第47卷 第2期 计 算 机 工 程 2021年2月
Vol.47 No.2 Computer Engineering February 2021

·人工智能与模式识别· 文章编号:1000-3428(2021)02-0046-06 文献标志码:A 中图分类号:TP18

Q-learning算法优化的SVDPP推荐算法

周运腾,张雪英,李凤莲,刘书昌,焦江丽,田 豆

(太原理工大学 信息与计算机学院,太原 030600)

摘 要: 为进一步改善个性化推荐系统的推荐效果,通过使用强化学习方法对SVDPP算法进行优化,提出一种新的协同过滤推荐算法。考虑用户评分的时间效应,将推荐问题转化为马尔科夫决策过程。在此基础上,利用Q-learning算法构建融合时间戳信息的用户评分优化模型,同时通过预测评分取整填充和优化边界补全方法预测缺失值,以解决数据稀疏性问题。实验结果显示,该算法的均方根误差较SVDPP算法降低了0.005 6,表明融合时间戳并采用强化学习方法进行推荐性能优化是可行的。

关键词: 协同过滤;奇异值分解;强化学习;马尔科夫决策过程;Q-learning算法

开放科学(资源服务)标志码(OSID):

中文引用格式: 周运腾,张雪英,李凤莲,等.Q-learning算法优化的SVDPP推荐算法[J].计算机工程,2021,47(2):46-51.

英文引用格式: ZHOU Yunteng,ZHANG Xueying,LI Fenglian,et al.SVDPP recommendation algorithm optimized by Q-learning algorithm[J].Computer Engineering,2021,47(2):46-51.

SVDPP Recommendation Algorithm Optimized by Q-learning Algorithm

ZHOU Yunteng,ZHANG Xueying,LI Fenglian,LIU Shuchang,JIAO Jiangli,TIAN Dou

(School of Information and Computer,Taiyuan University of Technology,Taiyuan 030600,China)

【Abstract】 To futher improve the recommendation performance of personalized recommendation systems,this paper proposes a Collaborative Filtering(CF) recommendation algorithm based on SVDPP algorithm optimized by reinforcement learning. Considering the time effect of user ratings,the recommendation problem is transformed into a Markov Decision Process(MDP). On this basis,the Q-learning algorithm is used to construct a user rating optimization model fused with timestamp information. At the same time,in order to solve the data sparse problem,the prediction score is rounded to the nearest integer to fill and optimize the boundary to make up for the missing value in the process of prediction.Experimental results show that the RMSE of this algorithm is 0.005 6 lower than that of SVDPP algorithm,which demonstrates that it is feasible to use the reinforcement learning method and timestamp to optimize the recommendation performance.

【Key words】 Collaborative Filtering(CF);Singular Value Decomposition(SVD);reinforcement learning;Markov Decision Process(MDP);Q-learning algorithm

DOI: 10. 19678/j. issn. 1000-3428. 0056332

0 概述

随着网络和信息技术的不断发展,现实社会中网络信息的数据量呈指数级增长。面对种类繁多的信息,如何获取个性化服务已成为人们的迫切需求。个性化推荐[1]通过各种推荐算法分析用户的行为喜好,能够有效过滤用户不需要的信息,主动为用户提供个性化的产品或服务。目前,个性化推荐已被广泛应用于社交[2]、新闻、音乐、图书和电影网站等应用[3],如网易云音乐[4]、淘宝商品推荐[5]、Netflix和MovieLens电影推荐等。

协同过滤(Collaborative Filtering,CF)技术[6]可用于推荐算法,其主要包括基于内存和基于模型两类算法。其中:基于内存的协同过滤推荐算法通过分析“用户-项目”评分矩阵计算相似度,并根据相似度进行预测推荐;基于模型的协同过滤推荐算法通过用户的历史购买记录、网络操作等数据训练一个预测模型,进而利用此模型对项目进行预测评分。许多研究通过改进协同过滤算法优化了推荐效果,如限制性玻尔兹曼机、K近邻算法[7]、奇异值分解[8](Singular Value Decomposition,SVD)算法及其改进模型(Singular Value Decomposition

基金项目: 山西省重点研发计划(社会发展领域)(201803D31045);山西省自然科学基金(201801D121138);山西省科技重大专项(20181102008)。

作者简介: 周运腾(1995—),男,硕士研究生,主研方向为推荐系统、强化学习;张雪英(通信作者)、李凤莲,教授、博士研究生;刘书昌,硕士研究生;焦江丽,博士;田 豆,硕士研究生。

收稿日期: 2019-10-18 **修回日期:** 2020-01-20 **E-mail:** tyzhangxy@163.com

Plus Plus，SVDPP）。SVD不仅是一个数学问题，其在很多工程中也得到了成功应用。在推荐系统方面，利用SVD可以很容易地得到任意矩阵的满秩分解，进而实现对数据的压缩降维。SVDPP在SVD基础上进一步融入了隐式反馈信息，采用隐式偏好对SVD模型进行优化，因此性能更优。但是SVDPP与SVD都没有考虑时间戳对推荐性能的影响，而实际推荐效果与时间戳仍然有一定的关联性，如十年前的用户对某一部电影的评分与当前用户的评分是有一定差异的，因此有必要对其进行改进，优化预测效果。

本文考虑时间戳对推荐性能的影响，通过马尔科夫决策过程（Markov Decision Process，MDP）对用户、评分、电影和时间进行建模，并利用强化学习Q-learning算法优化推荐算法，从而提升推荐效果，实现更准确的预测。

1　问题描述

在使用基于模型的推荐算法进行预测时，SVD和SVDPP模型都没有考虑时间戳对于推荐准确性的影响，而用户之前看过的电影会对他之后选择观看的电影类型及其对电影的评分产生影响。因此，本文利用马尔科夫决策过程对这种时序决策问题建模，反映时间戳数据与评分的关系，并通过强化学习对推荐算法进行优化。

Q-learning算法是一种基于反馈和智能体的无模型的强化学习方法，本文提出一种基于Q-learning算法优化的SVDPP推荐算法RL-SVDPP，以解决SVDPP在电影推荐预测中未考虑时间戳影响的问题。

强化学习[9-10]作为一种机器学习方法，主要原理是智能体以试错的方式进行学习，通过自身与环境交互获得奖励。目前，强化学习已经被成功应用到神经网络和文本处理等领域，但将该方法直接应用于推荐算法的研究较少，现有研究主要通过结合深度神经网络进行模型训练并推荐预测[11]。笔者受启发于Netflix Prize比赛中竞赛选手将时间戳应用到矩阵分解模型[12]，以及文献[13]将强化学习用于协同过滤的思路，考虑到用户对一部未看过电影的评分可以通过他之前看过的电影评分来预测，即时间戳会影响用户对未知电影的评分，将SVDPP推荐算法得到的预测评分进一步采用马尔科夫决策过程中的奖惩函数进行优化，建立推荐预测评分与马尔科夫决策过程之间的映射关系，并利用强化学习Q-learning算法[14]进行模型训练，以优化预测过程。

2　建模过程

2.1　奇异值分解

现实生活中的“用户-项目”矩阵规模很大，但是由于用户的兴趣和消费能力有限，单个用户消费产生评分的物品是少量的，SVD的核心思想是将高维稀疏的矩阵分解为2个低维矩阵，相对于特征值分解只能用于对称矩阵，SVD能对任意$M\times N$矩阵进行满秩分解，以实现数据压缩。但是在采用SVD对矩阵进行分解之前，需要对矩阵中的空白项进行填充，以得到一个稠密矩阵。假设填充前的矩阵为$\boldsymbol{R}$，填充后为$\boldsymbol{R}'$，则计算公式为：

$$\boldsymbol{R}'=\boldsymbol{P}\boldsymbol{R}\boldsymbol{Q}^{\mathrm{T}} \tag{1}$$

利用SVD算法获取预测评分的计算公式如下：

$$\hat{r}_{ui}=\mu+b_u+b_i+\boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{p}_u \tag{2}$$

其中：μ代表评分的平均值；b_u、b_i分别代表用户u和电影i的偏置量；$\boldsymbol{q}_i$、$\boldsymbol{p}_u$分别对应电影和用户在各个隐藏特质上的特征向量，上标T代表转置。

如果用户对某个电影进行了评分，则说明他看过这部电影，这样的行为蕴含了一定的信息，从而可以推理出评分这种行为从侧面反映了用户的喜好，据此可将这种喜好通过隐式参数的形式体现在模型中，得到一个更为精准的模型SVDPP[15]。

利用SVDPP模型获取预测评分的计算公式如下：

$$\hat{r}_{ui}=\mu+b_u+b_i+\boldsymbol{q}_i^{\mathrm{T}}\left(\boldsymbol{p}_u+\frac{1}{\sqrt{|\boldsymbol{N}(u)|}}\sum_{j\in N(u)}y_j\right) \tag{3}$$

其中：$\boldsymbol{N}(u)$为用户u浏览和评价过的所有电影的集合；y_j为隐藏的评价了电影j的个人喜好偏置；用户u的偏好程度由显式反馈$\boldsymbol{p}_u$和隐式反馈$\frac{1}{\sqrt{|\boldsymbol{N}(u)|}}\sum_{j\in N(u)}y_j$两部分组成。

2.2　马尔科夫决策过程

马尔科夫决策过程是决策理论规划、强化学习及随机域中其他学习问题的一种直观和基本的构造模型[16]。在这个模型中，环境通过一组状态和动作进行建模，可用于执行控制系统的状态。通过这种方式来控制系统的目的是最大化一个模型的性能标准。目前，很多问题（如多智能体问题[17]、机器人学习控制[18]和玩游戏的问题[19-20]）成功通过马尔科夫决策过程进行建模，因此，马尔科夫决策过程已成为解决时序决策问题的标准方法[21]。

一般的马尔科夫决策过程由五元组$<\boldsymbol{S},\boldsymbol{A},P,\gamma,R_{\mathrm{ew}}>$表示，如图1所示，其中，$s_t$表示状态，$a_t$表示动作，$r_t$表示回报函数。智能体感知当前环境中的状态信息，根据当前状态选择执行某些动作，环境根据选择的动作给智能体反馈一个奖惩信号，根据这个奖惩信号，智能体就从一个状态转移到了下一个状态。

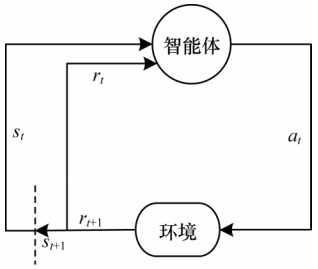

图1　马尔科夫决策过程

Fig.1　Markov decision process

采用强化学习方法对SVDPP推荐模型进行优化，首先需要建立推荐预测模型与马尔科夫决策过程的映

射关系。由于本文采用MovieLens 1M数据集作为研究对象,因此需要将用户在不同时间戳下对电影的评分转换成五元组以构造马尔科夫决策过程。下面给出本文设计的电影评分到马尔科夫决策过程的映射关系。

1)状态空间$\boldsymbol{S}$。本文将用户u在时间t下对电影的评分记为状态$s_t^{(u)}$,因为数据集中用户对电影的评分是[1,5]区间内的5个整数,所以$s_t^{(u)}$的范围为[1,5],所有时间戳下的状态$s_t^{(u)}$构成了状态空间$\boldsymbol{S}$。

2)动作空间$\boldsymbol{A}$。考虑到用户u在时间t下看了电影并给出了评分$s_t^{(u)}$,该评分会影响其(t+1)时间对电影的评分$s_{t+1}^{(u)}$,所以将$a_t^{(u)}$记为从$s_t^{(u)}$到$s_{t+1}^{(u)}$的动作,如式(4)所示,所有时刻的动作$a_t^{(u)}$构成了动作空间$\boldsymbol{A}$。

$$s_t^{(u)} \xrightarrow{a_t^{(u)}} s_{t+1}^{(u)} \tag{4}$$

3)状态转移概率P。用户u在状态$s_t^{(u)}$下采取的动作$a_t^{(u)}$是由时间戳决定的,动作$a_t^{(u)}$一旦确定,则下一个状态$s_{t+1}^{(u)}$也同时确定,由此认为状态之间的转移概率也是确定的,即$a_t^{(u)}=s_{t+1}^{(u)}$,P=1,动作$a_t^{(u)}$的取值范围为[1,5]。

4)折扣因子γ。在模型中,每次动作会产生对应的奖励,但是同一用户观看电影的时间远近对选择下一步将观看电影的影响也会不同,γ就是反映该影响的一个因子。越是后期的奖励折扣越大,同时得到的回报总是有限的,因此,设置$0\leqslant\gamma<1$。

5)奖惩函数R_{ew}。奖惩函数值表征了一个状态中完成某个动作所获得的奖励,本文定义奖惩函数值R_{ew}如下:

$$R_{\mathrm{ew}}(s_t^{(u)}, a_t^{(u)}) = s_{t+2}^{(u)} - \hat{r}_{ui} \tag{5}$$

其中:$s_{t+2}^{(u)}$为(t+2)时用户u对电影的评分;$\hat{r}_{ui}$表示用SVD或SVDPP模型计算出的用户u对电影i的预测评分;R_{ew}表示用户u在状态$s_t^{(u)}$下采取动作$a_t^{(u)}$所获得的奖惩值,根据奖惩函数可得到对应的奖惩表。

2.3 状态表生成

由上述马尔科夫决策过程可知,一个状态转移到下一个状态的动作对应下一个时间电影的评分,虽然这样在表面上忽略了电影名及电影类型,但用户对电影的喜好被隐式地反映在时间戳里,通过这个过程可将MovieLens 1M数据集处理为表1所示的形式。其中,括号中的第1个数字反映了对应行用户给对应列电影的评分,第2个数字反映了对应行用户观看对应列电影的时间戳信息或者时间顺序,如第1行第1列(4,3th)表示用户1观看电影1的时间顺序是第3个,因此,时间戳t=3且用户1对电影1打了4分,NaN表示对应用户未观看这部电影。

表1 MovieLens 1M数据集部分数据

Table 1 Partial data of MovieLens 1M data set

用户	电影1	电影2	电影3	电影4	电影5
用户1	(4,3th)	(3,2nd)	(5,1st)	NaN	(3,4th)
用户2	(4,1st)	NaN	NaN	NaN	(5,2nd)
用户3	NaN	(1,1st)	(4,4th)	(4,3th)	(3,2nd)
用户4	(5,2nd)	(3,1st)	NaN	(2,3th)	NaN
用户5	(4,1st)	NaN	(2,4th)	(1,2nd)	(4,3th)

将表1的数据按照时间戳排序,生成的状态转移路径如下:

5→3→4→3
4→5
1→3→4→4
3→5→2
4→1→4→2

根据表1得到该状态转移路径的规则,以第1行为例进行说明。第1行状态转移路径5→3→4→3反映了用户1在时间戳t=1时看电影3,对电影3的评分为5,t=2时看电影2,对电影2的评分为3,t=3时看电影1,对电影1的评分为4,t=4时看电影5,对电影5的评分为3。其余4个转移路径采用类似方式得到。

此状态转移路径表示马尔科夫决策过程中状态的转移,指引了Q表更新的方向。

3 RL-SVDPP算法

本文提出的RL-SVDPP算法包括训练与预测两部分。训练时,首先采用SVDPP算法对训练集进行模型训练,得到SVDPP推荐模型,如式(3)所示,然后对强化学习模型进行训练,利用式(5)所示的奖惩函数计算状态转移的奖惩值R_{ew},完成强化学习Q表的更新,用于SVDPP推荐评分的优化模型。预测时,首先根据SVDPP推荐模型得到预测评分值,再用本文设计的优化模型对预测评分进行优化,得到最终的预测评分。本文设计的优化模型表示如下:

$$\hat{r}'_{ui} = \hat{r}_{ui} + Q(s_{t-2}^{(u)}, a_{t-2}^{(u)}) \tag{6}$$

其中:$\hat{r}_{ui}$为利用SVDPP推荐模型计算得到的用户u对第i个电影的预测评分;$s_{t-2}^{(u)}$为用户u在看电影i之前时间戳为(t-2)时看电影的评分,$a_{t-2}^{(u)}$为时间戳(t-1)时看电影的评分,$Q(s_{t-2}^{(u)}, a_{t-2}^{(u)})$为$(s_{t-2}^{(u)}, a_{t-2}^{(u)})$坐标下Q表的值,需要采用强化学习算法并基于SVDPP推荐模型的预测评分得到,用于实现对最终预测评分的优化,若$(s_{t-2}^{(u)}, a_{t-2}^{(u)})$的值不存在,则取当前Q表均值;$\hat{r}'_{ui}$为优化后的预测评分。

3.1 训练过程

首先通过式(3)对训练集进行训练,得到SVDPP推荐模型;然后对强化学习模型进行训练,利用式(5)计算奖惩值R_{ew},进而将R_{ew}用于Q-learning算法中Q值的更新过程。Q表更新公式如下:

$$Q(s_{t+1}^{(u)}, a_t^{(u)}) = Q(s_t^{(u)}, a_t^{(u)}) + \alpha\left[R_{\mathrm{ew}}(s_t^{(u)}, a_t^{(u)}) + \gamma \max_{a_t^{(u)\prime}} Q(\delta(s_t^{(u)}, a_t^{(u)}), a_t^{(u)\prime}) - Q(s_t^{(u)}, a_t^{(u)})\right] \tag{7}$$

其中:$Q(s_t^{(u)}, a_t^{(u)})$为一个5×5的Q表;$Q(s_t^{(u)}, a_t^{(u)})$的初始值为0;$Q(s_t^{(u)}, a_t^{(u)})$为Q表坐标$(s_t^{(u)}, a_t^{(u)})$处的Q值;$R_{\mathrm{ew}}(s_t^{(u)}, a_t^{(u)}) + \gamma \max_{a_t^{(u)\prime}} Q(\delta(s_t^{(u)}, a_t^{(u)}), a_t^{(u)\prime}) - Q(s_t^{(u)}, a_t^{(u)})$是选择下一步动作的奖惩值;$\alpha$为学习率,$\gamma$为折扣因子。$Q$值越大,说明执行下一步动作得到的奖励越多,反之奖励越少。

RL-SVDPP算法训练过程的伪代码如下：

算法　RL-SVDPP算法训练过程

输入 用户数量N，用户已评分的电影M，学习率α，折扣因子γ

输出 预测回报$Q(s,a)$

初始化Q(s,a)=0，对任意s∈S，a∈A：

训练SVDPP模型，由式(3)计算$\hat{r}_{ui}$；

从数据中获取初始状态s和动作a；

for each episode i=1：N do

for k=1：M_i do

根据式(5)计算奖惩函数；

将计算出的奖惩函数用于式(7)，更新Q表；

end for

end for

3.2　预测过程

预测过程根据SVDPP推荐模型得到的预测评分，结合训练的Q表来预测用户u对电影i的评分，同时可预测用户u未观看但是其他用户观看过的电影。

此外，在式(6)所示的预测评分优化模型中不用$s_{t-1}^{(u)}$的原因在于：如果用$s_{t-1}^{(u)}$，则优化模型会变为$\hat{r}'_{ui}=\hat{r}_{ui}+Q(s_{t-1}^{(u)},a_{t-1}^{(u)})$，因为$a_{t-1}^{(u)}=s_t^{(u)}$，而$s_t^{(u)}$正是需要预测的评分$\hat{r}'_{ui}$，所以选用$s_{t-2}^{(u)}$。

3.3　数据稀疏性及边界点问题的处理

本文所采用的MovieLens 1M数据集存在缺省值，即存在没有评分的电影信息。根据本文优化模型的构建思路，后续优化过程中需要利用未评分电影评分信息，这将导致$Q(s_{t-2}^{(u)},a_{t-2}^{(u)})$中可能缺少$s$或者$a$的值，从而使优化模型失效。为避免出现这一情况，本文采用SVDPP模型对缺失值进行预测再取整填充，以解决数据稀疏的问题。

此外，当t=1，2时，边界的$s_{t-2}^{(u)}$和$a_{t-2}^{(u)}$会超出下标范围，出现没有对应取值的情况。因此，本文采用最后两列的预测评分作为第－1列和第0列的预测评分数据，以保证数据的完整性。

4　实验

4.1　实验数据

本文实验采用MovieLens 1M数据集，其中包含6 040个用户对3 952个影片的近1亿条评分，评分范围为1分~5分。本文将数据的80%作为训练集来训练RL-SVDPP模型，其他的20%作为测试集，通过均方根误差(Root-Mean-Square Error，RMSE)来评价推荐算法的准确性。

4.2　评价指标

评分预测的准确度一般通过均方根误差来决定，定义如下：

$$\mathrm{RMSE}=\frac{\sqrt{\sum_{u,i\in T}\left(r_{ui}-\hat{r}'_{ui}\right)^2}}{N} \tag{8}$$

其中：r_{ui}表示测试集中用户u对电影i的真实评分；$\hat{r}'_{ui}$为采用本文算法得到的预测评分；T为电影集合；N表示该用户看过的电影总数。

4.3　实验结果与分析

为验证本文算法的有效性，除了对SVDPP模型进行优化得到RL-SVDPP模型外，同时也对SVD模型进行训练，建立优化模型RL-SVD。实验分别建立SVD及SVDPP模型，并求出预测评分，以得到奖惩函数R_{ew}，根据奖惩函数可得到对应的奖惩表，如表2和表3所示。奖惩函数作为马尔科夫决策过程中最重要的部分，能够隐式地反映学习目标，指出马尔科夫决策过程的前进方向。在表2和表3中，行表示状态，列表示动作，如－1.137 11表示在状态1时，进行动作1得到的奖励值为－1.137 11，其他以此类推，其中奖励值为正表明对正确行为给予奖励，奖励值为负表明对错误动作给予惩罚。

表2　由SVD预测评分得到的奖惩表

Table 2　Reward and punishment table by SVD prediction scores

s	a=1	a=2	a=3	a=4	a=5
1	－1.137 110	.0.995 822	－1.818 880	0.027 404	2.268 370
2	－1.995 260	0.485 455	1.212 400	0.825 068	1.433 140
3	－2.524 050	0.292 321	1.195 120	－1.704 800	0.326 522
4	1.407 930	0.732 077	0.948 159	1.693 940	－2.686 380
5	－0.329 110	－1.000 000	1.652 400	1.181 120	1.532 800

表3　由SVDPP预测评分得到的奖惩表

Table 3　Reward and punishment table based by SVDPP prediction scores

s	a=1	a=2	a=3	a=4	a=5
1	1.000 000	2.000 000	2.000 000	3.000 000	3.000 000
2	1.583 054	0.893 456	1.639 407	0.605 505	0.581 861
3	－1.362 890	2.382 008	1.052 259	－0.446 430	0.869 946
4	－0.584 160	－1.084 340	－1.865 920	－0.873 420	0.415 578
5	－1.000 000	－2.000 000	－2.000 000	－1.387 400	－0.394 720

将奖惩函数R_{ew}用于Q表更新过程，更新后的Q表如表4和表5所示。可以看出，通过Q-learning算法训练生成的Q表中的值有正有负。为更形象地进行描述，将表中数据绘制成三维空间图，如图2和图3所示，其中，凸起和凹陷部分表示在某状态下采取动作获得的期望收益有好有坏。可以看出：RL-SVD算法Q表三维图中Q值动态变化范围较大，变化范围为－0.979 930~1.000 000，25个Q值中有14个为负值；RL-SVDPP算法得到的Q表三维图中Q值动态变化范围较小，变化范围

为 – 0.145 190~0.175 280,25个Q值中有10个为负值。这表明RL-SVDPP选择正确动作得到奖励的情况多于选择错误动作进行惩罚的情况,因此,其优化性能优于RL-SVD。下文将通过RMSE性能对比进一步验证该结论。

表4 由SVD预测评分得到的Q表

Table 4 Q table by SVD prediction scores

s	a=1	a=2	a=3	a=4	a=5
1	– 0.214 670	– 0.122 690	– 0.979 930	– 0.014 350	0.017 254
2	– 0.129 890	– 0.239 010	– 0.190 750	– 0.037 290	0.038 101
3	– 0.108 970	– 0.235 290	– 0.254 930	0.141 152	0.201 773
4	– 0.037 480	– 0.057 550	0.092 474	0.692 833	0.631 899
5	– 0.008 240	0.011 406	0.172 137	0.592 628	1.000 000

表5 由SVDPP预测评分得到的Q表

Table 5 Q table by SVDPP prediction scores

s	a=1	a=2	a=3	a=4	a=5
1	0.001 378	0.007 158	0.019 195	0.009 397	0.000 638
2	0.004 215	0.048 743	0.175 280	0.103 664	0.004 104
3	– 0.001 150	0.007 451	0.123 456	0.216 657	0.015 837
4	– 0.002 020	– 0.030 190	– 0.145 190	– 0.128 320	0.000 657
5	– 0.000 560	– 0.005 520	– 0.038 520	– 0.083 030	– 0.018 690

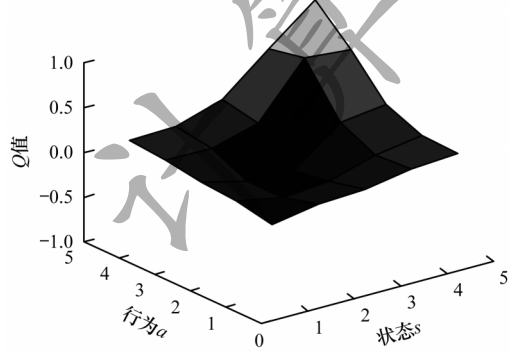

图2 RL-SVD算法Q表三维图

Fig.2 3D diagram of Q table for RL-SVD algorithm

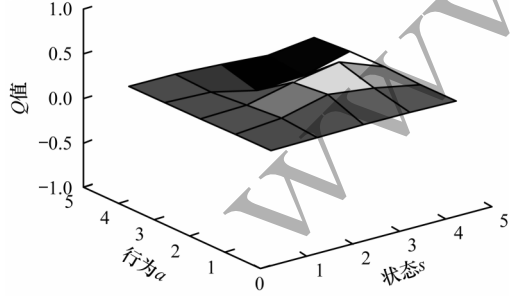

图3 RL-SVDPP算法Q表三维图

Fig.3 3D diagram of Q table for RL-SVDPP algorithm

对20%的测试集采用本文提出的优化模型RL-SVD和RL-SVDPP计算预测评分,并通过式(8)求解其与实际评分的均方根误差,验证本文优化方法的有效性。RMSE比较结果如表6所示。

表6 本文算法与已有SVD/SVDPP的RMSE对比

Table 6 Comparison of RMSE by the proposed algorithm and the existing SVD/SVDPP

算法	γ	α	RMSE	RMSE降低值
SVD	0.5	0.000 006	0.877 279 492	0.004 3
RL-SVD			0.873 017 775	
SVDPP	0.5	0.000 006	0.827 929 099	0.005 6
RL-SVDPP			0.822 270 711	

可以看出:相对SVD算法,采用RL-SVD算法得到的预测结果比优化前SVD算法预测结果的RMSE降低了0.004 3;相对SVDPP算法,采用本文提出的RL-SVDPP算法得到的预测结果比优化前SVDPP的RMSE降低了0.005 6,验证了本文融合时间戳信息建立的强化学习优化的推荐模型的有效性,也说明用户对电影的评分与时间戳确实有一定的关系。

由于学习率α和折扣因子γ是可以动态调整的,因此进一步研究RL-SVDPP算法中α和γ的变化对预测性能的影响,实验结果如图4和图5所示。由图4可知,当γ一定时,α从0.000 003增大到0.3,10倍递增,RMSE的值会增大,并且当α比较大时,RMSE变化很小,因此,α应尽可能取较小的值。由图5可知,当α一定时,γ从0.4增大到0.6,RL-SVDPP算法的RMSE不断减小,实验中最好的效果是当α=0.000 003和γ=0.6时,此时RMSE能达到0.819 48,相比之前降低了0.008 6,由此证明了RL-SVDPP算法的可行性。

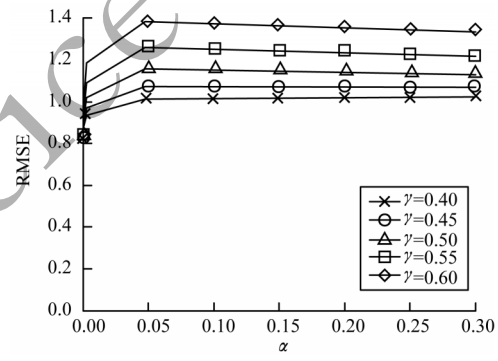

图4 γ一定时α变化对RMSE的影响

Fig.4 Effect of α change on RMSE with constant γ

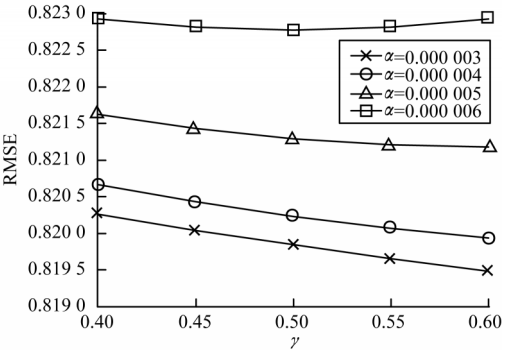

图5 α一定时γ变化对RMSE的影响

Fig.5 Effect of γ change on RMSE with constant α

5　结束语

本文提出一种强化学习Q-learning算法优化的SVDPP推荐算法RL-SVDPP。将用户在不同时间戳下对电影的评分动作转化为马尔科夫决策过程，结合协同过滤算法与强化学习奖惩过程进行建模，对SVDPP推荐预测评分进行优化，并通过调整影响因子来改善预测效果。实验结果表明，用户过去的评分数据对当前的评分有显著影响，将用户对电影的喜好隐式地反映在时间戳中，有助于得到更精确的结果。本文仅采用强化学习方法中的Q-Learning对SVDPP进行优化，如何能通过融入时间戳对算法直接进行优化，或者将强化学习与其他推荐方法（如深度学习网络）相结合进行优化，将是下一步的研究方向。

参考文献

[1] WANG Guoxia, LIU Heping. Overview of personalized recommendation system[J]. Computer Engineering and Applications, 2012, 48(7): 66-76. (in Chinese)
王国霞，刘贺平. 个性化推荐系统综述[J]. 计算机工程与应用，2012，48(7)：66-76.

[2] GUO Jingjing, MA Jianfeng. Trust recommendation algorithm for Internet of things in virtual community[J]. Journal of Xi'an Dianzi University, 2015, 42(2): 52-57. (in Chinese)
郭晶晶，马建峰. 面向虚拟社区物联网的信任推荐算法[J]. 西安电子科技大学学报，2015，42(2)：52-57.

[3] XIANG Liang. Practice of recommendation system[M]. Beijing: People's Posts and Telecommunications Press, 2012. (in Chinese)
项亮. 推荐系统实践[M]. 北京：人民邮电出版社，2012.

[4] LI Zhuoyuan, ZENG Dan, ZHANG Zhijiang. Research on music recommendation system based on collaborative filtering and music emotion[J]. Industrial Control Computer, 2018, 31(7): 127-128, 131. (in Chinese)
李卓远，曾丹，张之江. 基于协同过滤和音乐情绪的音乐推荐系统研究[J]. 工业控制计算机，2018，31(7)：127-128，131.

[5] WANG Xiaohao, SUN Yanwu, HU Haoming, et al. Commodity recommendation modeling and simulation analysis based on reputation[J]. Computer Knowledge and Technology, 2019, 15(13): 294-296. (in Chinese).
王小豪，孙彦武，胡浩明，等. 基于信誉度的商品推荐建模与仿真分析[J]. 电脑知识与技术，2019，15(13)：294-296.

[6] HERLOCKER J L, KONSTAN J A, TERVEEN L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 1-47.

[7] YIN Hang, CHANG Guiran, WANG Xingwei. K-nearest neighbor collaborative filtering algorithm optimized by clustering algorithm[J]. Journal of Chinese Computer Systems, 2013, 34(4): 806-809. (in Chinese)
尹航，常桂然，王兴伟. 采用聚类算法优化的K近邻协同过滤算法[J]. 小型微型计算机系统，2013，34(4)：806-809.

[8] WANG Yan, LI Fenglian, ZHANG Xueying, et al. An efficient SVD++ algorithm for improving learning rate[J]. Modern Electronics Technology, 2018(3): 35. (in Chinese)
王燕，李凤莲，张雪英，等. 改进学习率的一种高效SVD++算法[J]. 现代电子技术，2018(3)：35.

[9] SUTTON R S, BARTO A G. Reinforcement learning[J]. A Bradford Book, 1998, 15(7): 665-685.

[10] KAELBLING L P, LITTMAN M L, MOORE A W. An introduction to reinforcement learning[J]. IEEE Transactions on Neural Networks, 2005, 16(1): 285-286.

[11] YEHUDA K. Collaborative filtering with temporal dynamics[J]. Communications of the ACM, 2010, 53(4): 1-5.

[12] CHEN Xu, XU Hongteng, ZHANG Yongfeng, et al. Sequential recommendation with user memory networks[C]// Proceedings of the 11th ACM International Conference on Web Search and Data Mining. New York, USA: ACM Press, 2018: 108-116.

[13] LEE J, OH B, YANG J, et al. RLCF: a collaborative filtering approach based on reinforcement learning with sequential ratings[J]. Intelligent Automation and Soft Computing, 2017, 23(3): 439-444.

[14] WANG Xiaolei, CHEN Yunjie, WANG Chen, et al. Virtual network function scheduling method based on Q-learning[J]. Computer Engineering, 2019, 45(2): 64-69. (in Chinese)
王晓雷，陈云杰，王琛，等. 基于Q-learning的虚拟网络功能调度方法[J]. 计算机工程，2019，45(2)：64-69.

[15] KOREN Y, BELL R. Advances in collaborative filtering[M]// RICCI F, ROKACH L, SHAPIRA B, et al. Recommender systems handbook. Berlin, Germany: Springer, 2015: 145-186.

[16] WHITE D J. Markov decision processes[J]. European Journal of Operational Research, 2010, 39(1): 1-16.

[17] ZHANG Wenxu, MA Lei, HE Huilin, et al. Cover study on the ground-space heterogeneous multi-agent cooperation of reinforcement learning[J]. Journal of Intelligent Systems, 2018, 13(2): 202-207. (in Chinese)
张文旭，马磊，贺荟霖，等. 强化学习的地-空异构多智能体协作覆盖研究[J]. 智能系统学报，2018，13(2)：202-207.

[18] ZHOU Yi, CHEN Bo. Method for robot obstacle avoidance based on improved dueling network[J]. Journal of Xi'an Dianzi University, 2019, 46(1): 46-50. (in Chinese)
周翼，陈渤. 一种改进dueling网络的机器人避障方法[J]. 西安电子科技大学学报，2019，46(1)：46-50.

[19] CHEN Xingguo, YU Yang. Reinforcement learning and its application in computer Go[J]. Acta Automatica Sinica, 2016, 42(5): 685-695. (in Chinese)
陈兴国，俞扬. 强化学习及其在电脑围棋中的应用[J]. 自动化学报，2016，42(5)：685-695.

[20] XU Yan, XIA Junjuan, WU Huijun, et al. Q-learning based physical-layer secure game against multiagent attacks[J]. IEEE Access, 2019, 7: 49212-49222.

[21] WIERING M, OTTERLO M. Reinforcement learning[G]// Adaptation, learning, and optimization. Berlin, Germany: Springer, 2012, 12: 3.

编辑　金胡考