



计算机工程与应用  
Computer Engineering and Applications  
ISSN 1002-8331, CN 11-2127/TP

## 《计算机工程与应用》网络首发论文

题目：基于深度学习的多交互混合推荐模型  
作者：李同欢，唐雁，刘冰  
网络首发日期：2018-09-15  
引用格式：李同欢，唐雁，刘冰. 基于深度学习的多交互混合推荐模型. 计算机工程与应用. <http://kns.cnki.net/kcms/detail/11.2127.TP.20180913.0636.010.html>



**网络首发：**在编辑部工作流程中，稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定，且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式（包括网络呈现版式）排版后的稿件，可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定；学术研究成果具有创新性、科学性和先进性，符合编辑部对刊文的录用要求，不存在学术不端行为及其他侵权行为；稿件内容应基本符合国家有关书刊编辑、出版的技术标准，正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性，录用定稿一经发布，不得修改论文题目、作者、机构名称和学术内容，只可基于编辑规范进行少量文字的修改。

**出版确认：**纸质期刊编辑部通过与《中国学术期刊（光盘版）》电子杂志社有限公司签约，在《中国学术期刊（网络版）》出版传播平台上创办与纸质期刊内容一致的网络版，以单篇或整期出版形式，在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊（网络版）》是国家新闻出版广电总局批准的网络连续型出版物（ISSN 2096-4188，CN 11-6037/Z），所以签约期刊的网络版上网络首发论文视为正式出版。

# 基于深度学习的多交互混合推荐模型

李同欢<sup>1</sup>, 唐 雁<sup>2</sup>, 刘 冰<sup>1,2</sup>

LI Tonghuan<sup>1</sup>, TANG Yan<sup>2</sup>, LIU Bing<sup>1,2</sup>

1.西南大学 计算机与信息科学学院, 重庆 400715

2.达州职业技术学院, 四川 达州 635001

1. School of Computer and Information Science, Southwest University, Chongqing 400715, China

2. Dazhou Vocational And Technial College, Dazhou, Sichuan 635001, China

**LI Tonghuan, TANG Yan, LIU Bing. Multi-interaction Hybrid Recommendation Model Based on Deep Learning. Computer Engineering and Applications**

**Abstract:** In the traditional recommendation systems, the approach of matrix factorization collaborative filtering only just considers the single information of rating, it as a shallow model, can hardly learn deeper feature information. This paper propose a multi-interaction deep matrix factorization model based on auxiliary information, firstly through deep learning model and merge more auxiliary information as input, effectively alleviate the problem of data sparsity. Then, we leverage the structure of multi-interactive non-linear network to learn the deep feature representation of more abstract and dense; through an inner product interactions on the latent features of users and items repeatedly, to obtain the different layers of feature representation results; the finally aggregating all the interaction results to predict. The experiment results on the MovieLens latest 100K dataset shown that the proposed model over the state-of-the-art methods in RMSE.

**Key words:** Collaborative Filtering; Deep Learning; Auxiliary Information; Multi-interaction; Neural Network; Recommendation System

**摘 要:** 传统的推荐系统中, 基于矩阵分解的协同过滤方法只考虑单一的评分信息, 而且作为浅层模型无法学习到更深层次的特征信息。文中提出一种基于深度学习的多交互混合推荐模型, 首先通过深度学习模型融合更多的辅助信息作为输入, 能够缓解数据的稀疏性问题; 然后利用多层交互的非线性网络结构去学习更抽象、稠密的深层次特征表示; 通过对用户和项目的隐表示进行多次内积交互获得不同层次的特征表示结果; 最后聚合所有的交互结果进行预测。在 Movieles latest 100K 数据集上进行实验, 采用 RMSE 指标进行评估, 结果表明所提模型在推荐效果上有所提升。

**关键词:** 协同过滤; 深度学习; 辅助信息; 多层交互; 神经网络; 推荐系统

**文献标志码:** A **中图分类号:** TP391.4 **doi:** 10.3778/j.issn.1002-8331.1803-0132

**基金项目:** 四川省教育厅自然科学重点项目(No.18ZA0217); 中央高校基本科研业务费专项资金(No.XDJK2015C110); 重庆市 2017 年度中小学创新人才工程项目(No.CY170217); 教育部“春晖计划”资助项目(No. Z2011149)。

**作者简介:** 李同欢(1994—), 男, 硕士, CCF 学生会员, 主要研究方向为推荐算法研究, E-mail: 524758503@qq.com; 唐雁(1965—), 通讯作者, 女, 教授, 主要研究方向为数据挖掘、智能科学, E-mail: ytang@swu.edu.cn; 刘冰(1970—), 男, 副教授, 主要研究方向为计算机视觉、数据挖掘。

## 1 引言

随着信息爆炸式地增长<sup>[1]</sup>, 如何从大量数据中发掘和利用有效的信息成了关键问题。推荐系统在缓解信息过载方面起着重要的作用<sup>[2]</sup>, 被许多在线服务所广泛采用, 如电子商务、在线新闻和社交媒体网站等。推荐系统<sup>[3]</sup>主要根据用户需求、偏好等信息, 通过推荐算法来获得用户对物品的偏好特征, 并据此向用户进行物品推荐。

传统的推荐系统算法<sup>[2]</sup>主要包括基于内容(Content-Based)的推荐算法<sup>[4]</sup>和基于协同过滤(CF)的推荐算法<sup>[5]</sup>。前者是通过抽取内容特征来描述物品, 然后通过用户过去的喜好特征描述偏好, 最后通过计算用户与项目之间的相关性进行推荐。后者是通过获取用户对项目的显式或隐式信息, 根据用户对项目的偏好来发现用户或项目的相关性, 最后基于这些相关性进行推荐。

个性化推荐系统中, 协同过滤算法应用更为广泛, 特别是矩阵因子分解<sup>[6]</sup> (MF), SVD++<sup>[7]</sup>等方法在 Netflix 大赛中屡创佳绩, 将基于协同过滤的推荐算法推上高潮。但是基于矩阵分解的算法往往只考虑用户与项目单一的交互信息(评分数据), 这也遭遇到严重的数据稀疏性问题(每个用户所评分项目只占项目总数的极少部分)和冷启动问题(新用户与新项目往往没有评分数据)。另外, 作为一个浅层模型也无法学习到用户与项目更深层次的特征。而基于内容的方法依赖人工设计特征的有效提取, 限制了其扩展性和有效性。

近年来, 由于强大的表示学习能力, 深度学习在图像处理、自然语言处理和语音识别等领域取得了巨大的突破与成就<sup>[8]</sup>, 也为推荐系统领域的研究带来新的机遇。2007 年 Salakhutdinov<sup>[8]</sup>等人首次将深度学习应用于推荐系统, 并提出了一种基于受限玻尔兹曼机(RBM)的协同过滤模型。通过对用户的评分数据进行重构去学习用户的隐表示并进行评分预测。2013 年 Huang<sup>[9]</sup>等人提出了深度结构化语义模型(DSSM), 2015 年 Elkahky<sup>[10]</sup>等人考虑到传统基于内容的推荐系统中用户特征难以获取的问题, 进一步提出多视角深度神经网络模型(MV-DNN)。2017 年薛洪建<sup>[11]</sup>等人提出了深度矩阵分解模型, 通过重构评分矩阵分别对用户和项目进行多层神经网络训练, 学习用户与项目的隐表示, 最后通过余弦相关度进行推荐。2016 年谷歌<sup>[12]</sup>发布了广泛&深度学习方法(Wide&Deep learning), 在特征嵌入层之上加上多层感知机(MLP), 表示出很强的泛化能力。2017 年何向南<sup>[13]</sup>等人进一步结合多层感知机(MLP)模型和广义矩阵分解(GMF)模型提出了神经协同过滤(NCF)的算法, 进一步提升了推荐模型的性能。上述方法都通过深度学习获取用户与项目更加抽

象、稠密的深层次表示; 利用更深层次的神经网络去构建模型, 获取了用户与项目之间交互的非线性结构特征。但上述算法模型大多数都是基于矩阵分解的思想, 只考虑了用户与项目的评分数据。另外, 上述算法模型都是在嵌入层之后将用户与项目的隐因子向量分别送入多层神经网络进行特征学习, 或者直接连接用户与项目的隐因子向量再经过多层神经网络进行特征学习, 这样在深层网络中训练时或许会丢失一部分重要的特征表示。

针对以上推荐模型的问题, 提出融合辅助信息的多交互深度矩阵分解模型 MA-DMF(Multi-interaction Deep Matrix Factorization Model Based on Auxiliary Information)。在用户-项目交互的评分数据基础上, 融合项目属性和标签等辅助信息, 经过 One-hot 编码向量分别作为输入层数据; 通过嵌入层分别获得用户、项目及其辅助信息的隐因子向量; 连接用户、项目及其对应的辅助信息隐因子向量作为模型隐藏层的输入; 分别通过并行的深层网络进行学习, 获得用户与项目的隐表示。在深层网络学习时, 经过每一层网络都对用户及项目的隐表示进行一次内积交互, 获得不同层所学习的特征表示, 最后聚合所有层的内积交互结果进行评分预测。在 Movielens latest 100K 数据集上的实验结果表明, 该推荐模型能更准确地进行评分预测, 提升推荐系统的性能。

## 2 相关知识

### 2.1 矩阵分解模型

基于矩阵分解的协同过滤推荐算法, 通过学习用户和项目的低维向量表示来进行推荐。而基于深度学习的推荐系统<sup>[14]</sup>主要是利用显式或隐式的反馈数据, 将用户的评分向量或项目的被评分向量作为输入, 利用深度学习模型学习用户或项目的隐表示, 然后利用逐点损失(point-wise loss)和成对损失(pair-wise loss)等类型的损失函数构建目标优化函数对深度学习模型的参数进行优化, 最后利用学习到的隐表示进行项目推荐。

假设有  $M$  用户  $U=\{u_1, u_2, \dots, u_M\}$ ,  $N$  个项目  $I=\{i_1, i_2, \dots, i_N\}$ , 用户与项目的交互矩阵(评分数据)为  $Y \in R^{M \times N}$ 。通过交互矩阵来获得反馈信息如下:

$$y_{ui} = \begin{cases} 1, & \text{用户 } u \text{ 与项目 } i \text{ 之间有交互} \\ 0, & \text{其他} \end{cases} \quad (1)$$

公式(1)中,  $y_{ui}=1$  表示用户  $u$  与项目  $i$  之间存在交互情况, 即存在用户  $u$  对项目  $i$  的评分, 而 0 则表示他们之间没有交互信息。

基于模型的推荐系统方法<sup>[10]</sup>是定义一个模型, 然后计算  $Y$  中未观察项的一个预测评分, 生成评分数据如下:

$$\bar{y}_{ui} = F(u, i | \theta) \quad (2)$$

公式(2)中,  $\bar{y}_{ui}$  表示  $Y_{ui}$  中用户  $u$  对项目  $i$  的预测评分,  $\theta$  表示这个模型的参数,  $F$  表示映射模型参数到预测评分的函数。

为了定义函数  $F$ , Koren<sup>[7]</sup>等人在 2009 年提出了隐因子模型(LFM), 通过将用户的隐表示  $p_u$  与项目的隐表示  $q_i$  做内积操作来预测  $\bar{y}_{ui}$ 。

$$\bar{y}_{ui} = F^{LFM}(u, i | p_u, q_i) = p_u^T q_i = \sum_{k=1}^K p_{uk} q_{ik} \quad (3)$$

公式(3)中  $K$  表示隐空间的维度(即隐因子的个数),  $p_u$  与  $q_i$  分别表示用户与项目的隐因子向量。

传统的基于神经网络的矩阵分解模型如图 1 所示。两个输入特征向量分别表示用户与项目的 ID 信息经过 One-hot 编码转换成的二进制稀疏向量, 长度分别为用户总数  $M$  与项目总数  $N$ 。该矩阵分解模型通过嵌入层获得用户与项目的隐因子向量并进行内积交互, 每个用户和项目在隐空间的维度都是相互独立并用相同权值线性联合的。所以, 该模型也可以视为一个隐因子线性模型。

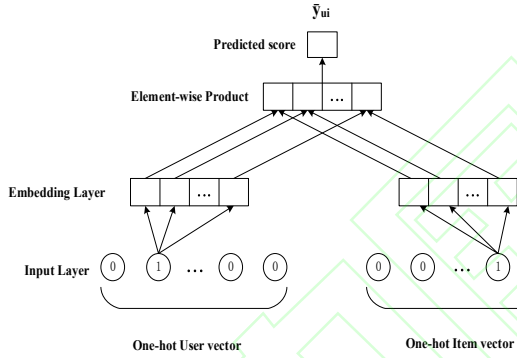


图 1 基于神经网络的矩阵分解模型

## 2.2 深度神经网络模型

最近, 何向南<sup>[13]</sup>等人证明了矩阵分解模型只是在低维空间通过简单的内积操作来估计用户与项目之间复杂的交互情况, 存在一定的限制性。虽然可以通过设置更多的隐因子个数  $K$  来缓解这个问题, 但这样会破坏模型的泛化能力(例如: 过拟合问题), 特别是在数据稀疏的情况下。

将深度神经网络(DNNs)<sup>[14,15,16]</sup>应用于推荐系统, 已经被证明具有强大学习特征表示的能力, 有潜力去学习用户与项目之间的非线性交互特征。可以通过多层感知机去学习用户与项目间的交互函数  $F$ , 来解决传统矩阵分解中的限制性问题。

在基于多层感知机(MLP)的推荐模型<sup>[13,16,17]</sup>中, 通过连接用户与项目的隐因子向量, 经过一个多层的神经网络来学习用户与项目隐特征之间的

交互信息。该模型在连接向量之上增加隐层让模型更灵活、非线性地学习  $p_u$  与  $q_i$  之间的交互, 而不像传统的矩阵分解直接进行内积交互作为输出结果。

通过多层感知机去建模用户与项目的交互如图 2 所示, 将用户与项目的 ID 信息经过 One-hot 编码后的稀疏向量作为输入层,  $v_u^U$  和  $v_i^I$  分别表示用户  $u$  和项目  $i$  的特征向量。在输入层之上是嵌入层, 作为一个完全连接层, 嵌入层将稀疏的表示映射为一个稠密向量; 嵌入层之后的向量视为用户与项目在隐因子模型下的隐因子向量。在嵌入层之上为一个多层神经网络结构, 每一层都去学习发现用户与项目交互的隐结构, 最后将隐向量映射为预测值。

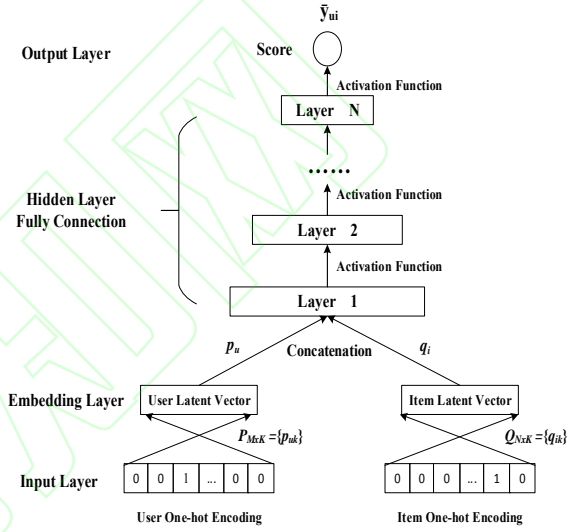


图 2 基于多层感知机的推荐模型

我们定义经过嵌入层之后用户的隐向量  $p_u$  为  $P^T v_u^U$ , 项目的隐向量  $q_i$  为  $Q^T v_i^I$ 。第一个神经网络层的映射函数为:

$$\Phi_1(p_u, q_i) = p_u \odot q_i \quad (4)$$

公式(4)中  $\odot$  表示向量对应元素相乘。我们将向量经过多层神经网络学习后映射到输出层:

$$\bar{y}_{ui} = a_{out}(h^T(p_u \odot q_i)) \quad (5)$$

公式(5)中  $a_{out}$  和  $h$  分别表示输出层的激活函数和边权重。模型的性能由最后的隐层  $N$  所决定, 计算 MLP 推荐模型可视为:

$$\bar{y}_{ui} = f(P^T v_u^U, Q^T v_i^I | P, Q, \Theta_f) \quad (6)$$

公式(6)中  $P \in R^{M \times N}$ ,  $Q \in R^{M \times N}$ , 分别表示用户与项目的隐因子矩阵,  $\Theta_f$  表示交互函数  $f$  的模型参数。由于函数  $f$  定义为一个多层神经网络, 它可以计算为:

$$f(P^T v_u^U, Q^T v_i^I) = \Phi_{out}(\Phi_X(\dots \Phi_2(\Phi_1(P^T v_u^U, Q^T v_i^I))\dots)) \quad (7)$$

公式(7)中  $\Phi_{out}$  和  $\Phi_X$  分别表示输出层和第  $X$  层



的映射函数,  $X$  为隐藏层的数目。

在 Huang 等人提出深度语义结构模型(DSSM)的基础上, Wang 等人提出了另一种基于协作的深度神经网络推荐模型<sup>[9,11,18]</sup>。通过构建两个并行的深度神经网络模型去分别学习用户和项目的隐特征, 最后在两个网络的输出层之上构建一个交互层来进行评分预测。

### 3 融合辅助信息的多交互深度矩阵分解模型

在上面所提的模型中, 大多数只考虑了评分数据, 通过简单连接用户与项目的隐向量然后进入深度神经网络中进行训练。有些模型虽然融入了内容信息, 但只是在最后一层进行了用户与项目的交互计算。我们认为, 在深度神经网络学习用户与项目的特征时, 每一层网络的学习能力不同, 学习的特征表示也不相同, 所抽取的特征代表意义也不相同, 我们应该重视每一层的训练结果, 对每一层进行一次用户与项目的交互操作。另外, 融合辅助信息可以一定地缓解数据稀疏性问题, 在辅助信息中也包含了用户与项目一定的隐藏偏好特征。我们的模型还可以进一步融合更多的跨域信息, 我们将在下一步工作中进行研究。我们所提模型如图 3 所示

#### 3.1 MA-DMF 模型

在模型中, 在用户对项目的评分信息基础上, 融合项目属性和标签等辅助信息作为输入数据。定义  $u$  和  $v$  分别表示用户和项目的 ID 经过 One-hot 编码的稀疏向量, 用  $u'$  和  $v'$  分别表示从项目属性或标签等辅助信息中抽取的用户与项目的特征向量。在输入层的向量中,  $u$  和  $v$  的维度分别为用户总数  $M$  和项目总数  $N$ ,  $u'$  和  $v'$  的维度分别表示用户与项目辅助信息中抽取的特征总数,  $u_i' = 0$  表示用户不具有  $i$  属性,  $v_j' = 1$  表示项目具有  $j$  属性。

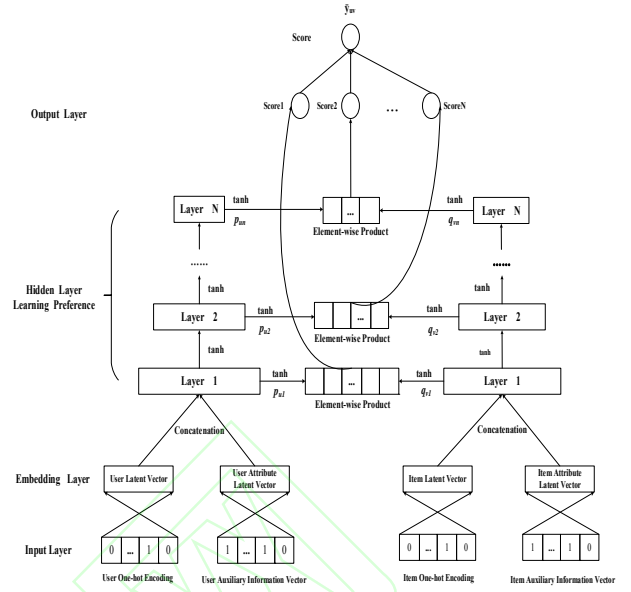


图 3 MA-DMF 推荐模型

在输入层之上为嵌入层, 嵌入层作为一个完全连接层, 将输入层中稀疏向量中的非 0 特征映射为一个稠密的向量表示, 帮助去学习用户与项目 ID 信息及其辅助信息的隐向量, 获得的嵌入向量可视作隐因子模型中的隐因子向量。分别连接用户与项目的 ID 信息和辅助信息的隐向量作为隐藏层的输入的特征向量。

在嵌入层之上是由多个完全连接层组成的隐藏层, 整个隐藏层为一个塔式结构, 通过两个并行的深度神经网络结构去分别抽取用户和项目的隐特征, 学习不同的隐特征表示。我们定义模型中隐藏层的计算如下:

$$Z_1 = F_1(W_1 x_1) \quad (8)$$

$$Z_2 = F_2(W_2 Z_1 + b_2) \quad (9)$$

$$\dots\dots\dots Z_i = F_i(W_i Z_{i-1} + b_i) \quad (10)$$

公式(10)中  $i$  表示隐藏层中神经网络的层数,  $Z_i$ ,  $F_i$ ,  $W_i$ ,  $b_i$  分别表示第  $i$  层的输出值, 第  $i$  层的激活函数, 第  $i$  层的权值矩阵和第  $i$  层的偏置向量。在模型的非线性激活函数选择中, 我们尝试使用 sigmoid, tanh 和 relu 作对比实验, 最后结果表明选用 tanh 函数作为我们的非线性激活函数能表现出更好的性能。

$$F(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (11)$$

在深度神经网络学习过程中, 我们认为每一层

网络的学习能力不同,所学习的隐特征表示也不同,而每一层学习的隐表示结果都有非常重要的表示意义。因此,我们每经过一个隐藏层都进行一次用户与项目隐表示的交互操作,计算如下:

$$X_u = W_{ui}u + W_{ua}u' \quad (12)$$

$$X_v = W_{vi}v + W_{va}v' \quad (13)$$

$$p_{u1} = F_1(W_{u1}X_u) \quad (14)$$

$$q_{v1} = F_1(W_{v1}X_v) \quad (15)$$

.....

$$p_{un} = F_n(W_{un}p_{u(n-1)} + b_{un}) \quad (16)$$

$$q_{vn} = F_n(W_{vn}q_{v(n-1)} + b_{vn}) \quad (17)$$

$$\bar{Y}_n = p_{un} \odot q_{vn} \quad (18)$$

在上述公式中,  $X_u$  和  $X_v$  分别表示用户与项目经过嵌入层之后连接 ID 与辅助信息的隐特征向量;  $W_{ui}$  和  $W_{ua}$  表示用户在嵌入层的 ID 与辅助信息的权值矩阵;  $W_{un}$  指深度网络训练中第  $n$  层的权值矩阵;  $p_{un}$  和  $q_{vn}$  分别表示用户和项目第  $n$  层的隐表示,  $\odot$  表示向量对应元素相乘,  $\bar{Y}_i$  表示第  $i$  层学习到的用户与项目之间的交互结果。

在最后的输出层中,将隐层网络中每一层所学习到的交互结果累加作为用户  $u$  与项目  $v$  的最终交互结果,计算预测评分  $\bar{Y}_{uv}$  如下:

$$\bar{Y}_{uv} = \sum_{x=1}^n \bar{Y}_x + u + b_u + b_v + b_g \quad (19)$$

公式(19)中  $\bar{Y}_x$  表示用户与项目第  $x$  层的交互结果,  $u$  表示可见评分的平均值;  $b_u$ 、 $b_v$ 、 $b_g$  分别表示用户、项目和全局的评分偏置值。

### 3.2 模型学习

基于深度学习的推荐系统中,模型通过学习用户或项目的隐表示,然后利用逐点损失(point-wise loss)和成对损失(pair-wise loss)等类型的损失函数构建目标优化函数对深度学习模型的参数进行优化,最后利用学习到的隐表示进行项目推荐。在本文中,我们只考虑逐点损失,使用均方误差作为目标函数:

$$L_{reg} = \sum_{(u,v) \in Y} (\bar{Y}_{uv} - Y_{uv})^2 \quad (20)$$

公式(20)中  $Y$  表示用户对项目的评分矩阵,  $Y_{uv}$  表示目标值,即用户  $u$  对项目  $v$  的真实评分数据。

针对目标函数的优化我们基于 Tensorflow 框架采用小批次梯度下降算法(Mini-BGD)去训练神经网络模型,分批次迭代地去训练实例  $x$ ,然后往负的梯度方向去更新每个模型参数直到收敛。目前大多数的深度学习框架(例如:Theano,Tensorflow,Keras

等)都提供了优化器函数。

在优化模型时,往往需要防止模型过拟合,提高模型是泛化能力。常见的方法便是正则化,即对模型的目标函数加上正则项。本文中使用 L2 正则化防止过拟合,实际优化的目标函数为:

$$L = \sum_{(u,v) \in Y} (\bar{Y}_{uv} - Y_{uv})^2 + \lambda(\|W_{ui}\|^2 + \|W_{vi}\|^2 + \|W_{ua}\|^2 + \|W_{va}\|^2 + \|W_{un}\|^2 + \|W_{vn}\|^2 + b_{un}^2 + b_{vn}^2 + b_u^2 + b_v^2 + b_g^2) \quad (21)$$

公式(21)中  $\lambda$  表示正则化参数,  $W_{ui}$  与  $W_{ua}$  分别表示映射用户 ID 信息与辅助信息的权重矩阵,  $W_{un}$  表示第  $n$  隐层的权重矩阵,  $b_u$ 、 $b_g$  分别表示用户的评分偏置和全局的评分偏置,其他于项目相关的参数与用户类似。

在训练多层神经网络时,随着输入数据的不断变化以及网络中参数不断调整,网络的各层输入数据的分布会不断变化,各层在训练的过程中需要不断地改变以适应这种新的数据分布,从而造成网络训练困难,难以拟合的问题。为了解决这一问题,Ioffe 和 Szegedy<sup>[19]</sup>等人提出了批标准化(batch normalization,BN),通过对每一层的输入进行归一化来保证每层输入数据分布的稳定性,使得深度学习网络更容易训练。

BN 算法对每一次迭代中的每一层输入都进行了归一化,将每次训练的小批次输入数据  $B=\{x_1, x_2, \dots, x_m\}$  的分布归一化为均值为 0,方差为 1 的高斯分布,如下式:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (22)$$

$$\sigma_B = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (23)$$

$$BN(x_i) = \gamma \cdot \left( \frac{x_i - \mu_B}{\sigma_B} \right) + \beta \quad (24)$$

上述公式中,  $\mu_B$  表示小批次数据的均值,  $\sigma_B$  表示小批次数据的方差,  $\gamma$  与  $\beta$  表示两个可学习的变量,通过这两个可学习的变量去还原上一层需要学习的数据分布,将原来不固定的数据分布全部转换为固定的需要学习到的数据分布。

在 MA-DMF 模型中,我们在嵌入层连接向量之后执行了 BN 操作,另外在每个连续的隐藏层上也应用了 BN 操作,结果表明应用了 BN 操作之后能明显提升模型的性能。

## 4 实验及分析

### 4.1 实验环境及数据集

实验硬件环境为: Intel(R) Xeon(R) CPU E7-4820 @2.00GHz(8 核) CPU、64GB RAM 的刀片

服务器;

1 统、Pycharm2017、Tensorflow 1.3.0 -CPU;

实验数据: Minnesota 大学的 GroupLens<sup>[20]</sup>项目组提供的 MovieLens-Latest-100K 电影评分数据集, 该数据集包括了 671 个用户对 9066 个电影的 100004 条评分记录和 1296 条标签信息。通过数据预处理, 我们从标签和电影类型等辅助信息中抽取了 764 个用户属性特征和 770 个电影属性特征。本实验将数据 80% 作为训练集, 10% 作为验证集, 剩下的 10% 数据作为测试集。

## 4.2 实验评价指标

均方根误差(RMSE)是推荐领域常用的预测准确的评价指标<sup>[6,21]</sup>, 本实验采用 RMSE 对实验的评分预测做评价。RMSE 的计算公式为:

$$RMSE = \sqrt{\frac{\sum_{u,v \in N} (r_{uv} - \bar{r}_{uv})^2}{|N|}} \quad (25)$$

RMSE 的值越小, 则表示预测的结果越准确, 推荐模型的性能越好。

## 4.3 实验结果及分析

### 4.3.1 不同推荐模型 RMSE 实验对比

为了验证本文所提 MA-DMF 模型的推荐性能, 我们基于 Tensorflow 框架与经典的 SVD++<sup>[7]</sup>算法、2015 年 Elkahky<sup>[10]</sup>等人提出的 MV-DNN 模型、2017 年 Xiangnan He<sup>[13]</sup>等人基于 Wide&Deep<sup>[16]</sup>提出的 MLP 模型、2017 年 Zheng L<sup>[22]</sup>等人提出的 Deep CoNN 模型等算法进行了对比实验。

在实验中, 由于采用了批处理操作, 我们可以选择一个更小的正则化参数和更大的学习率。我们设置隐因子个数为 64、L2 正则化参数为 0.00003、学习率为 0.1、隐藏层结构为[128,64,32,16]、嵌入层和深度神经网络层中分别采用均值为 0 方差为  $[0.01/n^2]$  和  $[1/n^2]$  的高斯分布去初始化权重参数( $n$  为嵌入层向量维度和隐藏层神经元个数)、batch-size 为 512, 用小批次梯度下降的方法对损失函数进行优化。通过 RMSE 评价参数进行比较, 本文所提 MA-DMF 模型有所提升, RMSE 最低达到 0.849, 实验结果如图 4 所示:

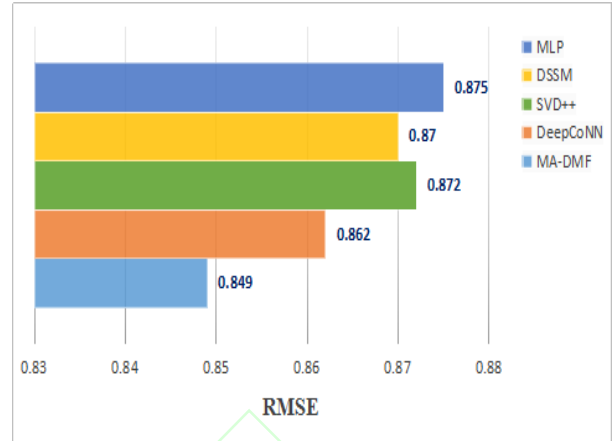


图 4 不同推荐模型 RMSE 实验结果

### 4.3.2 MA-DMF 模型性能分析

基于深度学习的推荐模型中, 通过多层神经网络学习用户与项目的交互时, 深度神经网络结构是一个关键部分。在本文所提的模型中, 每一层网络所学习到的特征表示都被认为是重要的, 我们进一步探索了隐藏层中神经网络层数对模型性能的影响。我们设置 MA-DMF 模型的隐因子个数为 64, L2 正则化参数为 0.00003, 学习率为 0.1, batch-size 为 512 进行对比实验, 结果如图 5 所示。在 4 层网络之前, 每增加隐层数都能一定程度地提升模型的推荐质量, 但隐层数到第 5 层后模型的性能反而下降, 这说明一定数目的隐层能够保证模型好的性能, 但更深层次的网络反而会导致一个更差的推荐效果。

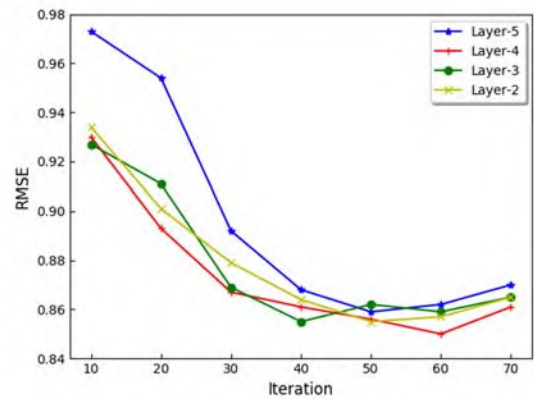


图 5 不同神经网络层数实验结果



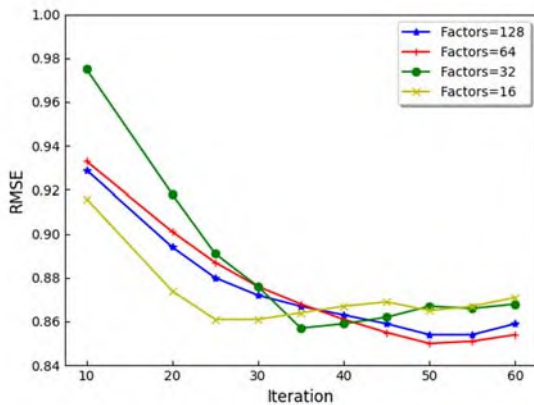


图6 不同隐因子个数实验结果

我们为了选择一个合适的隐因子个数,针对不同的隐因子个数[16,32,64,128]进行了对比实验。我们设置隐层数为4、L2正则化参数为0.00003、学习率为0.1、batch-size为512,实验结果如图6所示。模型在隐因子个数为64时取得较好的推荐效果,说明适当增加隐因子个数能够提升推荐系统的准确度,但超过一定个数,不但对推荐效果没有提高,反而增加了计算和内存的负担。

在实验中,为了验证批处理(Batch Normalization)的有效性,我们设置因子个数为64,隐层结构为[128,64,32,16]、L2正则化参数为0.00003、学习率为0.1、batch-size为512,进行了对比实验,结果如图7所示。通过添加批处理操作能够对深度神经网络层的输入进行归一化,可以保证输入数据分布的稳定性,能一定程度上提升模型的推荐性能。

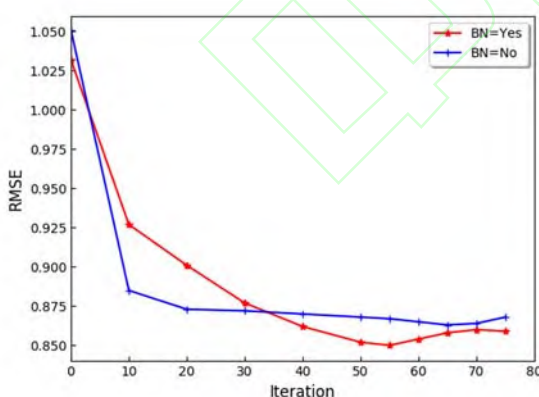


图7 批标准化(BN)作用实验结果

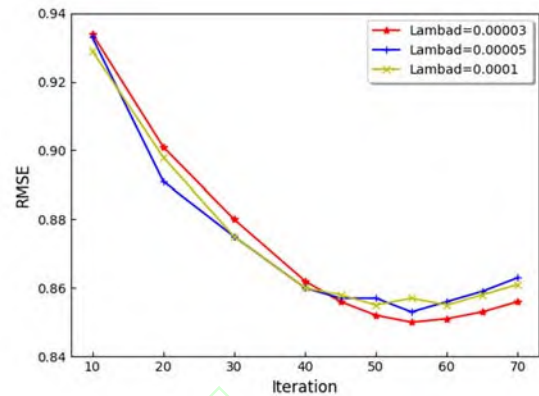


图8 不同正则化参数 $\lambda$ 实验结果

此外,为了研究正则化系数对模型的影响效果,我们也对损失函数设置了不同的L2正则化系数来进行了对比实验。我们设置因子个数为64,隐层结构为[128,64,32,16]、学习率为0.1、batch-size为512,实验结果如图8所示。正则化系数能一定程度上影响模型的性能,在采取BN操作之后,可以选择一个更小的正则化系数。在本次实验中, $\lambda=0.00003$ 时,预测结果与实际评分相差最小。

## 5 结束语

随着大数据时代的不断发展,基于深度学习的推荐系统研究已经成为了当前的研究热点。与传统的推荐系统相比,基于深度学习的推荐模型能够融合更多的异构数据和辅助信息去自动学习用户和项目抽象的隐藏特征,提高推荐的准确性。本文在传统的基于深度学习的推荐模型基础上,融合了辅助信息并对用户与项目的隐表示进行了多次内积交互操作。通过实验,证明了改进的模型能一定程度上提高模型推荐的准确性。但是基于深度学习的推荐模型很难直接对推荐结果给出合理的解释性,而且融合更多的信息并抽取更多的特征之后也增加了计算和内存的负担。在未来的工作中,我们将尝试进行跨域信息的融合,考虑引入注意力机制,探索在循环神经网络(RNN)中进行推荐研究。

## 参考文献:

- [1] Marz N, Warren J. Big Data: Principles and best practices of scalable realtime data systems[M]. Manning Publications Co, 2015.
- [2] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions[J]. IEEE Trans. on Knowledge and Data Engineering, 2005, 17(6):734-749.
- [3] Lu L, Medo M, Yeung C H, et al. Recommendation systems[J], Physics Reports, 2012, 519(1): 1-49.
- [4] Michael J Pazzani and Daniel Billsus. Content-based



- recommendation systems[C]//In The adaptive web, Springer, 2007:325-341.
- [5] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]//In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008: 426-434.
- [6] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems[J]. Computer, IEEE, 2009, 42(8):30-37.
- [7] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S.T. Roweis (eds.), Advances in Neural Information Processing Systems[C]//Curran Associates, Inc, 2008: 1257-1264.
- [8] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th international conference on Machine learning. ACM, 2007: 791-798.
- [9] Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using click through data[C]//Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013: 2333-2338.
- [10] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems[C]//In Proceedings of the 24th International Conference on World Wide Web. ACM, 2015:278-288.
- [11] Hong-Jian Xue, Xin-Yu Dai, Jianbing Zhang, et al. Deep Matrix Factorization Models for Recommendation Systems[C]. ICJAI, 2017.
- [12] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. arXiv preprint arXiv:1606.07792, 2016.
- [13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering[C]. In Proceedings of the 26th International World Wide Web Conference, 2017.
- [14] Liwei Huang, Yanbo Liu. Recommendation system based on Deep Learning[J]. Journal of Computer. 2017, 40(156): 1-28.
- [15] Wang X, He X, Nie L, et al. Item Silk Road: Recommending Items from Information Domains to Social Users[J]. 2017:185-194.
- [16] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. arXiv preprint arXiv:1606.07792, 2016.
- [17] 黄立威, 刘艳博. 基于深度学习推荐系统[J]. 计算机学报. 2017, 40(156):1-28.
- [18] H. Wang, N. Wang, and D.-Y. Yeung. Collaborative deep learning for recommender systems[C]//In KDD, 2015: 1235-1244.
- [19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. In ICML, 2015.
- [20] F. M. Harper and J. A. Konstan. The movielens datasets: History and context[C]. ACM Transactions on Interactive Intelligent Systems, 2015.
- [21] D. Cao, X. He, L. Nie, X. Wei, X. Hu, S. Wu, and T.-S. Chua. Cross-platform app recommendation by jointly modeling ratings and texts[J]. ACM TOIS, 2017, 35(4):37.
- [22] Zheng L, Noroozi V, Yu P S. Joint deep modeling of users and items using reviews for recommendation[C]//Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017: 425-434.