

文章编号: 2095-2163(2020)09-0186-05

中图分类号: TP391

文献标志码: A

# 基于注意力机制的深度协同推荐模型构建

杨 希, 何月顺, 王同罕

(东华理工大学 信息工程学院, 南昌 330013)

**摘要:** 协同过滤中的矩阵分解算法在推荐系统中应用广泛,但其很难充分利用用户和项目描述文档信息,存在显式数据稀疏的问题。由于注意力机制与深度神经网络强大的特征学习能力,本文提出了基于注意力机制的深度协同推荐模型,将评分矩阵的行和列融合用户和项目文档中的描述信息后作为输入,以此缓解数据稀疏性问题;引入注意力机制,有利于神经网络层更好地提取输入向量的非线性特征。在公开的数据集上的实验表明,本文所提出的算法提高了评分预测的准确性。

**关键词:** 注意力机制; 协同推荐; 矩阵分解; 深度神经网络

## Deep collaborative recommendation model based on attention mechanism

YANG Xi, HE Yueshun, WANG Tonghan

(School of Information Engineering, East China University of Technology, Nanchang 330013, China)

**[Abstract]** The matrix decomposition algorithm in collaborative filtering is widely used in the recommendation system, but it is difficult to make full use of the information in the user and item description documents, causing the problem of explicit data sparseness. Due to the powerful feature learning ability of the attention mechanism and deep neural network, this paper proposes a deep collaborative recommendation model based on the attention mechanism, which combines the rows and columns of the scoring matrix with the description information in the user and item documents as input to ease the problem of data sparsity. The attention mechanism is helpful for the neural network layer to better extract the nonlinear characteristics of input vectors. Experiments on the public dataset show that the algorithm proposed in this paper improves the accuracy of scoring prediction.

**[Key words]** attention mechanism; collaborative recommendation; matrix factorization; deep neural network

## 0 引言

随着互联网技术的高速发展,各类信息数据呈现爆炸式增长的趋势,“信息过载”导致用户很难从互联网准确获取到感兴趣的资源。推荐系统作为满足用户的个性化需求,使用户能够快速有效地获取信息的重要工具,近年来被广泛应用于电影视频、音乐推荐、新闻推送及电子商务等领域。

协同过滤(Collaborative Filtering, CF)是推荐系统中应用最成功和最广泛的算法之一<sup>[1]</sup>,其核心思想是利用与目标用户兴趣偏好相同的用户群体的喜好情况来实现推荐<sup>[2]</sup>。在不同类型的协同过滤算法中,矩阵分解(Matrix factorization, MF)因其简单有效的特性而被广泛采用。传统的矩阵分解及其改进方法大多是基于浅层的模型,由于系统中用户和项目的数量非常大,而每个用户只与少数项目有交互,模型的推荐准确性受到稀疏的评分矩阵影响。

近年来,很多学者将深度学习与传统的推荐算

法相结合,利用深度学习提取高层特征、融合多源信息的优势,改善传统算法的推荐性能。Suvash等构建基于自动编码器的推荐模型 AutoRec,对输入的用户或项目向量进行重构,最小化重构误差实现模型的训练<sup>[3]</sup>;He等提出了神经协同过滤(NCF)模型,使用多层感知器对隐式数据建模,提取非线性特征,结合矩阵分解提取线性特征,使用神经网络结构得到模型的输出<sup>[4]</sup>;张等使用二次多项式回归模型获取潜在特征向量,将其输入深度神经网络模型中,得到预测评分<sup>[5]</sup>。上述模型在一定程度上提高推荐的效果,但仅使用评分信息,未充分利用用户和项目描述文档,存在数据稀疏性问题。

随着注意力机制在机器翻译、图像分类等领域的广泛应用,其在推荐领域也备受关注。王等提出了融合注意力机制的深度推荐算法,在项目级和特征级的隐式反馈中分别引入注意力机制取得了较好的推荐效果<sup>[6]</sup>;谢等将注意力机制应用于深度神经

**基金项目:** 国家自然科学基金(41872243)。

**作者简介:** 杨 希(1993-),女,硕士研究生,主要研究方向:深度学习、推荐系统;何月顺(1971-),男,博士,教授,硕士生导师,主要研究方向:无线传感网络与物联网技术、大数据分析 with 智能信息处理;王同罕(1984-),男,博士,讲师,硕士生导师,主要研究方向:模式识别与人工智能。

**通讯作者:** 何月顺 Email: heys@ecit.cn

**收稿日期:** 2020-07-25

网络,用以提取隐式反馈数据中的潜在特征<sup>[7]</sup>;Xiao等提出了将注意力机制与因子分解机融合的 AFM 模型,使用注意力机制学习交叉特征的权重<sup>[8]</sup>。

由于注意力机制与深度神经网络强大的特征学习能力,本文提出一种基于注意力机制的深度协同推荐模型(Attention-based Deep Collaborative Recommendation Model, A-DCRM),以评分矩阵的行和列作为输入,融合用户和项目的文档描述信息以缓解数据稀疏性问题;将注意力机制与神经网络相结合,更好地提取用户和项目的非线性特征,在输出层构建 Hadamard 乘积并以线性加权方式输出,提高模型的泛化能力。

### 1 协同推荐模型的算法基础

推荐系统不仅能满足用户高效获取信息的需求,还能为企业最大限度地吸引用户、留存用户,从而带来商业利益。本节将对矩阵分解、多层感知器和注意力机制进行介绍。

#### 1.1 矩阵分解

矩阵分解模型假设用户对项目的偏好可以由少数特征决定,基于矩阵分解的协同过滤推荐可以描述为:将原始的评分矩阵分解为隐含类别个数相同的低维用户和项目矩阵,然后将两者相乘来还原原始的稀疏矩阵,以得到用户对未交互项目的评分。假设用户数为  $M$ ,项目数为  $N$ ,那么用户对项目的评分矩阵可以表示为  $R \in R^{M \times N}$ ,其中的非空元素  $r_{ij}$  则表示用户  $i$  对项目  $j$  的评分值。在将高维矩阵分解后,用户  $i$  的隐向量表示为  $p_i \in R^{1 \times k}$ ,项目  $j$  的隐向量表示为  $q_j \in R^{k \times 1}$ , $k$  表示隐含类别的维度。通过  $p_i$  和  $q_j$  的向量内积,得到评分  $r_{ij}$  的近似值  $\hat{r}_{ij}$ ,用公式(1)表示;通过最小化实际评分与预测评分之间的误差来学习  $p_i$  和  $q_j$ ,用公式(2)表示:

$$\hat{r}_{ij} = p_i q_j, \tag{1}$$

$$\min: L = \sum_{r_{ij} \neq 0} (r_{ij} - \hat{r}_{ij})^2 + \lambda (\|p_i\|^2 + \|q_j\|^2). \tag{2}$$

其中,后一项增加正则化项防止过拟合, $\lambda$  为正则化参数。

#### 1.2 多层感知器

深度神经网络也称为多层感知器(Multi-layer Perception, MLP),将其应用于推荐中,主要是利用其深层特征提取能力。多层感知器一般由输入层、隐含层和输出层构成,层与层之间是全连接的。假设  $x_l$  为网络第  $l$  层的输出,其计算过程为式(3):

$$x_l = \sigma(x_{l-1} W_l + b_l). \tag{3}$$

其中,  $x_{l-1}$  表示第  $l-1$  层的输出; $W_l$  和  $b_l$  表示第  $l$  层的权重和偏差项; $\sigma(\cdot)$  是激活函数。

多层感知器通过前向传播得到网络的输出,并使用反向传播算法进行参数的更新。

### 1.3 注意力机制

注意力机制一定程度上模拟了人类视觉的特点,在观察事物时往往会有重点关注目标,利用注意力机制可以帮助神经网络专注于与目标任务更加相关的特征信息。其基本原理如图 1 所示,将需要处理的数据  $S$  构成一系列的键值对  $\langle \text{Key}, \text{Value} \rangle$ ,  $\text{Key}$  是属性,  $\text{Value}$  表示对应的值,给定目标中的某个属性  $Q$ ,通过计算  $Q$  与  $\text{Key}$  之间的相关性得到  $\text{Value}$  对应的权重系数,然后对  $\text{Value}$  进行加权处理,得到最终的 Attention 值。

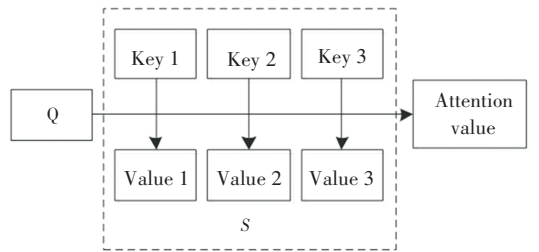


图 1 注意力机制原理图

Fig. 1 Schematic diagram of attention mechanism

计算过程如式(4)~(6):

$$a_i = \text{softmax}(\text{sim}(Q, \text{Key}_i)), \tag{4}$$

$$\text{softmax}(x_i) = e^{x_i} / \sum_{i=1}^k e^{x_i}, \tag{5}$$

$$\text{Atten} = \sum_{i=1}^n a_i \text{Value}_i. \tag{6}$$

其中,公式(4)中  $\text{sim}(\cdot)$  表示相关性计算函数,一般采用点乘、缩放点乘或 MLP 算法计算; $\text{softmax}(\cdot)$  为归一化函数,其计算过程如公式(5)所示; $a_i$  表示每个  $\text{Value}$  对应的权重系数; $\text{Atten}$  为最终得到的注意力值。

### 2 模型研究与构建

本文基于注意力机制和深度神经网络构建的深度协同推荐模型 A-DCRM 结构如图 2 所示,包括输入层、嵌入层、注意力层、神经网络层和输出层。

#### 2.1 输入层

与隐式反馈数据相比,显式反馈数据能够直接反映用户的兴趣偏好,数据噪声较少,可信度较高,本文使用显式数据作为输入进行训练。与大多数将用户和项目的 ID 经过 one-hot 编码处理后作为输入不同,本文使用用户对项目的评分数据构建评分

矩阵  $\mathbf{R} \in \mathbf{R}^{M \times N}$ ,  $M$  表示用户数量,  $N$  表示项目数量, 评分矩阵按公式(7)构建, 其中  $r_{ij}$  表示用户  $i$  对项目  $j$  的评分值, 如果没有评分值则以 0 填充。

$$\mathbf{R} = \begin{cases} r_{ij}, & \text{if } r_{ij} = \text{已知}; \\ 0, & \text{其它}. \end{cases} \quad (7)$$

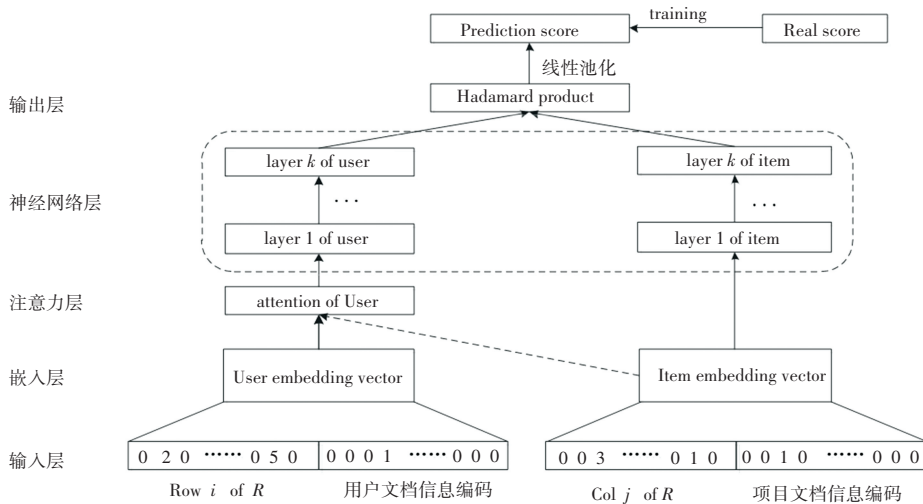


图2 A-DCRM 模型结构

Fig. 2 Model structure of A-DCRM

由于矩阵  $\mathbf{R}$  的行表示用户对所有项目的评分情况, 在一定程度上代表用户的全局偏好,  $\mathbf{R}$  的列表示项目由所有用户打分, 在一定程度上代表项目的总体评价情况, 所以使用矩阵  $\mathbf{R}$  的行和列作为用户和项目的输入, 分别记为  $R_{user}$  和  $R_{item}$ 。此外, 用户和项目描述文档中的静态信息可作为评分数据的补充。利用 one-hot 编码对用户的年龄、性别和职业、项目的类型特征等离散数据特征进行处理, 得到用户和项目的静态特征, 分别记为  $U_{tag}$  和  $I_{tag}$ 。最后使用拼接函数对用户和项目的矩阵数据和静态特征进行融合, 计算过程为式(8):

$$\begin{cases} \mathbf{x}_{user} = R_{user} \oplus U_{tag}, \\ \mathbf{x}_{item} = R_{item} \oplus I_{tag}. \end{cases} \quad (8)$$

其中,  $\oplus$  表示对两个向量的拼接操作,  $\mathbf{x}_{user}$  和  $\mathbf{x}_{item}$  分别表示用户和项目经过拼接融合后的特征向量, 以此作为模型的输入。

## 2.2 嵌入层

为减少输入层特征向量的维度, 便于神经网络更好地提取向量特征, 利用嵌入层将输入层的高维稀疏向量映射到低维空间, 得到可以表示用户和项目潜在特征的低维稠密向量。嵌入层的计算过程为式(9):

$$\begin{cases} \mathbf{x}_{user\_embed} = \sigma(\mathbf{x}_{user} \mathbf{W}_u), \\ \mathbf{x}_{item\_embed} = \sigma(\mathbf{x}_{item} \mathbf{W}_i). \end{cases} \quad (9)$$

其中,  $\mathbf{W}_u$  和  $\mathbf{W}_i$  分别为用户特征向量和项目特征向量的嵌入层权重矩阵;  $\sigma(\cdot)$  是非线性激活函

数;  $\mathbf{x}_{user\_embed}$  和  $\mathbf{x}_{item\_embed}$  表示经过嵌入层处理过的用户和项目的潜在特征向量。

## 2.3 注意力层

注意力层利用用户和项目的潜在特征向量获取用户向量中每一维度特征对目标任务的影响程度, 帮助神经网络层更好地学习向量的深层特征。此过程主要分为二步, 首先根据公式(10)获取用户潜在特征向量每个维度对应的关注度。

$$A_{atten} = \text{softmax}(\mathbf{x}_{user\_embed} \odot \mathbf{x}_{item\_embed}). \quad (10)$$

其中,  $\odot$  表示 Hdamard 积操作;  $\text{softmax}(\cdot)$  函数通过归一化计算获取每一维度对应的注意力概率值;  $A_{atten}$  表示各个维度对应的关注度。然后将各维度对应的关注度与用户潜在特征向量相乘, 得到注意力层的输出, 式(11)。

$$\mathbf{x}_{out} = A_{atten} \odot \mathbf{x}_{user\_embed}, \quad (11)$$

其中,  $\mathbf{x}_{out}$  用于输入神经网络层非线性特征提取。

## 2.4 神经网络层及输出层

神经网络层采用多层感知器学习输入的潜在特征向量中的非线性特征。将用户特征向量  $\mathbf{x}_{out}$  和项目特征向量  $\mathbf{x}_{item\_embed}$  分别输入多层感知器, 计算过程如公式(3)所示, 将经过神经网络层处理后的用户潜在特征向量和项目潜在特征向量分别表示为  $\mathbf{p}_u$  和  $\mathbf{q}_i$ , 在输出层融合时, 利用 Hdamard 积操作对用户和项目的潜在特征向量进行计算, 并通过线性加权的方式融合, 最终得到模型的预测值。数学表

达式(12)如下:

$$\hat{y}_{ui} = W(p_u \odot q_i), \quad (12)$$

其中,  $W$  表示输出层的权重,  $\hat{y}_{ui}$  表示模型的预测值。

### 2.5 损失函数

本文采用逐点的平方损失来计算模型预测值与目标值之间的误差,模型通过梯度下降方法优化,以达到最小化损失函数的目的。损失函数的数学表达式可表示为式(13):

$$Loss = \frac{1}{2} \sum_{u,i \in \Omega} (y_{ui} - \hat{y}_{ui})^2 + \lambda \|W\|_F^2, \quad (13)$$

其中,  $y_{ui}$  表示用户  $u$  对项目  $i$  的真实评分;  $\hat{y}_{ui}$  表示模型的预测值;  $W$  表示模型各层中的参数集合,后一项表示分别对该集合中的参数进行正则化约束以防止过拟合;  $\lambda$  为正则化系数。

## 3 实验与结果分析

### 3.1 实验数据集

本文采用公开数据集 Movielens 100K 和 Movielens 1M 进行算法性能对比实验, Movielens 数据集是电影评分数据集,所有评分值为 1~5 的整数,分值越高代表用户对该电影的喜好值越高。Movielens 100K 中是 943 个用户对 1 682 个项目的 100 000 条评分记录, Movielens 1M 是 6 040 个用户对 3 952 个项目的评分数据,一共有 1 000 209 条记录,两个数据集中每个用户至少评价 20 部电影,详细信息见表 1。

表 1 Movielens 数据集描述

Tab. 1 Description of Movielens datasets

数据集	用户数	项目数	评分数	稀疏度/%
Movielens 100K	943	1682	100 000	93.7
Movielens 1M	6040	3952	1 000 209	95.8

运行环境为: Windows10 64 位操作系统, CPU 为 Intel i7-8550U, 显卡 NVIDIA GeForce MX150, 本文模型基于 tensorflow 构建, 在实验过程中, 将每个数据集随机地切分为两个部分, 选择 90% 作为训练集, 10% 作为测试集, 实验 5 次后取平均值来评估评分预测准确性和模型性能。

### 3.2 评价指标

均方误差 ( $RMSE$ ) 和平均绝对误差 ( $MAE$ ) 是推荐系统中常用的度量评分预测准确度的评价指标, 将使用这两个指标来评估本文提出的 A-DCRM 模型的性能。计算表达式为式(14)和式(15):

$$RMSE = \sqrt{\frac{\sum_{u,i \in T} (y_{ui} - \hat{y}_{ui})^2}{|T|}}, \quad (14)$$

$$MAE = \frac{\sum_{u,i \in T} |y_{ui} - \hat{y}_{ui}|}{|T|}. \quad (15)$$

其中,  $T$  表示测试集,  $|T|$  表示测试集的大小。

### 3.3 模型训练参数

基于深度学习构建的模型涉及到大量需要训练的参数, 本文用标准正态分布进行参数初始化, 采用 Adam 优化器进行参数优化, 学习率设置为 0.000 5, 正则化参数设置为 0.5, 在嵌入层中用户和项目的潜在特征向量取相同维度, 设置为 250 维。神经网络中多层感知器采用塔式结构, 最高层数设置为 3, 每一层的维度为上一层的一半。

### 3.4 实验结果与分析

将本文提出的 A-DCRM 和当前一些具有代表性的推荐算法进行对比, 其中包括在 SVD 基础上考虑隐式反馈信息的 SVD++ 算法, 通过自编码器对输入的项目向量进行重构的 I-AutoRec 算法, 融合边缘降噪自编码器与 MF 的 mSDA-MF, 结合多层感知器和矩阵分解的 NCF 算法。各算法在  $RMSE$  和  $MAE$  两个指标上的对比结果见表 2。

表 2 不同推荐算法在 Movielens 数据集上的结果对比

Tab. 2 Results comparison of different algorithms on Movielens datasets

算法	Movielens 100K		Movielens 1M	
	$RMSE$	$MAE$	$RMSE$	$MAE$
SVD++	0.920	0.722	0.862	0.673
I-AutoRec	0.905	0.712	0.841	0.699
mSDA-MF	0.919	0.741	0.852	0.672
NCF	0.897	0.721	0.861	0.673
A-DCRM	0.884	0.694	0.840	0.661

从表 2 中可以看出, 各算法在 Movielens 1M 上的实验结果均优于 Movielens 100K, 这表明数据集越大, 算法可以学习到更多的用户和项目特征信息, 从而使预测评分更准确。基于深度神经网络的算法 I-AutoRec、NCF 和 A-DCRM 的效果比浅层的 SVD++ 效果好, 这在一定程度上表明了深度学习技术在特征提取方面的优势。从表 2 中 Movielens 100 K 的结果可以看出, 本文提出的 A-DCRM 模型相较于其他的算法而言, 在  $RMSE$  指标上效果最少提升 1.3%, 最多提升了 3.6%, 在  $MAE$  指标上最少提升 1.8%, 最多提升 4.7%, 在 Movielens 1M 数据集上两个 (下转第 196 页)