

文章编号: 1007-1423(2023)18-0018-06

DOI: 10.3969/j.issn.1007-1423.2023.18.003

基于深度学习的隐语义协同过滤推荐模型研究

樊艳清¹, 纪佳琪^{2,3*}

(1. 河北民族师范学院信息中心, 承德 067000; 2. 河北民族师范学院数学与计算机科学学院, 承德 067000;
3. 河北省文化旅游大数据技术创新中心, 承德 067000)

摘要: 隐语义模型是一种基于协同过滤的推荐模型, 由于其较好的推荐性能已在各个领域广泛应用。近些年机器学习的一个重要研究分支深度学习也在各个领域大放异彩。为进一步提高推荐系统的推荐性能, 设计了一种结合隐语义模型和深度学习技术的模型。通过对隐语义模型表示形式的等价转换, 可以用深度神经网络表示隐语义模型, 提出了基于深度学习的隐语义协同过滤推荐模型。实验结果表明, 提出的模型在 precision 和 recall 评价指标上明显优于其他协同过滤推荐模型。

关键词: 深度学习; 协同过滤; 隐语义模型; 推荐系统

0 引言

推荐系统已广泛应用于电子商务、新闻、音乐、电影、短视频等领域, 已成为互联网应用必不可少的关键技术之一。以电子商务领域为例, 其与实体店不同, 电子商务门户网站能够售卖数以百万计的产品。用户不可能在有限的时间内筛选所有商品, 推荐系统可以使目标用户快速发现自己想购买的物品, 因此用户越来越依赖于推荐系统给他们推荐喜欢的商品。推荐系统不仅给用户购物带来好的用户体验, 也增加了商家的收入。

在推荐系统刚出现时, 使用了经典的基于信息检索的方法, 如基于内容的过滤^[1]。然而, 这些技术依赖于专家知识, 并且只能应用于某一特定行业, 如应用于图书的推荐系统就很难应用于电影推荐, 因此很难建立一个通用的模型。这导致了它并没有得到广泛的应用, 取而代之并得到广泛应用的其中一种方法是协同过滤推荐模型^[2]。协同过滤推荐模型可以分为几个分支, 最初是采用基于邻域的方法, 它直观、

易于理解和实现, 但缺乏准确性; 第二种方法是基于分类的方法, 这会产生稍好的结果, 但很难解释; 第三种方法, 也是目前较流行的方法, 即基于隐语义模型(latent factor model), 这种方法更为抽象, 需要对数学有很好的理解才能进行解释, 但推荐效果更为准确^[3]。

近些年, 深度学习由于其具有强大的特征提取能力, 已被广泛应用到计算机科学的各个方面: 自然语言处理、语音处理、计算机视觉等^[4], 其在信息检索方面也有一些应用。基于深度学习的成功, 本文提出了一种基于深度学习的隐语义协同过滤推荐模型。浅层的隐语义模型是基于矩阵分解的方法, 本文在基本隐语义模型的基础上将其扩展到更深的版本, 使用深度学习方法, 把矩阵分解扩展为深度矩阵分解版本。

1 相关工作

1.1 基于邻域模型

基于邻域模型最初想法来源于人们在电

收稿日期: 2023-05-08 修稿日期: 2023-08-17

基金项目: 河北省省级科技计划资助项目(20310301D); 承德市科技计划项目(202201A059)

作者简介: 樊艳清(1987—), 女, 黑龙江北安人, 本科, 助理实验师, 研究方向为数据挖掘; *通信作者: 纪佳琪(1984—), 男, 河北大名, 博士, 副教授, 研究方向为大数据机器学习与深度学习、推荐系统, E-mail: 52987541@qq.com

子商务活动中的规律, 它发现用户 u 一定与某些用户具有相同的喜好, 因此可以根据相似用户的喜好给用户 u 做推荐。在这些相似用户中, 会给每个用户分配不同的权重, 相似度越高权重也就越大。这些相似用户对物品 i 的评分再乘以他们对应的权重就是用户 u 对物品 i 的喜好程度。以上描述可以形式化地表示如下:

$$\hat{r}_{ui} = \sum_{v \in \text{neighbours}(u)} w_v r_{vi} \quad (1)$$

其中: u 表示需要推荐物品的用户, \hat{r}_{ui} 表示用户 u 对物品 i 的预测评分(即预测的喜好程度)。我们找到用户 u 的所有近邻 v , 然后用每个近邻的权重 w_v 乘以这个近邻对物品 i 的评分, 最后把所有结果进行加和就可以得到 \hat{r}_{ui} 。这种邻域模型又称为基于用户的协同过滤推荐模型^[5]。

以上是从用户的角度来看待协同过滤, 我们也可以从物品的角度来看待协同过滤, 从而产生了基于物品的协同过滤模型^[6]。基于物品的协同过滤算法不需要寻找相似用户, 而是寻找相似物品, 给用户推荐那些和他们之前喜欢的物品相似的物品。其具体过程是用户 u 对物品 i 的预测评分是用户 u 对所有与物品 i 相似物品的评分乘上这些物品的权重后加和得到的。

1.2 隐语义模型

隐语义模型是一种基于物品过滤推荐模型的推广, 隐语义模型^[7]认为每个用户可以用用户的属性表示, 每个物品也可以用物品的属性表示, 而这些属性是什么事先并不知道, 因此这些属性称为隐属性, 即隐语义。对于具有 m 个用户, k 个隐语义的用户隐矩阵 $P^{m \times k}$ 和具有 n 个物品, k 个隐语义的物品隐矩阵 $Q^{k \times n}$, 用户对物品预测评分矩阵 \hat{R} (即隐语义模型)的计算方式如图 1 所示。

预测评分矩阵				用户隐矩阵			物品隐矩阵			
	i_1	i_2	i_3		k_1	k_2		i_1	i_2	i_3
u_1	5	4.28	4	u_1	1.7	1.79	k_1	1.4	1.42	1.02
u_2	3	1.06	2	u_2	1.28	0.82	k_2	1.47	1.06	1.2

图 1 隐语义模型

图 1 的计算过程可以形式化表示如下:

$$\hat{R}^{m \times n} = P^{m \times k} \times Q^{k \times n} \quad (2)$$

对用户隐语义矩阵 $P^{m \times k}$ 和物品隐语义矩阵

$Q^{k \times n}$ 实际是求解如下的优化问题。

$$\min_{P, Q} (R - \hat{R})^2 + \lambda (\|P\|_2^2 + \|Q\|_2^2) \quad (3)$$

其中: R 是实际评分矩阵, $\|\cdot\|_2$ 是 2 范式的正则化项, 用来防止过拟合问题, λ 是正则化参数。

1.3 基于深度学习的推荐模型

受限玻尔兹曼机^[8]作为深层信念网络的构建块得到了一定发展, 最初也使用它来解决协同过滤问题。然而, 由于其输入需限制为 1 或 0, 因此受限玻尔兹曼机在基于上下文的协同过滤场景中并未得到广泛使用。修正的高斯-伯努利受限玻尔兹曼机也不适用于此类输入, 因为它期望连续值输入在 0 到 1 之间。

近年来, 基于自动编码器的协同过滤技术显示出良好的效果^[9], 它们依赖于一个神经网络, 其输出与输入相同。用于协同过滤时输入(和输出)存在缺失值, 当数据中存在此类缺失条目时, 相应的网络权重不会更新。一旦训练完成, 编码器的表示与解码器相乘, 得到完整的评分矩阵。通过与隐语义模型的相似性分析, 可以把网络权重看作是用户隐语义表示或物品隐语义表示。深度堆叠式自动编码器也可以用于协同过滤, 一般说来它和自动编码器的输入和输出一样, 浅层和深层自动编码器的唯一区别在于后者有更多层的编码器和解码器, 然而实践证明过深的层次对效果的提升帮助不大。

还有一些基于深度学习的协同过滤模型的研究都是启发性的^[10]。如向深度神经网络中的输入一般是用户和物品的 ID 及用户对物品的评分, 输出是用户对物品评分的预测值。此类模型最大的缺陷是用户和物品的 ID 不包含彼此的任何有用信息(如用户或物品的属性特征、用户对物品的评价特征等)。因此, 这些基于深度学习的协同过滤模型对预测评分的准确性有限。

2 基于深度学习的隐语义协同过滤推荐模型

2.1 神经网络表示隐语义

在隐语义模型中, 某一用户可以用 k 维隐向量表示(P 矩阵中的一行), 某一物品也可以用 k 维隐向量表示(Q 矩阵中的一列)。本文中, 我

们可以把这种隐向量的表示方式转换成一种神经网络的表示方式,即把用户隐向量的表示转换成物品隐向量到用户对物品评分的一种神经网络连接,如图2所示。该神经网络的输入是所有用户对物品 j 的评分,用 r_j 表示,输出是物品 j 的隐向量表示,用 q_j 表示。这样就把隐语义模型的最初表示形式转换成了神经网络的表示形式,接下来就可以对网络进行扩展,使其成为一个深度神经网络模型。在深度学习体系结构中,来自上一层的隐语义表示可以充当到下一层的输入。本文就以上述方式提出了基于深度学习的隐语义协同过滤推荐模型。

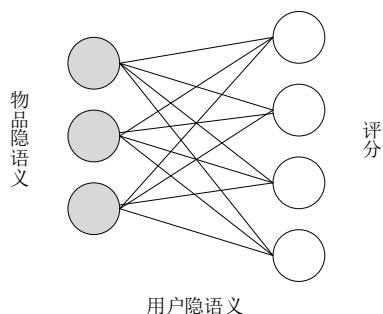


图2 神经网络表示隐向量

2.2 深度隐语义模型

基于深度学习的隐语义协同过滤推荐模型如图3所示,图3中的隐藏层可以扩展到更深层次。理论上讲,随着网络的加深,会提取到更深层次的隐语义表示。比如当我们选择一本书或一部电影时,对于普通的隐语义模型来说只有一小部分隐语义会影响用户的选择(对于一本书来说影响的因素可能是作者、类型、出版社;对于一部电影来说,可能是明星、导演、题材)。然而对于深度隐语义来说,不仅能够捕捉到上述的隐语义,还能够捕捉到其他更深层次的特征。例如,更深层次的神经网络节点可能捕捉到用户的性格特征:内向、外向、中性等,也可能捕捉到物品的一些深层次特征,这些特征的捕获不需要人为干预,深度神经网络会根据不同的数据集自动完成。基于这些更抽象的特征,本文的模型比其他的模型具有更优的鲁棒性和准确性。基于此,我们详细介绍本文提出模型的具体形式化表示。

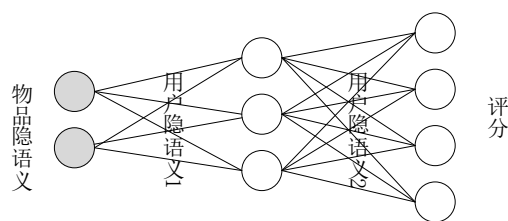


图3 基于深度学习的隐语义协同过滤推荐模型

如1.2节所述,在标准的隐语义模型中,评分矩阵可以看作是用户隐矩阵和物品隐矩阵的乘积。

$$R = PQ \quad (4)$$

本文的深度隐语义模型中,我们将多层的用户隐语义和一个物品隐语义进行结合,假设有2个用户隐语义,则可以表示为

$$R = P_1 P_2 Q \quad (5)$$

其中: P_1 和 P_2 是2层用户隐语义矩阵, Q 是物品隐语义矩阵。因此该模型很容易扩展到 N 层用户隐语义,如公式所示。

$$R = P_1 P_2 \cdots P_N Q \quad (6)$$

由于 R 是评分矩阵,在实际数据集中,矩阵中的很多值都是缺失的,因此我们把缺失值的位置都填充为零,可以形式化表示为公式。

$$Y = I \cdot R \quad (7)$$

其中: \cdot 表示点乘, I 是只包含0或1的矩阵。结合公式(6)就可以形成本文的深度隐语义模型,见公式。

$$Y = I \cdot R = I \cdot (P_1 P_2 \cdots P_N Q) \quad (8)$$

因此,我们需要优化求解如下损失函数:

$$\min_{P_1, P_2, \dots, P_N, Q} \frac{1}{2} \|Y - I \cdot (P_1 P_2 \cdots P_N Q)\|_2^2, \quad P_1 \geq 0, P_2 \geq 0, \dots, P_N \geq 0, Q \geq 0, R = P_1 P_2 \cdots P_N Q \quad (9)$$

本文使用投影梯度法对上述问题进行求解,从而形成了本文的算法框架。

算法1: 基于深度学习的隐语义协同过滤推荐模型优化求解算法框架

Initialize: $R^0, P_1^0, P_2^0, P_3^0, \dots, P_N^0, Q^0$

For $k=1, 2, \dots$

(1) $\bar{R}^k = R^k - \gamma(I^T \cdot (I \cdot R^k - Y))$

(2) 迭代得到 $(R^{k+1}, P_1^{k+1}, P_2^{k+1}, P_3^{k+1}, \dots, P_N^{k+1}, Q^{k+1})$,

求解

$$\min_{X, P_1, P_2, \dots, P_N, Q} \| \bar{R}^k - R \|_2^2, \text{ 其中 } P_1 P_2 \cdots P_N Q = R, X \geq 0, P_1 \geq 0,$$

$$P_2 \geq 0, \dots, P_N \geq 0, Q \geq 0$$

上述算法的关键是如何迭代求解 $R^{k+1}, P_1^{k+1}, P_2^{k+1}, P_3^{k+1}, \dots, P_N^{k+1}, Q^{k+1}$, 由于是在非凸集上进行求解, 所以很难直接得到全局最小值。因此, 本文提出了对约束进行交替投影的方法^[11], 以得出该问题的近似解。每个变量都将 Gauss-Seidel^[12] 的方式依次处理, 然后投影到其关联的约束集上。具体形式化的表示如下:

$$\begin{cases} R^{k+1} = \text{Projection}_+(\bar{R}^k) \\ P_1^{k+1} = \text{Projection}_+\left(\text{Projection}_{\{P_1, P_2^k, P_3^k, \dots, P_N^k, Q^k = R^{k+1}\}}(P_1^k)\right) \\ P_2^{k+1} = \text{Projection}_+\left(\text{Projection}_{\{P_1^{k+1}, P_2, P_3^k, \dots, P_N^k, Q^k = R^{k+1}\}}(P_2^k)\right) \\ P_3^{k+1} = \text{Projection}_+\left(\text{Projection}_{\{P_1^{k+1}, P_2^{k+1}, P_3, \dots, P_N^k, Q^k = R^{k+1}\}}(P_3^k)\right) \\ \dots \\ P_N^{k+1} = \text{Projection}_+\left(\text{Projection}_{\{P_1^{k+1}, P_2^{k+1}, P_3^{k+1}, \dots, P_N, Q^k = R^{k+1}\}}(P_N^k)\right) \\ Q^{k+1} = \text{Projection}_+\left(\text{Projection}_{\{P_1^{k+1}, P_2^{k+1}, P_3^{k+1}, \dots, P_N^{k+1}, Q = R^{k+1}\}}(Q^k)\right) \end{cases} \quad (10)$$

公式中的 Projection_+ 表示在正方向上的投影, 也就是说如果值出现负数我们会把它置为0 (使用 ReLU 函数即可完成)。然后可以应用矩阵 P 在线性约束 R 上的投影 $R = A \cup B$ 的一般性质, 如下:

$$\text{Projection} = P - A^+(A \cup B - R)B^+ \quad (11)$$

公式中的+号表示伪逆矩阵(pseudo-inverse), 因此公式(10)的具体展开计算见公式(12)。

$$\begin{cases} R^{k+1} = \text{Projection}_+(\bar{R}^k) \\ P_1^{k+1} = \text{Projection}_+\left(\begin{matrix} P_1^k - (P_1^k P_2^k P_3^k \dots P_N^k Q^k - R^{k+1}) \cdot \\ (P_2^k P_3^k \dots P_N^k Q^k)^+ \end{matrix}\right) \\ P_2^{k+1} = \text{Projection}_+\left(\begin{matrix} (P_2^k - (P_1^{k+1})^+ \cdot \\ (P_1^k P_2^k P_3^k \dots P_N^k Q^k - R^{k+1}) \cdot \\ (P_3^k \dots P_N^k Q^k)^+ \end{matrix}\right) \\ P_3^{k+1} = \text{Projection}_+\left(\begin{matrix} (P_3^k - (P_1^{k+1} P_2^{k+1})^+ \cdot \\ (P_1^k P_2^k P_3^k \dots P_N^k Q^k - R^{k+1}) \cdot \\ (P_4^k \dots P_N^k Q^k)^+ \end{matrix}\right) \\ \dots \\ P_N^{k+1} = \text{Projection}_+\left(\begin{matrix} (P_N^k - (P_1^{k+1} P_2^{k+1} \dots P_{N-1}^{k+1})^+ \cdot \\ (P_1^k P_2^k P_3^k \dots P_{N-1}^k P_N^k Q^k - R^{k+1}) \cdot \\ (Q^k)^+ \end{matrix}\right) \\ Q^{k+1} = \text{Projection}_+\left(\begin{matrix} (Q^k - (P_1^{k+1} P_2^{k+1} \dots P_N^{k+1})^+ \cdot \\ (P_1^k P_2^k P_3^k \dots P_N^k Q^k - R^{k+1}) \end{matrix}\right) \end{cases} \quad (12)$$

结合算法(1)和公式(12)就完成了模型参数的求解, 在后续的实验中该算法也能较好地收敛。

3 实验

3.1 数据集

本文使用 Movielens 提供的3个电影评分数据集对模型进行评估。

(1) movie-100K: 该数据集包含了943个用户对1682部电影的一万个评分。

(2) movie-1M: 该数据集包含了6040个用户对3900部电影的一百万个评分。

(3) movie-10M: 该数据集包含了71567个用户对10681部电影的一千万个评分。

这些数据集官方在提供时已经划分好了训练集和测试集, 本实验中我们使用5折交叉验证得到实验结果。

3.2 模型收敛性

图4显示模型在 movie-100K 上损失值的变化趋势, 从图4可以看到, 损失值很快下降到最低点, 说明模型的收敛速度较快。我们在其他两个数据集上也得到了相同的实验结果, 由于篇幅关系, 本文不再另行给出。

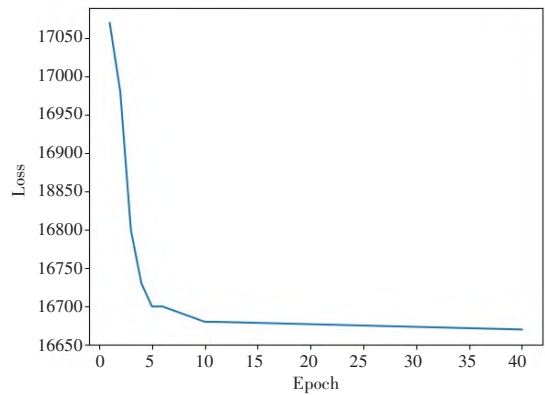


图4 训练损失值变化趋势

3.3 超参数选择

本文提出的模型主要有两个超参数, 一个是神经网络的层数, 另一个是投影梯度法的步长 γ 值。一般说来, 神经网络的层数和数据量有一定的关系。如果数据量过少, 网络层数过多时很容易造成模型过拟合, 通过本文的实验

数据(表1~表3)可以看出,当网络隐藏层数大于3时,效果并没有提升。由于每层的神经元个数设置多少并没有统一的理论支撑,本文也是根据以往的经验采取逐层减半的方法。首先我们只设置1个隐藏层,这个隐藏层的神经元个数分别设置为 $[8, 16, 32, 64, \dots, 128, 256]$,在三个数据集上的实验结果如图5所示,从图5我们可以看出当神经元个数为128时,模型的效果最好。这个结果决定了我们后续增加隐藏层时每层神经元个数的设置。比如,当有2个隐藏层时,我们把隐藏层的神经元个数分别设置为64-32;当有3个隐藏层时,每个隐藏层神经元个数分别设置为64-32-16;当有4个隐藏层时,每个隐藏层神经元个数分别设置为64-32-16-8。实验中发现当隐藏层为4时,效果不仅没有提升反而略有下降。

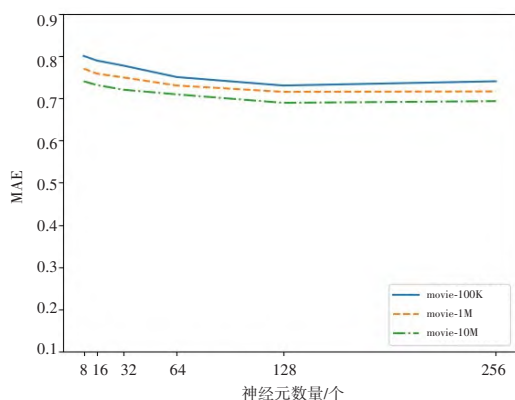


图5 不同隐藏层数在三个数据集上MAE值

接下来我们需要设置投影梯度法的步长 γ 。在实验中,我们发现当 γ 的值在0.01~0.8之间时,对最终的结果影响不大,但是在这个范围外,算法效果会下降。因此我们随机在上述范围内固定了一个 γ 值为0.1。

3.4 结果比较

本文选取了一些基于深度学习的经典推荐模型作为比较对象,分别是(1)collaborative deep learning(CDL)、(2)marginalized deep auto-encoder(MDA)和(3)deep matrix factorization(DMF)。本文选取了精确度(Precision)和召回率(Recall)作为模型的评价指标,其中Precision@N表示推荐数量为N时模型的精确度。在三个数

据集上各模型的实验结果见表1~表3。

表1 在数据集 movie-100K 上的实验结果

模型	Precision		Recall	
	@10	@20	@10	@20
DMF	0.49761	0.3686	0.62177	0.75175
CDL	0.4947	0.36569	0.62274	0.74205
MDA	0.49761	0.36084	0.62177	0.74399
64-32	0.50634	0.3686	0.62953	0.75854
64-32-16	0.51992	0.38703	0.63826	0.76824
64-32-16-8	0.5044	0.36472	0.62274	0.75466

表2 在数据集 movie-1M 上的实验结果

模型	Precision		Recall	
	@10	@20	@10	@20
DMF	0.65087	0.50731	0.60625	0.77503
CDL	0.65087	0.51507	0.6111	0.77794
MDA	0.64893	0.51022	0.60625	0.94187
64-32	0.65281	0.51507	0.61692	0.77503
64-32-16	0.67027	0.52671	0.62177	0.78473
64-32-16-8	0.6499	0.51216	0.61595	0.77212

表3 在数据集 movie-10M 上的实验结果

模型	Precision		Recall	
	@10	@20	@10	@20
DMF	0.65087	0.55193	0.61207	0.7857
CDL	0.65184	0.55096	0.61401	0.7857
MDA	0.6596	0.53835	0.6014	0.77794
64-32	0.66833	0.55193	0.61304	0.78861
64-32-16	0.67512	0.56163	0.6208	0.7954
64-32-16-8	0.67221	0.55484	0.61886	0.79055

从实验结果可以看出,本文提出的模型整体效果要比其他模型效果好,虽然隐语义模型和深度学习模型对于推荐结果的解释性较差,但是实验结果说明隐语义模型结合深度神经网络能够更好地提取特征,从而提升推荐结果的性能。从模型自身对比来看,当本文模型隐藏层数逐步增加到3层时,推荐的效果也逐步增高;但是当隐藏层数达到4层时,推荐效果不增反降,这主要因为层数增加造成学习的参数增加,过多的参数会造成模型过拟合。

4 结语

本文介绍了基于深度学习的隐语义协同过滤推荐模型,通过把浅层隐语义模型和深度学

习技术相结合, 形成了深度隐语义模型。通过使用交替投影的方法对模型进行了优化, 并且模型收敛较快。与其他深度模型对比, 实验结果表明, 本文提出模型的Precision和Recall都高于其他深度模型。近年来张量分解、因子分解在机器学习中越来越重要, 由于张量维度更高, 能表示更多维度的用户或物品特征, 因此在未来的工作中, 我们将研究把多级张量分解和深度学习进行结合来完成推荐。

参考文献:

- [1] CHANDRA A, AHMED A, KUMAR S, et al. Content-based recommender system for similar products in E-commerce[C]//Edge Analytics: Select Proceedings of 26th International Conference (ADCOM 2020). Singapore: Springer Singapore, 2022: 617-628.
- [2] PAN L, SHAO J. Review of improved collaborative filtering recommendation algorithms [C]//Advancements in Mechatronics and Intelligent Robotics: Proceedings of ICMIR 2020. Springer Singapore, 2021: 21-26.
- [3] NOULAPEU NGAFFO A, CHOUKAIR Z. A deep neural network-based collaborative filtering using a matrix factorization with a twofold regularization[J]. Neural Computing and Applications, 2022, 34(9): 6991-7003.
- [4] DONG S, WANG P, ABBAS K. A survey on deep learning and its applications [J]. Computer Science Review, 2021, 40: 100379.
- [5] THAKKAR P, VARMA K, UKANI V, et al. Combining user-based and item-based collaborative filtering using machine learning [C]//Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Springer Singapore, 2019: 173-180.
- [6] SINGH P K, AHMED R, RAJPUT I S, et al. A comparative study on prediction approaches of item-based collaborative filtering in neighborhood-based recommendations [J]. Wireless Personal Communications, 2021, 121: 857-877.
- [7] 范慧婷, 钟春琳, 龚海华. 基于隐语义模型的个性化推荐[J]. 计算机应用与软件, 2017, 34(12): 206-210.
- [8] DECELLE A, FURTLEHNER C. Restricted Boltzmann machine: recent advances and mean-field theory[J]. Chinese Physics B, 2021, 30(4): 040202.
- [9] YU M, QUAN T, PENG Q, et al. A model-based collaborate filtering algorithm based on stacked Auto-Encoder [J]. Neural Computing and Applications, 2022: 1-9.
- [10] LIU B, ZENG Q, LU L, et al. A survey of recommendation systems based on deep learning [C]//Journal of Physics: Conference Series. IOP Publishing, 2021, 1754(1): 012148.
- [11] 练鹏, 戴华. 非负矩阵分解的一种新型交替投影梯度方法[J]. 高等学校计算数学学报, 2017, 39(2): 177-192.
- [12] BAI Z Z, WANG L, WU W T. On convergence rate of the randomized Gauss-Seidel method [J]. Linear Algebra and Its Applications, 2021, 611: 237-252.

Research on latent factor collaborative filtering recommendation model based on deep learning

Fan Yanqing¹, Ji Jiaqi^{2, 3*}

(1. Information Center, Hebei Normal University for Nationalities, Chengde 067000, China;

2. School of Mathematics and Computer science, Hebei Normal University for Nationalities, Chengde 067000, China;

3. The Technology Innovation Center of Cultural Tourism Big Data of Hebei Province, Chengde 067000, China)

Abstract: Latent factor model is a recommendation model based on collaborative filtering, which has been widely used in various fields due to its better recommendation performance. In recent years, deep learning, an important research branch of machine learning, has also made a splash in various fields. In order to further improve the performance of the recommendation system, designed a model that combines the latent factor model and deep learning techniques. By equivalent transformation of the latent factor model representation, the latent factor model can be represented by a deep neural network, and a latent factor collaborative filtering recommendation model based on deep learning is proposed. Experiment results show that the proposed model significantly outperforms other collaborative filtering recommendation models in terms of precision and recall evaluation metrics.

Keywords: deep learning; collaborative filtering; latent factor model; recommender system