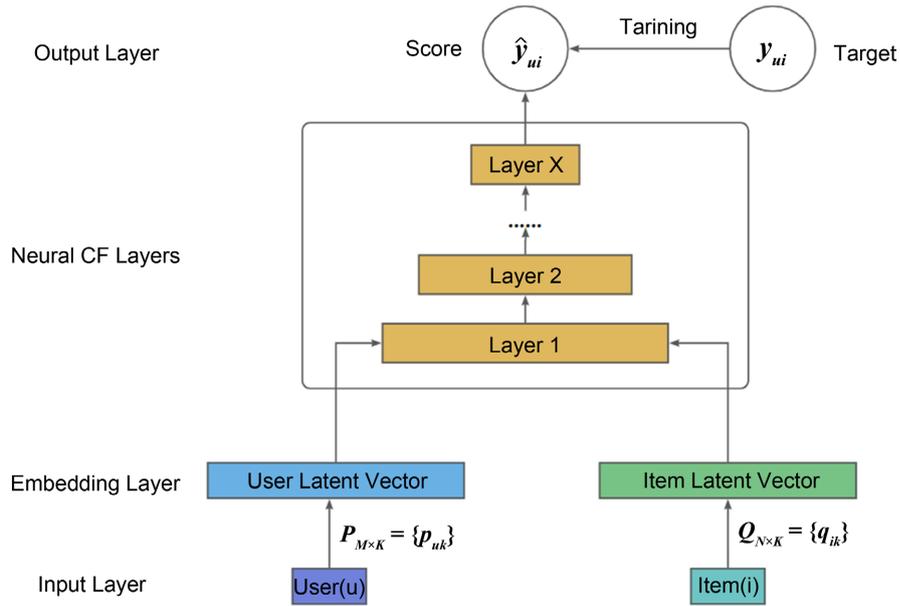


Figure 1. NCF frame
图 1. NCF 框架

2.2. CNMF 模型

2.2.1. General Matrix Factorization(GMF)

广义矩阵分解用的是向量对应元素相乘, 得到的是向量, 而传统的矩阵分解使用的是向量内积, 得到的是标量。这是二者的不同之处。

对于通过嵌入层得到的用户 u 嵌入向量 p_u 和物品 i 嵌入向量 q_i , 将他们逐元素相乘, 如式(1)~(3)所示。

$$p_u = [p^1, \dots, p^k] \tag{1}$$

$$q_i = [q^1, \dots, q^k] \tag{2}$$

$$\phi_0(p_u, q_i) = p_u \odot q_i = [p^1 q^1, \dots, p^k q^k] \tag{3}$$

由于运算规则的限定, 用户嵌入层和物品嵌入层的嵌入维数要相同, 都为 k 。

在这之后通过无 bias 的全连接层和激活函数得到预测分数, 如式(4)所示:

$$\hat{y}_{ui} = \sigma(\phi_x(z_{x-1})^T h) \tag{4}$$

其中 σ 为激活函数, h 为全连接层的权重矩阵, 大小为 $k \times 1$, 如图 2 所示。

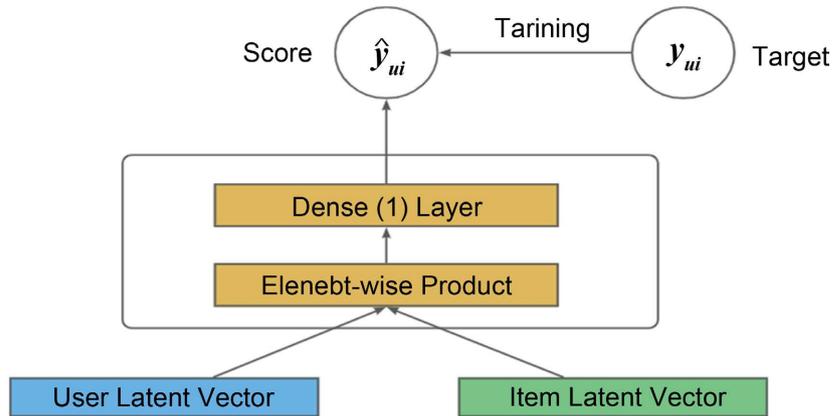


Figure 2. GMF model
图 2. GMF 模型

2.2.2. Convolutional Neural Network (CNN)

本文所提出的模型使用 CNN 作为其非线性模块的核心, 用 ReLU 函数作为激活函数, 这个模型对嵌入向量的处理与 GMF 模型中不同。首先, 作为输入的用户 u 嵌入向量 p_u 和物品 i 嵌入向量 q_i 要先经过一层 Attention layer 得到加权的物品向量 I_w , 将加权向量 I_w 与用户 u 嵌入向量 p_u 进行拼接, 如式(5)~(6)所示。

$$I_w = q_i \times \text{softmax}(p_u^T \cdot q_i) \tag{5}$$

$$z_1 = \phi_1(p_u, I_w) = [p_u, I_w] \tag{6}$$

通过这一层加强其特征提取, 提高其准确性。将拼接后的向量送入 CNN 层, 如式(7)所示。

$$\phi_x(z_{x-1}) = \sigma_x(z_{x-1}^T W_x + b_x) \tag{7}$$

其中 σ 是激活函数, 这里选择的是 ReLU 函数, W_i 和 b_i 是两个可学习的参数。最后通过无 bias 的全连接层和激活函数得到预测分数, 如式(4)所示。整体结构如图 3 所示。

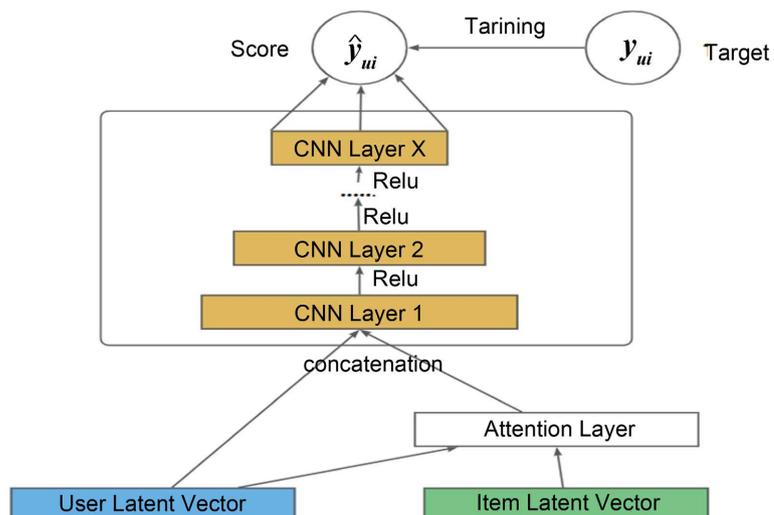


Figure 3. CNN model
图 3. CNN 模型

2.2.3. CNN Matrix Factorization (CNMF)

CNMF 模型把 GMF 和 CNN 模型结合起来，使他们相互强化，以得到更好的效果。

为了增加融合模型的灵活性，本文采用了分别学习广义矩阵分解(GMF)和卷积神经网络(CNN)的嵌入层矩阵的方法。在这种方法中，分别对 GMF 和 CNN 进行训练，学习它们各自的嵌入表示。在模型的最后一层之前，将 GMF 和 CNN 的输出向量连接在一起，形成一个更大的向量。通过把这个连结后的向量传递给最后的 CNMF 层以预测最终的分数。

令 GMF 的嵌入矩阵为 P^G 和 Q^G ，分别对应用户嵌入矩阵和物品嵌入矩阵，CNN 的嵌入矩阵为 P_M 和 Q_M 。那么对于用户 u 和物品 i ，GMF 中的嵌入向量为 p_u^G 和 q_i^G ，CNN 中的嵌入向量为 p_u^M 和 q_i^M 。具体公式如下：

左侧 GMF 部分：

$$\phi_{ui}^{GMF} = p_u^G \Theta q_i^G \tag{8}$$

右侧 CNN 部分：

$$z_1 = \phi_{ui}^{CNN}(p_u, I_w) = [p_u, I_w] \tag{9}$$

$$\phi_{ui}^{CNN} = \phi_x^{CNN}(z_{x-1}) = \sigma_x(z_{x-1}^T W_x + b_x) \tag{10}$$

这时候得到了两个向量 ϕ_{ui}^{GMF} 和 ϕ_{ui}^{CNN} ，将其进行连接传入最后的 CNMF 层，这一层的作为全连接层，把向量映射到一维空间，得到最后的预测分数：

$$z_{ui} = [\phi_{ui}^{GMF}, \phi_{ui}^{CNN}] \tag{11}$$

$$\hat{y}_{ui} = \sigma(Z_{ui}^T h) \tag{12}$$

整体结构如图 4 所示。

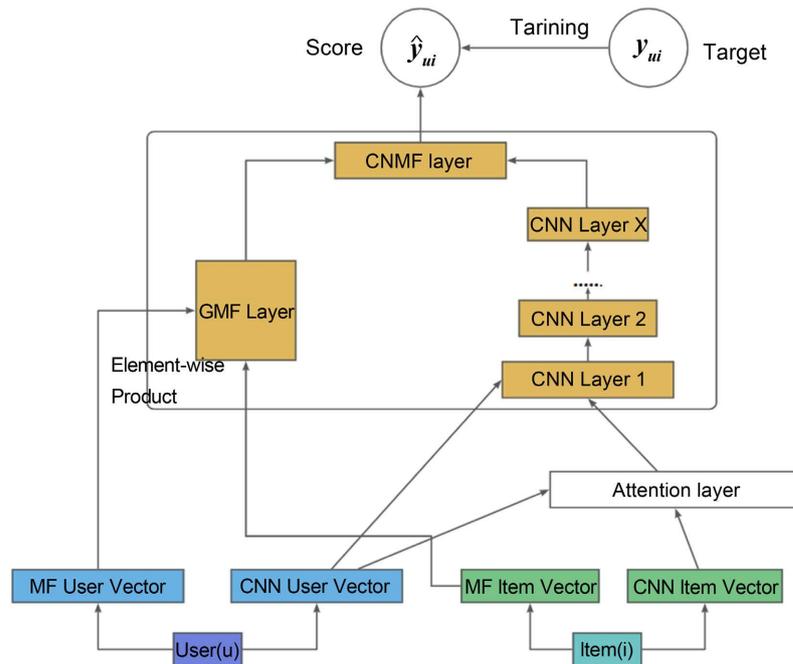


Figure 4. CNMF model
图 4. CNMF 模型

3. 实验分析

3.1. 实验数据集

本文使用数据集详细信息如表 1 所示。

Table 1. Details of the datasets

表 1. 数据集详细信息

数据集	版本	用户数量	项目数量	评分数量	评分密度
MovieLens-1M	1.0	6040	3900	1,000,209	4.468%
MovieLens-10M	2.0	69,878	10,681	10,000,054	1.321%
MovieLens-20M	2.0	138,493	27,278	20,000,263	0.946%

MovieLens 数据集是一个常用的电影推荐系统的数据集，是美国 Minnesota 大学 GroupLens 小组开发的 Movie Lens 站点所提供的数据集。该数据集包含了多个版本，其中最为常用的是 MovieLens 100K、MovieLens 1M、MovieLens 10M 和 MovieLens 20M。

MovieLens 100K 版本包含 100,000 个评分(从 1 到 5)和 943 位用户对 1682 部电影的评分数据。此外，它还包含有关电影元数据(电影类型、发布年份等)和用户的个人信息(年龄、性别等)。

MovieLens 1M 版本包含 1,000,209 个评分和 6040 名用户对 3900 部电影的评分数据。与 100K 版本相比，1M 版本提供了更丰富的信息，例如用户的请求时间戳和更多的电影元数据。

MovieLens 10M 版本是包含 10,000,054 个评分和 69,878 名用户对 10,681 部电影的评分数据。与之前的版本相比，它提供了更大的规模和更多的用户及电影。

MovieLens 20M 版本是最大的 MovieLens 数据集，包含 20,000,263 个评分和 138,493 名用户对 27,278 部电影的评分数据。除了评分数据，它还包含与电影和用户相关的元数据。

本文主要采用 MovieLens 1M、MovieLens 10M 和 MovieLens 20M 三个数据集对模型进行评估学习，对于每个数据集，取其 80% 的数据作为训练集数据，20% 作为测试集。

3.2. 评价指标

Recall@K: 本文的第一个评估指标是 Recall@K，召回率(Recall) [6] 是评估推荐系统性能的指标之一，用于衡量在给定的测试用例中，前 K 个推荐项目中包含了多少个实际需求项目。召回率不考虑项目的实际排名顺序，而是关注所需项目是否被推荐。这很好地模拟了某些实际场景，在这些场景中没有突出的建议，绝对顺序也不重要。Recall 在传统意义上的计算公式为：

$$R^{\text{Recall}} = \frac{TP}{TP + FN} \quad (12)$$

式中： TP 表示实际需求项目在前 K 个推荐项目中被正确地推荐出来的数量， FN 表示实际需求项目在前 K 个推荐项目中被正确地推荐出来的数量，Recall 度量有多个正例被分为正例。在推荐系统的个性化排序任务中，召回率的计算被定义为式(13)的形式。

$$R^{\text{Recall}} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (13)$$

式中： $R(u)$ 指的是为用户 u 推荐的 N 个物品的列表， $T(u)$ 指的是用户 u 在测试集上喜欢的物品的集合。

MRR@K：实验中使用的第二个度量是平均倒数排名(mean reciprocal rank)，这是所需项目的倒数排名的平均值。如果排名大于 K ，则倒数排名被设为 0。MRR 考虑项目的排序。计算公式为：

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{|Q|} \frac{1}{r_i^{\text{ank}}} \quad (14)$$

式中： $|Q|$ 表示用户感兴趣的项目数量， r_i^{ank} 表示用户感兴趣的项目在推荐列表中的排名。当真实值的排名大于设置的截断值时，排名的倒数设置为 0。MRR 更能反映在排序问题中推荐的质量，因为人们往往更关注推荐列表中的前几项，当真实值的排名十分靠后的时候，即使真实值在推荐列表中，也不能算高质量的推荐结果[7]。

3.3. 实验结果分析

为探究在 CNMF 模型的推荐性能,将 CNMF 模型与 NeuMF (Neural Collaborative Filtering)模型和 BPR (Bayesian Personalized Ranking)模型进行对比实验。

NeuMF 模型[5]，该模型同样基于 NCF 框架，结合了矩阵分解和神经网络的优势。它采用了矩阵分解方法将用户和物品映射到向量空间，并使用神经网络模型对用户和物品的向量进行非线性投影和交互，以预测用户对未知物品的喜好。

BPR 模型[8]通过最大化基于排名的个性化目标函数来进行推荐，而不是使用传统的回归或分类目标函数。BPR 模型主要针对隐反馈的推荐问题，例如用户喜好或评分数据的缺失情况。它通过引入排序方法，对用户与物品之间的偏好进行建模，并通过梯度下降的方式来优化模型参数。

表 2~4 是 CNMF 模型与其对比模型在 MovieLens 数据集上的 Recall@K 和 MRR@K 的结果。

通过表 2~4 我们可以看出在不同大小的数据集上，CNMF 和 NeuMF 的效果要优于 BPR。CNMF 模型使用了 CNN 与注意力机制，其推荐效果更优。通过数据结果还可以看出在数据量较大时，基于深度学习的推荐模型优势更加明显，效果更好。

Table 2. The results for Recall@K and MRR@K on the MovieLens-1m dataset

表 2. MovieLens-1m 数据集上的 Recall@K 和 MRR@K 结果

模型	Recall@5	Recall@10	Recall@20	MRR@5	MRR@10	MRR@20
BPR	0.145	0.185	0.241	0.070	0.087	0.095
NeuMF	0.178	0.246	0.323	0.090	0.112	0.121
CNMF	0.191	0.258	0.345	0.099	0.120	0.126

Table 3. The results for Recall@K and MRR@K on the MovieLens-10m dataset

表 3. MovieLens-10m 数据集上的 Recall@K 和 MRR@K 结果

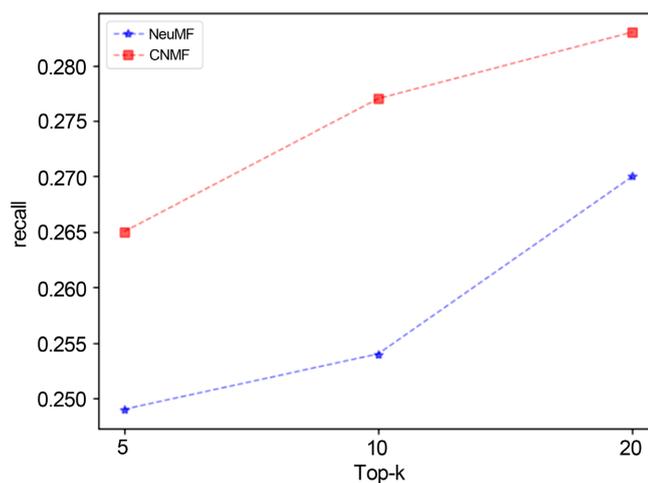
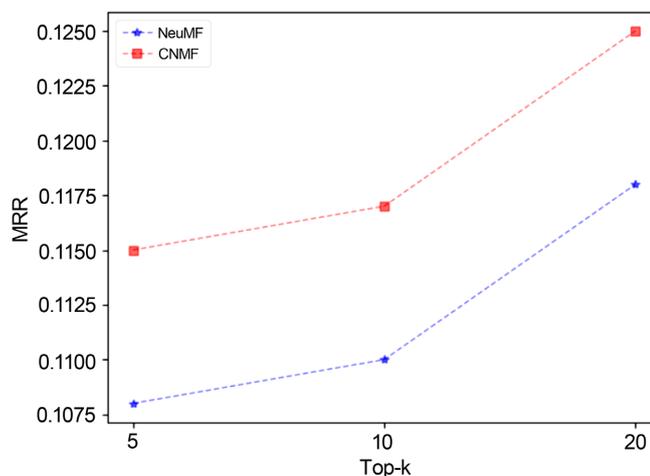
模型	Recall@5	Recall@10	Recall@20	MRR@5	MRR@10	MRR@20
BPR	0.139	0.184	0.232	0.063	0.083	0.092
NeuMF	0.188	0.251	0.324	0.091	0.116	0.124
CNMF	0.202	0.275	0.353	0.097	0.125	0.130

Table 4. The results for Recall@K and MRR@K on the MovieLens-20m dataset**表 4.** MovieLens-20m 数据集上的 Recall@K 和 MRR@K 结果

模型	Recall@5	Recall@10	Recall@20	MRR@5	MRR@10	MRR@20
BPR	0.133	0.177	0.221	0.057	0.072	0.081
NeuMF	0.205	0.267	0.338	0.098	0.123	0.133
CNMF	0.218	0.279	0.351	0.103	0.132	0.141

CNMF 和 NeuMF 是基于深度学习的推荐模型，深度学习模型能够自动地从原始数据中学习出高层次的抽象特征，并且适用于具有海量数据的推荐场景，能够实现快速的计算和训练，并在大规模数据集上表现良好。

通过图 5，图 6 也可以清晰的看出来 CNMF 相较于 NeuMF 有着较为明显的效果上的提升。也反映了当 k 值的选取越大时，效果越好。

**Figure 5.** Comparison of CNMF and NeuMF's rise in recall**图 5.** CNMF 与 NeuMF 在 recall 上提升的对比**Figure 6.** Comparison of CNMF and NeuMF's rise in MRR**图 6.** CNMF 与 NeuMF 在 MRR 上提升的对比

4. 结束语

本文在设计 CNMF 模型时, 基于 NCF 框架, 引入了 CNN 与注意力机制, 提升了原模型的推荐效果和准确度。CNN 与注意力机制对于特征的提取, 多层次的学习等能力要优于原模型, 所以可以做到在准确度和推荐效果上一个提升。Transformer 与知识图在推荐系统中已表现出巨大潜力, 相比传统的 RNN、CNN 模型, Transformer 可以将用户的短期行为序列进行建模, 通过自注意力机制计算某一时刻的项目与所有时刻项目的相关性。知识图的关联性挖掘以及其丰富的语义表示可以优化推荐算法的冷启动问题以及拓展推荐算法的多样性。下一步工作可以对 Transformer 的融合以及将知识图引入推荐算法做进一步的探索。

参考文献

- [1] 张玉叶. 基于协同过滤的电影推荐系统的设计与实现[J]. 电脑知识与技术, 2019, 15(6): 70-73.
- [2] 张鹏飞, 熊娇娇, 罗绳焯, 吴方君. 面向电商的基于协同过滤的个性化推荐[J]. 科技广场, 2016(6): 15-19.
- [3] 隋占丽, 李影, 于娟, 王波. 基于协同过滤技术的音乐推荐系统的研究[J]. 福建电脑, 2015, 31(2): 12-13+112.
- [4] 赵宇凤. 基于协同过滤的图书推荐系统[J]. 微型电脑应用, 2022, 38(1): 181-184.
- [5] He, X., Liao, L., Zhang, H., *et al.* (2017) Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web*, Geneva, 3-7 April 2017, 173-182. <https://doi.org/10.1145/3038912.3052569>
- [6] Hidasi, B., Karatzoglou, A., Baltrunas, L., *et al.* (2015) Session-Based Recommendations with Recurrent Neural Networks. <http://arxiv.org/abs/1511.06939>
- [7] Dong, X.Z., Jin, B.H., Zhuo, W., *et al.* (2021) Improving Sequential Recommendation with Attribute-Augmented Graph Neural Networks. In: Karlapalem, K., *et al.*, Eds., *Advances in Knowledge Discovery and Data Mining. Lecture Notes in Computer Science*, Vol. 12713, Springer, Berlin, 373-385. https://doi.org/10.1007/978-3-030-75765-6_30
- [8] Rendle, S., Freudenthaler, C., Gantner, Z. and Schmidt-Thieme, L. (2009) BPR: Bayesian Personalized Ranking from Implicit Feedback. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Montreal, 18-21 June 2009, 452-461.