

基于偏置度量分解与隐反馈的协同过滤推荐算法

陈一然, 郑 骏

(华东师范大学 计算机科学与软件工程学院, 上海 200333)

摘要: 矩阵分解由于其简单可靠的特性, 是推荐系统中最重要的算法之一, 然而, 由于内积无法完全捕捉用户和商品间的交互, 矩阵分解的性能难以继续提升。为了解决这个问题, 改进了基础的距离度量分解模型, 提出了基于偏置度量分解与隐反馈的协同过滤推荐算法, 并对用户评分时间动态建模, 进一步提升了模型性能。针对推荐系统中最常见的评分预测任务, 分别在三个数据集上进行实验验证, 实验结果表明所提出的模型的预测准确率有明显提升。

关键词: 推荐系统; 矩阵分解; 度量学习; 隐反馈; 协同过滤

中图分类号: TP301.6 **doi:** 10.19734/j.issn.1001-3695.2019.02.0038

Collaborative filtering recommendation algorithm based on biased distance metric factorization and implicit feedback

Chen Yiran, Zheng Jun

(College of Computer Science & Software Engineering, East China Normal University, Shanghai 200333, China)

Abstract: Matrix factorization is one of the most important algorithms in recommendation systems due to its simple and reliable characteristics. However, since the inner product cannot fully capture the interaction between users and items, it is difficult to further enhance the performance of matrix factorization. In order to solve this problem, this paper modified the basic distance metric factorization model, proposes a collaborative filtering recommendation algorithm based on biased metric factorization and implicit feedback, and dynamically models users' rating timestamps, which further improves the model performance. In this paper, the most common rating prediction task in the recommendation system is tested on three data sets. The experimental results show that the prediction accuracy of the proposed model is significantly improved.

Key words: recommendation system; matrix factorization; metric learning; implicit feedback; collaborative filtering

0 引言

近年来, 互联网上的信息量呈爆炸式增长, 普通用户愈发难以从海量数据中获取有效信息, 推荐系统作为一个能够有效缓解信息超载的可靠工具, 如今正被广泛应用于各种领域, 如电子商务, 社交平台等。

协同过滤是推荐系统中最著名的方法论, 它的核心思想是基于用户过去的行为来分析用户的偏好, 而无须创建显式的用户画像。值得注意的是, 协同过滤技术直接依赖于用户行为, 而不需要特定领域的专业知识, 从而避免了大量耗时的数据收集工作^[1]。矩阵分解由于其简单且有效的特性, 成为协同过滤中最流行的方法, 然而, 由于矩阵分解使用的交互建模方式(即内积)过于简单, 其性能仍然受到局限, 因此, 最近的研究提出了多种方法来取代内积, 如度量学习^[2,3]和神经网络^[4-8]

在过去的研究中, 推荐系统的输入通常是用户给予商品的显式反馈, 例如数字评分或星级等, 尽管显式反馈十分可靠, 但事实上, 用户不总是乐意提供明确的反馈, 因而基于显式反馈的推荐算法始终面临数据稀疏的问题。而用户的浏览历史, 点击行为, 购买历史等间接反映用户偏好的隐反馈则更易获取, 且内容丰富。因此, 近几年的研究逐渐转向基于隐反馈的推荐算法, 例如 Koren^[1]提出了同时整合了显式反馈和隐式反馈的模型 SVD++, 进而提升了推荐系统的准确

率。

在现实世界中, 用户的喜好和商品的属性不是一成不变的, 因此需要捕获数据中随时间改变的因素, Koren^[9]在 SVD++ 的基础上提出了对时间动态建模的因子模型 timeSVD++, JianWei^[8]进一步结合神经网络, 提出了基于自编码器的时间感知模型。

本文基于距离度量和矩阵分解, 提出了融合隐反馈的时间感知模型框架, 通过在现实数据集 Movielens 上的对比实验结果可以证明本文提出的模型在评分预测的推荐任务中取得了更高的推荐准确率。

1 相关工作

1.1 协同过滤

协同过滤是推荐系统中应用最广泛、最著名的算法之一, 它的基本假设为: 一个用户的行为, 可以通过和他历史行为相似的用户进行推测。传统的协同过滤算法是基于用户的, 即通过大量用户的历史行为数据, 使用相似性度量算法找出和指定用户最相似的邻居用户, 再根据这些邻居用户的喜好向用户进行推荐。

协同过滤可以视为一个矩阵补全问题, 首先, 将原始输入数据转换为一个大型的交互(即评分)矩阵, 如表 1 所示。其中, $Item_n$ 表示第 n 个商品, $User_m$ 表示第 m 个用户, $Rating_{mn}$ 表示第 m 个用户对第 n 个商品的评分。

收稿日期: 2019-02-20; 修回日期: 2019-04-28

作者简介: 陈一然(1995-), 女, 浙江温州人, 硕士研究生, 主要研究方向为推荐系统(772765729@qq.com); 郑骏(1957-), 男, 上海人, 高级工程师, 主要研究方向为 Web 应用技术、软件工程。

由于商品的总数量远远大于用户评价过的商品, 因此这个交互矩阵会高度稀疏 (常用的现实数据集密度通常在 5% 以下), 如何准确的补全交互矩阵中缺失的评分数据是推荐系统最常见的任务之一。

表 1 用户-商品评分矩阵

Table 1 User-Item rating matrix

	<i>Item</i> ₁	<i>Item</i> ₂	<i>Item</i> _{<i>n</i>}
<i>User</i> ₁	Rating ₁₁	Rating ₁₂	Rating _{1<i>n</i>}
<i>User</i> ₂	Rating ₂₁	Rating ₂₂	Rating _{2<i>n</i>}
.....
<i>User</i> _{<i>m</i>}	Rating _{<i>m</i>1}	Rating _{<i>m</i>2}	Rating _{<i>m</i><i>n</i>}

近年来, 基于矩阵分解的隐因子模型成为了协同过滤中最流行的方法之一, 其原理为将巨大的用户-商品交互矩阵分解为两个低维矩阵的乘积, 进而将用户和商品转移到同一个低维隐因子空间中。最基础的矩阵分解模型将每个商品 *i* 表示为一个商品隐向量 *q_i*, 将每个用户 *u* 表示为一个用户隐向量 *p_u*, 其中 *q_i* 的每一维都表示商品的某一种属性, *p_u* 的每一维都表示用户对商品对应的某一种属性的偏好程度, 而两者的内积就表示用户对商品的所有属性的偏好总和, 因此内积能够用来预测评分值, 即

$$\hat{r}_{ui} = p_u^T \times q_i \quad (1)$$

其中 \hat{r}_{ui} 为预测评分值。

之后可以构造一个损失函数, 并通过最小化该函数的方法来学习隐向量 *p_u* 和 *q_i*, 即:

$$\min_{p, q} \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 \quad (2)$$

其中 *r_{ui}* 为真实评分值。

然而, 用户的评分值通常受到与用户和商品间关联无关的因素影响, 例如, 某些挑剔的用户总是倾向于给出较低分数, 即使该商品的质量已经非常优秀, 又例如电影《泰坦尼克号》, 由于其本身的制作水平优秀, 即使是不太喜欢看爱情片的用户也会给出较高的评分。以上两个例子表明, 每个用户或每个商品的基础分都是不同的, 宽容的用户和质量好的商品的基础分会更高, 而挑剔的用户和质量差的商品的基础分则较低。因此, Koren 等人^[10]提出了 BiasMF, 通过引入用户和商品的偏差项提升了基础的矩阵分解模型的性能。

1.2 基于神经网络的模型

近年来, 深度学习在计算机视觉、自然语言处理、语音识别等多个领域都取得了巨大成功, 由于深度学习在完成复杂任务上的优越性, 其应用范围也越来越广泛。近期的研究也表明, 许多人开始尝试将深度学习应用在推荐系统和信息检索中, 不仅克服了传统模型的许多缺点, 还展现了高质量的推荐能力^[11]。Salakhutdinov 等人^[2]首次尝试将深度学习应用于推荐系统, 提出了基于受限玻尔兹曼机的推荐系统。更近期的研究例如 Dziugaite 等人^[4]提出了基于神经网络的矩阵分解模型 (NNMF), 使用多层感知机取代了矩阵分解中使用的内积, Zhang 等人^[7]提出了基于自编码器的混合协同过滤模型 AutoSVD++。

1.3 基于度量学习的模型

度量 (也可以称为距离或距离函数) 可以定义为任何满足以下四个条件的映射函数: 非负性、对称性、不可分辨的同一性、三角不等式。度量学习是机器学习的一条分支, 其目的在于学习一种能够捕获数据间的重要联系的距离度量, 可用于改进相似性学习算法或降维等应用。

尽管矩阵分解可以通过计算内积获取用户和商品间的相似度, 但由于内积不满足度量性质中关键的三角不等式性质,

因此限制了矩阵分解的能力。为了解决这一问题, Hsieh 等人^[2]将度量学习算法与协同过滤相结合, 提出了协同度量学习 (CML), 使用大间隔近邻算法 (LMNN) 学习用户偏好的同时, 对用户间和商品间的相似度建模。Zhang 等人^[3]提出了分解度量学习 (FML), 使用欧氏距离取代内积, 在评分预测和商品排序两个经典的推荐任务中都取得了最先进的结果。

2 模型描述

2.1 问题定义

假设使用的数据集中包括 *M* 个用户和 *N* 个商品, 以及 *L* 条评分数据, 评分是 [1,5] 的整数。由此可以构造一个评分矩阵 $R \in \mathbb{R}^{M \times N}$, 矩阵中的每个元素 *r_{ui}* 表示用户 *u* 对商品 *i* 的评分, 由于数据集中的评分数量 *L* 远小于矩阵大小 (*M* × *N*), 因此评分矩阵中的大部分元素都是未知的。评分预测任务的目标就是尽可能准确的补全矩阵 *R* 中的缺失值。

2.2 距离分解

矩阵分解将用户和商品映射到低维隐空间中, 并通过两者的内积对用户和商品的交互 (即评分) 建模。而距离分解模型将用户和商品视为同一个低维空间中的点, 通过点之间的距离对其交互建模, 距离越近, 表示用户和商品间的相似度越高, 其预测评分也越高。

本文使用机器学习中常用的欧氏距离 (Euclidean distance, 欧几里德距离) 来计算用户和商品间的距离, 即:

$$d(u, i) = \|u - i\|_2 = \sqrt{\sum_{k=1}^n (u_k - i_k)^2} \quad (3)$$

实践中, 为避免计算平方根, 通常使用欧氏距离的平方作为距离函数, 即

$$d_{ui} = [d(p_u, q_i)]^2 = p_u - q_i^2 \quad (4)$$

由于距离和相似度 (即评分) 是相反的概念, 即距离越近, 相似度反而越高, 无法直接通过距离计算评分, 因此本文使用一个简单的公式将距离转换为评分:

$$\hat{r}_{ui} = r_{max} - d_{ui} \quad (5)$$

其中: *r_{max}* 表示真实评分的最大值, 例如, 假设使用的数据中评分为 [1,5], 预测的距离值为 2.3, 则评分的预测值 \hat{r}_{ui} 为 5-2.3=2.7

2.3 基线估计

2.2 节讨论了用户与商品间的交互关系, 但是用户和商品各自的独立因素影响同样重要。例如, 假设整个数据集中的所有评分的平均值为 3, 某位挑剔的用户 A 的历史评分的均值为 2.5, 比总体均值低了 0.5, 而某件商品 B 由于其本身的质量出众, 其平均评分为 4.5, 比均值高出 1.5, 因此可以计算出用户 A 对商品 B 的预测基础分为 3-0.5+1.5=4。为了去除用户和商品的偏差影响, 本文引入偏差矩阵分解模型 (biasMF)^[10]中提出的基线估计, 即

$$b_{ui} = \mu + b_u + b_i \quad (6)$$

其中: μ 为整体平均评分, *b_u* 为用户 *u* 的偏差项, *b_i* 为商品 *i* 的偏差项。

与矩阵分解中的基线估计不同的是, 此处需要将平均评分 μ 转换为平均距离, 即

$$b_{ui} = (r_{max} - \mu) + b_u + b_i \quad (7)$$

结合上节提出的距离分解, 可以得到模型的基础结构, 即

$$\hat{r}_{ui} = r_{max} - p_u - q_i^2 + b_{ui} \quad (8)$$

2.4 时间动态建模

在上一节中提出的基线估计是时间无关的, 也就是假定

用户的偏好和商品的流行度都是不随时间变化的, 这一点在现实世界中显然不成立。在本文使用的现实数据集 MovieLens 中, 最长的时间间隔长达 14 年, 在这样漫长的时间跨度中, 不仅仅是某个用户, 甚至整个社会的流行趋势都会发生巨大的改变, 因此对时间动态建模是推荐系统中非常关键的一环。

为了更具体的体现时间对评分的影响, 本文在 MovieLens-10M 数据集中进行了实验, 结果如图 1 和 2 所示。图 1 表现了 1995 年到 2009 年间时代的变迁对电影整体平均评分的影响, 可以明显看出在 1999 年(从数据集中的第一个评分出现后的 1700 天左右), 平均评分突然由 3.5 跃至 3.7, 又逐年下降, 在 2005 年左右降至最低点 3.4, 之后又逐年回升。图 2 表现了电影上映之后其评分随时间波动的趋势, 可以明显看出大多数电影刚上映时的评分都较高, 评分数量也更密集, 往后逐渐回落, 评分数量也逐渐减少(为准确起见, 去掉了数据集中较早年的电影)。

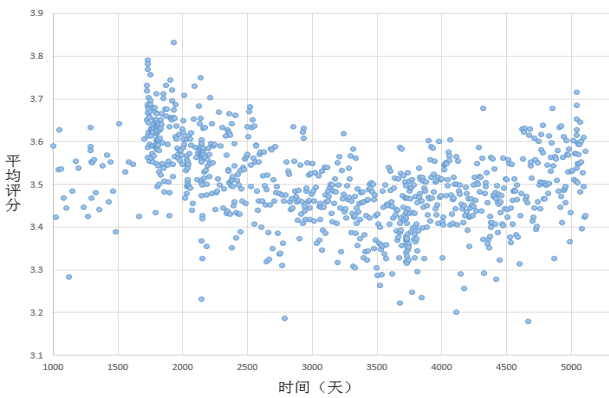


图 1 平均评分随时间变化

Fig. 1 Average movie rating changes over time

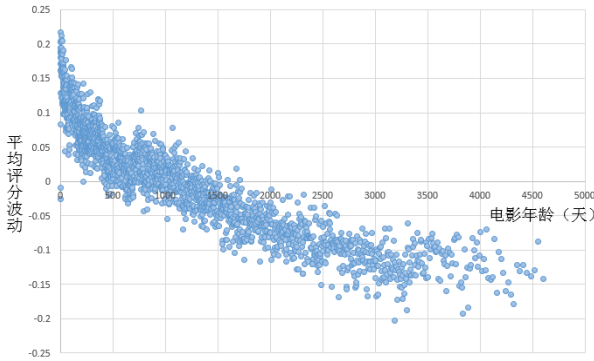


图 2 电影上映后评分随时间在平均分上下波动

Fig. 2 After the movie is released, the rating fluctuates with the average rating over time.

而在上一节提出的基线估计中, 平均评分、用户偏差和商品偏差都被视作一个常数项, 这样是无法捕获时间动态影响的^[9], 因此本文重新修正基线估计, 将其表示为时间的函数, 即

$$b_{ui}(t) = \mu + b_i + b_u(t) + b_t(t) \quad (9)$$

其中向量 b_i 表示时间偏差项, 以捕获时代变迁对整体平均评分的影响。

针对时间对于用户的影响, 其中已知用户偏差是会随时间改变的, 例如一个宽容的用户可能愈发挑剔和严格, 为了捕获这一影响, 令

$$b_u(t) = b_u + d_u^T \times x_{t-t_u} \quad (10)$$

其中 d_u 表示用户的特征向量, x_{t-t_u} 表示时间的特征向量, $t-t_u$

表示用户 u 评分的时刻 t 和他在数据集中第一次评分的时刻 t_u 间的间隔时长。

与用户偏差的改变类似, 商品偏差也是会随时间变化的, 例如商品的热度会逐渐减弱, 因此令

$$b_i(t) = b_i + d_i^T \times z_{t-t_i} \quad (11)$$

其中: d_i 表示商品的特征向量, z_{t-t_i} 表示时间的特征向量, $t-t_i$ 表示商品 i 被评分的时刻 t 和它在数据集中第一次被评分的时刻 t_i 间的间隔时长。

因此最终获得的基线估计为

$$b_{ui}(t) = \mu + b_i + b_u + d_u^T \times x_{t-t_u} + b_i + d_i^T \times z_{t-t_i} \quad (12)$$

2.5 隐反馈

如引言中所述, 隐反馈指的是用户行为中并未明确表达喜好的部分。例如在电影领域中, 评分即为显式反馈, 票房即为隐反馈, 表示用户看过这部电影, 尽管看过并不代表用户最终会喜欢这部电影, 但购票这一行为仍然可以表示用户的某种态度。

然而本文使用的 MovieLens 数据集并没有直接提供可用的隐反馈, 如点击率, 电影票房等, 但可以在获取显式评分的同时, 统计评价过某样商品的用户合集, 而不记录用户具体的评分, 这也可以视为一种隐反馈。也就是说, 不管用户给予商品的评分是高是低, 只要用户给予了商品评分, 他就透露了他的某些偏好, 实践证明, 利用这种隐反馈也可以有效的提高推荐的准确率^[1]。

图 3 通过一个简单的示例展示了使用显式反馈矩阵和隐反馈矩阵的对比, 其中, I_i 表示商品 i , U_j 表示用户 j , “?” 表示对应的用户和商品间并没有交互。

	I_1	I_2	I_3	I_4	...
U_1	4	5	3	?	...
U_2	3	?	?	5	...
U_3	1	?	2	?	...
...

	I_1	I_2	I_3	I_4	
U_1	1	1	1	?	...
U_2	1	?	?	1	...
U_3	1	?	1	?	...
...

(a) 显式反馈的评分矩阵

(b) 隐反馈的评分矩阵

(a) Rating matrices for explicit feedback

(b) Rating matrices for implicit feedback

图 3 显式反馈的评分矩阵(左图)和隐反馈的评分矩阵(右图)

Fig. 3 Rating matrices for explicit feedback(left) and implicit feedback(right)

之前仅仅将商品 i 表示为一个隐特征向量 q_i , 其与真实特征仍存在误差, 现在基于隐反馈数据, 在表示商品 i 时不仅使用商品本身特征, 还使用与它有过交互的用户集合 $N(i)$ 的特征, 将商品 i 的向量表示重新修正为

$$\frac{1}{2} \left(q_i + |N(i)|^{-\frac{1}{2}} \sum_{u \in N(i)} y_u \right) \quad (13)$$

其中: q_i 仍然是从显式反馈(即评分)中学习到的商品向量, 而 $N(i)$ 表示对商品 i 有过交互的用户集合, $\sum_{u \in N(i)} y_u$ 则表示 $N(i)$ 中所有用户向量之和。

在基础模型中加入隐反馈之后, 模型变为

$$\hat{r}_{ui} = r_{max} - p_u - \frac{1}{2} \left(q_i + |N(i)|^{-\frac{1}{2}} \sum_{u \in N(i)} y_u \right)^2 + b_{ui}(t) \quad (14)$$

2.6 正则化

推荐系统的数据过于稀疏的特性通常容易导致过拟合的问题, 因此在模型中加入正则化项以约束参数大小, 防止最终解过于复杂, 本文使用最常见的 L^2 -范数正则化:

$$\text{Regularization} = \lambda_b (b_u^2 + b_i^2 + b_j^2) + \lambda_{u_i} \left(q_i^2 + p_u^2 + \sum_{u \in N(i)} y_u^2 \right) + \lambda_i (x_i^2 + z_i^2 + d_u^2 + d_i^2) \quad (15)$$

其中: λ_b , λ_{u_i} , λ_i 分别为偏差项、用户和商品向量、时间向量的正则化项系数。

2.7 优化算法

通过计算预测的评分值与真实评分值之间的误差, 最小化如下的损失函数:

$$\text{Loss} = \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 + \text{Regularization} \quad (16)$$

本文使用 Adam 优化算法^[12]来学习模型参数, Adam 优化算法是随机梯度下降 (SGD) 的延伸算法。SGD 是神经网络中最常用的优化算法, 它通过迭代更新模型的参数值来最小化目标函数, 然而朴素 SGD 算法也存在收敛速度慢, 易停留于鞍点或局部最小值等问题。Adam 算法通过在训练过程中自适应学习率, 使不同的参数可以使用不同的学习率, 解决了稀疏梯度的问题。同时, 为了加速收敛并减小震荡, Adam 算法引入了动量 (momentum) 概念, 使更新方向不仅由当前梯度决定, 还受之前的下降方向影响。具体算法如下:

$$m_\theta \leftarrow \beta_1 \times m_\theta + (1 - \beta_1) \times g_\theta \quad (17)$$

$$v_\theta \leftarrow \beta_2 \times v_\theta + (1 - \beta_2) \times g_\theta^2 \quad (18)$$

$$\theta \leftarrow \theta - \alpha \times \frac{m_\theta}{\sqrt{v_\theta + \varepsilon}} \quad (19)$$

首先使用式(17)(18)分别计算一阶动量 m_θ 和二阶动量 v_θ , 再使用式(19)更新参数。其中 β_1 与 β_2 为超参数, α 为学习率, ε 为避免零除的平滑项, g_θ 为参数 θ 的梯度, 本模型中所有需要更新的参数 b_u , b_i , b_j , p_u , q_i , d_u , d_i , x_i , z_i , $\forall u \in N(i): y_u$ 均采用 Adam 优化算法更新。

3 实验

3.1 数据集描述

本实验使用了推荐系统领域使用率最高的公共数据集 MovieLens 的三个版本, 分别是 MovieLens 100K, MovieLens 1M, 以及 MovieLens 10M。这些数据集全部来自 GroupLens 研究组 (<http://grouplens.org/datasets/movieLens/>)。详细的数据集统计数据见表 2。

表 2 数据集统计数据
Table 2 Datasets statistics

数据集	商品数	用户数	评分数	密度
MovieLens 100K	1,700	1,000	100,000	5.9%
MovieLens 1M	4,000	6,000	1,000,000	4.2%
MovieLens 10M	10,000	72,000	10,000,000	1.4%

3.2 评价指标

本文采用广泛应用于推荐系统任务的均方根误差 (RMSE) 和平均绝对误差 (MAE) 作为评价指标。这两种指标都是通过计算预测值和真实值之间的误差来反映推荐算法在评分预测上的准确率, RMSE 和 MAE 的数值越小, 表示推荐系统的推荐准确率越高。

RMSE 的定义为

$$RMSE = \sqrt{\frac{1}{N} \sum_{i,j \in T} (R_{ij} - \hat{R}_{ij})^2} \quad (20)$$

MAE 的定义为

$$MAE = \frac{1}{N} \sum_{i,j \in T} |R_{ij} - \hat{R}_{ij}| \quad (21)$$

其中: T 表示测试集, N 表示测试集中的评分总数, R_{ij} 表示

用户 i 对商品 j 的真实评分值, \hat{R}_{ij} 表示推荐系统给出的用户 i 对商品 j 的预测评分值。

3.3 实现细节

本文将每个数据集随机的划分为训练集和测试集, 划分比例为 9:1。实验使用的最佳超参数如下: $\alpha=0.0003$, $\beta_1=0.9$, $\beta_2=0.999$, $\varepsilon=e^{-8}$, $\lambda_b=0.01$, $\lambda_{u_i}=0.02$, $\lambda_i=0.2$, $k_{u_i}=25$, $k_i=20$ 。其中 k_{u_i} 为用户特征向量的维数和商品特征向量的维数, k_i 为时间特征向量的维数。

3.4 对比模型

将本文提出的模型与以下模型进行对比:

- BiasMF^[10]。该模型在基础矩阵分解模型的基础上引入了用户和商品的偏差项。
- SVD++^[11]。该模型在 BiasMF 模型的基础上引入了隐反馈。
- AutoSVD++^[7]。该模型结合了自编码器与协同过滤, 并加入了隐反馈和商品的内容信息。
- NNMF^[13]。该模型使用多层感知机来对用户和商品间的关系建模, 实验中使用了四个隐藏层 (hidden layer, HL)。
- FML^[3]。该模型引入了度量学习对用户和商品间的关系建模。

以上模型的实验均采用其相关文献中设置的参数, 以达到模型的最佳性能。

3.5 模型对比分析

图 4 显示了在数据集 MovieLens 100K 中, 本文所提出的模型与其他所有基线模型的 RMSE 值随迭代次数逐渐下降的曲线, 可以看出本文所提出的模型的 RMSE 值始终低于其他所有模型, 且收敛更稳定, 可看出 Adam 优化算法的优势。图中还可以看出基于神经网络的 NNMF 模型效果并不好, 主要是由于其使用的多层感知机对于复杂关系的抓取能力不如内积。

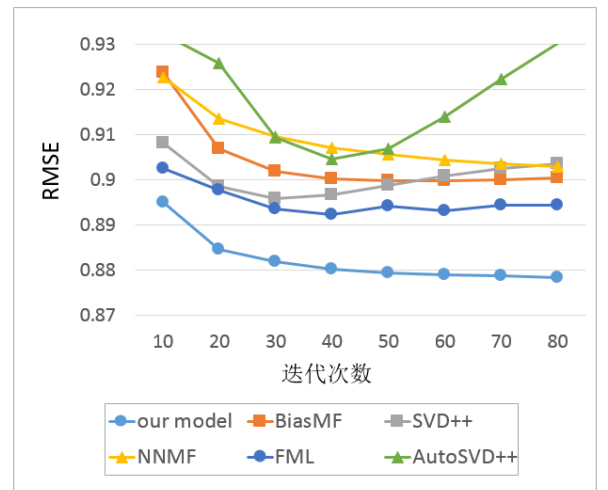


图 4 使用不同迭代次数的实验结果

Fig. 4 Experiment results using different iterations

表 3 显示了所有模型在三个数据集的最佳实验结果, 可以看出在三个数据集中, 本文所提出的模型均优于其他对比模型。

与模型 BiasMF、SVD++、AutoSVD++、NNMF (4HL)、FML 纵向对比, 本文提出的模型在 MovieLens 100K 数据集上的 RMSE 值分别降低了 3.00%, 2.46%, 3.32%, 3.21%, 1.91%。可以看出, 引入时间偏差和隐反馈可以在一定程度上提升推荐系统的效果。

与模型 FML 横向对比, 本文提出的模型在 MovieLens 100K、MovieLens 1M、MovieLens 10M 三个数据集上的 RMSE 值分别降低了 1.91%, 1.57%, 1.03%。可以看出在评分矩阵非常稀疏的情况下, 仅依靠评分和时间数据, 准确率的提升仍然有限, 因此, 如何融合更多的额外信息以提升准确率, 将成为未来工作的重点。

表 3 所有模型的实验结果

Table 3 Experimental results for all models

数据集	MovieLens 100K	MovieLens 1M	MovieLens 10M
方法	RMSE MAE	RMSE MAE	RMSE MAE
BiasMF	0.901 0.709	0.847 0.663	0.803 0.604
SVD++	0.896 0.703	0.839 0.657	0.786 0.597
AutoSVD++	0.904 0.713	0.848 0.665	* *
NNMF(4HL)	0.903 0.709	0.843 0.669	0.800 0.602
FML	0.891 0.696	0.836 0.654	0.779 0.596
本文算法	0.874 0.687	0.823 0.645	0.771 0.591

3.6 超参数分析

为了研究模型的超参数对实验结果的影响, 实验使用不同的超参数设置进行测试以分析。限于篇幅, 本实验仅在数据集 MovieLens-100K 中进行, 对比模型为 BiasMF、SVD++ 和 FML, 评价指标为 RMSE。

用户和商品的特征向量维数 k_u 图 5 显示了在不同的特征向量维数下, RMSE 值的变化曲线。可以看出在同样的向量维数设置下本文所提出的模型均优于对比模型, 且曲线平稳, 说明模型受特征维数的影响较小。基于矩阵分解的 BiasMF 和 SVD++ 模型同样受特征维数影响较小, 但效果一般。基于度量学习的 FML 模型在特征维数逐渐增大的过程中 RMSE 会随之降低, 但仍未超过本文提出的模型。

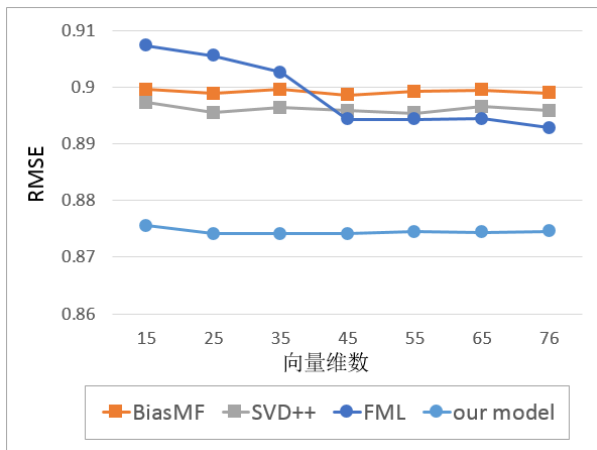


图 5 使用不同特征向量维数的实验结果

Fig. 5 Experiment results using different feature vector dimensions

4 结束语

本文探讨了推荐系统中矩阵分解的缺点和局限, 并介绍了多种改进方法, 如神经网络和度量学习; 之后进一步提出了基于隐反馈和时间感知的距离分解模型。本文在三个真实数据集上进行对比实验, 实验结果表明, 相比基于神经网络的模型与基于矩阵分解的模型, 本文所提出的模型在评分预测任务中取得了明显的性能提升。

在今后的工作中, 研究重点将放在如何将用户和商品的内容信息(如用户的年龄和性别, 商品的类别和描述等)、用户的社交关系网络和上下文信息(如地理位置等)更好地融入推荐模型中, 以进一步提升推荐系统的准确率。

参考文献:

- [1] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]// Proc of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM Press,2008: 426-434.
- [2] Hsieh C K, Yang Longqi, Cui Yin, et al. Collaborative Metric Learning [C]// Proc of the 26th International Conference on World Wide Web. New York:ACM Press,2017: 193-201
- [3] Zhang Shuai, Yao Lina, Tay Yi, et al. Metric Factorization: Recommendation beyond Matrix Factorization [J]. arXiv preprint. 2018: 1802. 04606
- [4] Dziugaite, Karolina G, Roy, et al. Neural network matrix factorization [J]. arXiv preprint. 2015: 1511. 06443
- [5] He Xiangnan, Liao Lizi, Zhang Hanwang, et al. Neural Collaborative Filtering [C]// Proc of the 26th International Conference on World Wide Web. New York:ACM Press,2017: 173-182
- [6] Ruslan S, Andriy M, Hinton, et al. Restricted Boltzmann machines for collaborative filtering [C]// Proc of the 24th International Conference on Machine Learning. 2007: 791-798
- [7] Zhang Shuai, Yao Lina, Xu. Xiwei. AutoSVD+: an efficient hybrid collaborative filtering model via contractive auto-encoders [C]// Proc of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York:ACM Press, 2017.
- [8] Wei Jian, He Jianhua, Chen Kai, et al. Collaborative filtering and deep learning based recommendation system for cold start items [J]. Expert Systems with Applications. 2017: 29-39
- [9] Koren Y. Collaborative filtering with temporal dynamics [C]// Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York:ACM Press, 2009: 447-456
- [10] Koren Y, Robert M. Bell, Volinsky C. Matrix factorization techniques for recommender systems [J]. Computer, 2009, 42 (8): 30-37
- [11] Zhang Shuai, Yao Lina, Sun Aixun. Deep Learning based recommender system: a survey and new perspectives [J]. arXiv preprint. 2017: 1707. 07435
- [12] Diederik P. Kingma, Jimmy Ba. Adam: a method for stochastic optimization [C]// Proc of the 3rd International Conference on Learning Representations. 2015
- [13] Dziugaite G K, Daniel M. Roy. Neural network matrix factorization [J]. arXiv. 2015: 1511. 06443
- [14] Suárez J L, García S, Herrera F. A tutorial on distance metric learning: mathematical foundations, algorithms and software [J]. arXiv. 2018
- [15] Paterek A. Improving Regularized singular value decomposition for collaborative filtering [C]// Proc of KDD Cup and Workshop. 2007.
- [16] 张宏, 王慧. 基于用户评分和共同评分项的协同过滤算法研究 [J]. 计算机应用研究, 2019, 36(1): 77-80.(Zhang Hong, Wang Hui. Research on collaborative filtering algorithm based on user score and common score [J]. Application Research of Computers, 2019, 36(1): 77-80.)
- [17] 毕华玲, 周微, 卢福强. 引入偏置的矩阵分解推荐算法研究 [J]. 计算机应用研究, 2018, 35 (10): 2928-2931, 2964. (Bi Hualing, Zhou Wei, Lu Fuqiang. Bias based matrix factorization recommender techniques. [J]. Application Research of Computers, 2018, 35 (10): 2928-2931, 2964.)