Journal of Computer Applications
计算机应用, 2023, 43(1): 15 - 21
ISSN 1001-9081
CODEN JYIIDU
2023-01-10
http://www.joca.cn

文章编号:1001-9081(2023)01-0015-07
DOI:10.11772/j.issn.1001-9081.2021111875

考虑多粒度反馈的多轮对话强化学习推荐算法

姚华勇,叶东毅*,陈昭炯
(福州大学 计算机与大数据学院,福州 350108)
(*通信作者电子邮箱 yiedy@fzu.edu.cn)

摘　要: 多轮对话推荐系统(CRS)以交互的方式获取用户的实时信息,相较于基于协同过滤等的传统推荐方法能够取得更好的推荐效果。然而现有的CRS存在用户偏好捕获不够准确、对话轮数要求过多以及推荐时机不恰当等问题。针对这些问题,提出一种基于深度强化学习且考虑用户多粒度反馈信息的对话推荐算法。不同于现有的CRS,所提算法在每轮对话中同时考虑用户对商品本身以及更细粒度的商品属性的反馈,然后根据收集的多粒度反馈对用户、商品和商品属性特征进行在线更新,并借助深度Q学习网络(DQN)算法分析每轮对话后的环境状态,从而帮助系统作出较为恰当合理的决策动作,使它能够在比较少的对话轮次的情况下分析用户购买商品的原因,更全面地挖掘用户的实时偏好。与对话路径推理(SCPR)算法相比,在Last.fm真实数据集上,算法的15轮推荐成功率提升了46.5%,15轮推荐轮次上缩短了0.314轮;在Yelp真实数据集上,算法保持了相同水平的推荐成功率,但在15轮推荐轮次上缩短了0.51轮。

关键词: 多轮对话推荐系统;反馈信息;深度Q学习网络;偏好挖掘;多粒度
中图分类号: TP181　　**文献标志码:** A

Multi-round conversational reinforcement learning recommendation algorithm via multi-granularity feedback

YAO Huayong, YE Dongyi*, CHEN Zhaojiong
(*College of Computer and Big Data, Fuzhou University, Fuzhou Fujian* 350108, *China*)

Abstract: Multi-round Conversational Recommendation System (CRS) obtains real-time information of users interactively, thus performing better than traditional recommendation methods such as collaborative filtering based method. However, existing CRS suffers from problems inaccurate mining of user preferences, too many conversational rounds required and inappropriate recommendation moments. Aiming at these problems, a new conversational recommendation algorithm based on deep reinforcement learning considering user's multi-granularity feedback information was proposed. Different from existing CRS, in each conversation, the feedback of users on items themselves and more fine-grained item attributes was considered by the proposed algorithm at the same time. Then, users, items and attribute features of items were updated online by using the collected multi-granularity feedback, and the environment state after each round of conversation was analyzed by Deep Q-Network (DQN) algorithm. As a result, more appropriate and reasonable decisions were made by the system, and the reasons of why user buying items were analyzed and the users' real-time preferences were mined comprehensively with fewer conversation rounds. Experimental results on two real datasets show that compared with Simple Conversational Path Reasoning (SCPR) algorithm, the proposed algorithm has the 15 turns success rate increased by 46.5%, and the 15 average turns decreased by 0.314 rounds in Last.fm dataset, while it maintains the same level of success rate but the 15 average turns decreased by 0.51 rounds in Yelp dataset.

Key words: multi-round conversational recommendation system; feedback information; Deep Q-Network (DQN); preference mining; multi-granularity

0　引言

随着互联网快速发展,网上购物成为了当今主流的一种购物方式,但海量的商品种类和款式带来了信息爆炸和信息过载的问题,用户难以在纷繁多样的商品中作出选择,推荐系统[1]应运而生。推荐系统致力于为用户提供个性化推荐,提升网购效率,实现用户与商家双赢。随着技术更迭,推荐系统经历了从静态推荐系统到与用户交互的动态对话推荐系统的发展过程。

静态推荐系统通过收集和分析历史数据,挖掘用户与用户、用户与商品间的内在关联,实现个性化推荐,例如协同过

收稿日期: 2021-11-09;**修回日期:** 2022-05-05;**录用日期:** 2022-05-07。　**基金项目:** 福建省科技计划项目(2018H6010)。
作者简介: 姚华勇(1998—),男,福建南平人,硕士研究生,主要研究方向:推荐算法;　叶东毅(1964—),男,福建泉州人,教授,博士,主要研究方向:机器学习;　陈昭炯(1964—),女,福建福州人,教授,硕士,主要研究方向:机器学习。

滤[2]推荐方法、基于内容的推荐方法[3]。静态推荐系统主要存在两方面局限：1)在缺少新用户行为数据的情况下，系统将面临冷启动这一问题。例如，电影推荐系统刚构建完成时，由于系统中没有任何的用户信息，系统难以为每位新用户推荐他们喜欢的影片，而对于新上映的电影，系统也难以将其准确地推荐给需要的用户。2)用户选购商品时，最终的选择会因商品展示的顺序和方式、所处环境以及周围事物发生改变[4]，静态推荐系统无法及时对变化的用户需求作出响应，导致推荐效果不理想。为克服静态推荐系统的不足，学者提出通过与用户交互以捕获用户变化的对话推荐系统。

对话推荐系统[5-8]是支持用户与系统进行多轮对话并推荐相关目标（商品、歌曲、酒店、餐厅等）的系统，挖掘用户实时偏好是其关键，也是难点。目前，多数对话推荐系统在与用户对话过程中引入商品属性信息（商品的颜色、款式、材质等）以挖掘用户实时偏好，并以限制用户应答自由度的方式（如使用多值应答策略）对用户进行商品属性的询问。这种询问方式相较于将注意力更多集中在自然文本生成的对话推荐系统[9-11]，获取用户信息的效率更高，能够更全面地挖掘用户实时偏好。例如，Christakopoulou等[10]基于贪心思想，在每轮对话后都进行商品推荐，直到用户接受商品，与传统方法相比，该方法提升了推荐的成功率，但是频繁的推荐降低了用户的交互体验感。Dhingra等[12]基于最大熵思想，将对话轮数固定为round，在前round−1轮选取熵最大的属性作为询问属性，并在第round轮推荐商品。该方法对商品属性的利用进行了思考，但固定了对话轮数，缺乏灵活性。在之后的研究中，Sun等[13]针对挖掘用户偏好这一问题，提出对话推荐模型（Coversational Recommendation Model，CRM），使用决策网络生成每轮询问的问题，与最大熵算法相比，CRM在已获得足够多的信息后主动为用户推荐，提高了灵活性。Lei等[14]基于CRM提出EAR（Estimation-Action-Reflection）系统。EAR将询问的商品属性看作多分类任务的标签，以{用户，历史偏好属性}元组作为预训练数据训练决策网络，提高用户历史偏好属性被选择的概率，在预训练完成后，系统根据与用户在线交互获取的奖励信息，基于策略梯度（policy gradient）优化商品属性的权重。该方法在推荐成功率以及推荐效率（平均推荐成功所需的对话轮数更少）这两方面性能均有所提高，但存在同一商品属性反复询问等问题。

针对上述方法存在的不足，本文就如何利用用户反馈信息使决策网络准确挖掘用户偏好以提升推荐准确率展开研究。一方面，为了解决何时推荐、何时询问这一问题并缩短对话轮数，引入强化学习中适用于对话推荐场景等拥有多种状态（用户状态、对话状态等）以及离散系统动作的DQN（Deep Q-Network）[15-16]算法，通过用户与系统交互中的反馈信息（奖励信号）学习最优动作-价值函数，帮助系统作出恰当的选择；另一方面，为了较为准确地挖掘用户实时偏好信息，与以往工作不同，本文提出多粒度反馈模型，同时分析用户对商品以及商品属性的反馈信息，并根据多粒度反馈信息推理用户购买商品的原因。实验结果表明本文算法通过用户的多粒度反馈信息能够较为准确地挖掘用户的偏好，在缩短对话轮数的同时大幅提升推荐成功率。

1 本文算法

1.1 本文算法思想

本文算法的主要流程如图1所示。首先根据用户历史交互信息生成用户、商品和商品属性的初始特征；然后计算观测状态向量并将其作为决策模型的输入，若当前决策模型的决策为推荐商品，则计算用户-商品评分生成推荐商品，推荐成功系统自动退出，推荐失败则收集用户的反馈信息；若当前决策模型的决策为询问属性，则计算用户-属性亲和度生成询问的属性，用户接受属性则继续推荐，用户拒绝属性则收集用户的反馈信息；最后根据用户的多粒度反馈信息更新用户、商品以及商品属性特征，为下一轮对话做准备。

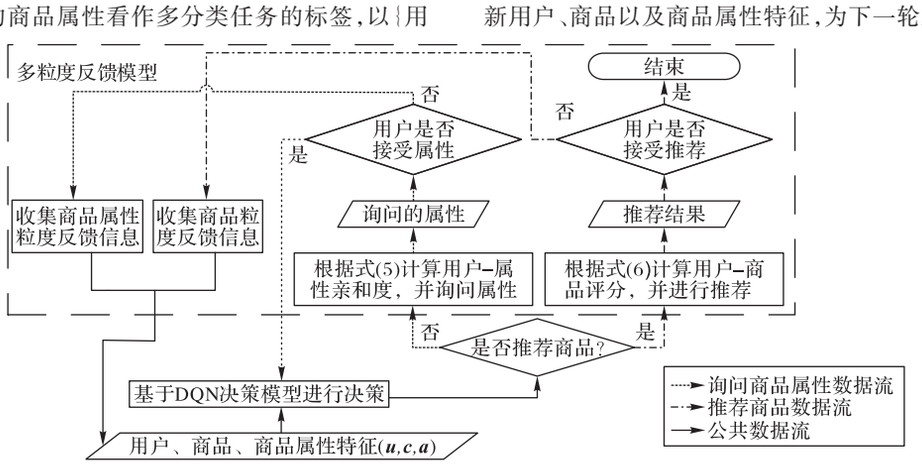

图1 本文算法流程

Fig. 1 Flowchart of the proposed algorithm

1.2 基于DQN的决策模型

以往工作将何时推荐/询问以及询问什么这两类问题进行糅杂，将决策模型看作以商品属性作为输出的多分类任务，直接在决策模型中输出询问的属性以表示当前询问操作，在商品属性数量为千级甚至万级的情况下，模型表现不佳。为了更好地解决这一问题，本文在决策网络中引入无模型（model-free）的DQN算法，将决策的输出动作简化为推荐动作和询问动作，以解决何时推荐、何时询问这一问题。

在多轮对话场景下,设智能体(系统)与环境(用户)在每轮交互中通过环境获得观测状态向量$\boldsymbol{s}$并选择最优动作以获取最大奖励,如文献[14],本文在决策网络中设定了$\boldsymbol{s}_{\text{user}}$、$\boldsymbol{s}_{\text{items}}$、$\boldsymbol{s}_{\text{conver}}$和$\boldsymbol{s}_{\text{attribute}}$共4种状态向量。$\boldsymbol{s}_{\text{user}}$为用户状态向量,根据用户历史交互记录训练生成,用于刻画用户在对话中的状态;$\boldsymbol{s}_{\text{items}}$为商品状态向量,表示系统当前可选的候选商品数;$\boldsymbol{s}_{\text{conver}}$为对话状态向量,记录系统与用户交互的过程,询问属性成功用1表示,询问属性失败用−1表示,推荐成功用2表示,推荐失败用−2表示,超出最大对话论数用0表示;$\boldsymbol{s}_{\text{attribute}}$为商品属性选择状态向量,系统根据用户对商品属性的反馈信息生成属性选择状态向量,使系统明确当前商品属性的选择状况以及用户对不同属性的偏好程度,帮助系统作出恰当的决策。本文将以上状态组合拼接,以找到最适合的状态组合作为决策网络的最终输入(观测状态向量),在2.4节展示单个状态以及状态组合的效果。

DQN作为一种基于价值(value-based)的强化学习方法,期望智能体在与环境的互动中所累积的奖励最大化。本文针对多轮对话推荐场景对智能体获得的奖惩r_t进行设定(表1),设未来每一步奖励的折扣因子为γ,截至T时刻,智能体所获得的总奖励为:

$$R_T = \sum_{t=1}^{T} \gamma^{t-1} r_t \tag{1}$$

表1　决策网络中的奖励设定

Tab. 1　Reward setting of decision-making network

奖励类型	奖惩分数	含义
$r_{\text{ask\_suc}}$	+0.1	用户接受系统所询问的属性
$r_{\text{ask\_fail}}$	−0.1	用户拒绝系统所询问的属性
$r_{\text{rec\_suc}}$	+1.0	用户接受系统所推荐的商品
$r_{\text{rec\_fail}}$	−0.3	用户拒绝系统所推荐的商品
$r_{\text{max\_turn}}$	−0.5	超出最大对话轮数

本文在决策模型中(图2)以用户组合状态向量作为输入,动作的价值(action value)作为输出,使用两层的全连接神经网络(Fully Convolutional Network,FCN)学习最优动作-价值函数。决策网络的输出单元为推荐动作和询问动作。最优动作-价值函数$Q^*(\boldsymbol{s},a)$表示在t时刻下,观测到状态$\boldsymbol{s}$,采取动作a,在所有可能的策略π下所能获得的最大期望回报,根据贝尔曼方程转化为如下形式:

$$Q^*(\boldsymbol{s},a;\ \boldsymbol{\theta}) = E_{s'\sim\varepsilon}\left[r + \gamma \max_{a'} Q^*(\boldsymbol{s}',a';\ \boldsymbol{\theta}) \,\middle|\, \boldsymbol{s},a\right] \tag{2}$$

其中:Q^*是Q网络近似的最优动作价值函数,也是决策网络的核心;$\boldsymbol{\theta}$为Q网络隐藏层中的权值和偏置;r为当前获得的观测状态为$\boldsymbol{s}$且采取动作a时所获得的奖励;ε表示所有状态的集合;$\boldsymbol{s}'$表示下一观测状态。

DQN属于无模型的强化学习方法,需要对数据进行采样。本文使用时序差分(Temporal Difference,TD)对模型采样,在每一时间步(每一次交互)更新模型,相较于蒙特卡洛方法(需要完整的对话过程)更适合于对话推荐场景。模型的损失函数如下:

$$L_i(\boldsymbol{\theta}_i) = E_{s,a\sim\rho(\cdot)}\left[(y_i - Q(\boldsymbol{s},a;\ \boldsymbol{\theta}_i))^2\right] \tag{3}$$

其中:$y_i = E_{s'\sim\varepsilon}\left[r + \gamma \max_{a'} Q(\boldsymbol{s}',a';\ \boldsymbol{\theta}_{i-1})\right]$,对于观测到的轨迹$\{\boldsymbol{s},a,r,\boldsymbol{s}'\}$,模型希望目标期望回报$y_i$与当前时间步的期望回报$Q(\boldsymbol{s},a;\ \boldsymbol{\theta}_i)$尽可能接近;$\boldsymbol{\theta}_i$为第$i$轮迭代中的神经网络隐藏层的权重参数;$\rho(\boldsymbol{s},a)$为状态$\boldsymbol{s}$和动作$a$的概率分布。对网络中的参数进行微分,得到梯度:

$$\nabla_{\boldsymbol{\theta}_i} L_i(\boldsymbol{\theta}_i) = E_{s,a\sim\rho(\cdot);\ s'\sim\varepsilon}\Big[(r + \gamma \max_{a'} Q(\boldsymbol{s}',a';\ \boldsymbol{\theta}_{i-1}) - Q(\boldsymbol{s},a;\ \boldsymbol{\theta}_i))\nabla_{\boldsymbol{\theta}_i} Q(\boldsymbol{s},a;\ \boldsymbol{\theta}_i)\Big]$$

并通过随机梯度下降(Stochastic Gradient Descent,SGD)方法优化模型。

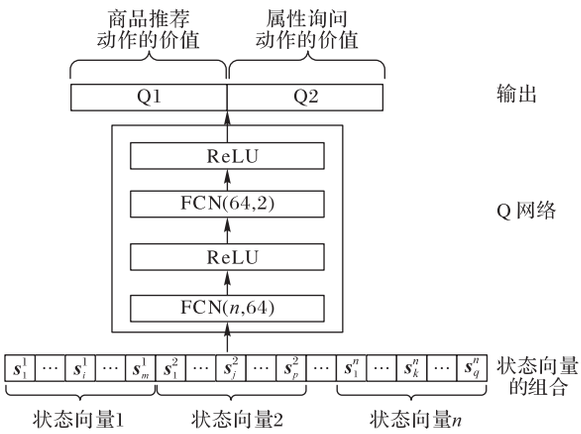

图2　决策模型结构

Fig. 2　Structure of decision model

1.3　多粒度反馈模型

为动态地挖掘用户隐藏的偏好信息,更好地了解用户需求,同时解决推荐什么、询问什么这一问题,本文提出多粒度反馈模型,模型利用“商品”和“商品属性”粒度的用户反馈信息在每次交互后动态地更新用户、商品和商品属性特征,在利用商品反馈信息分辨用户心仪商品的同时,进一步分析用户对更细粒度的商品属性的反馈信息,分析用户购买商品的内在原因。

设用户的实时偏好属性特征向量集(用户在对话过程中已接受的商品属性集)为$\boldsymbol{A}_{\text{u}} = \{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_K\}$,$K$为用户实时偏好属性总数,属性$i$的特征向量$\boldsymbol{a}_i = [a_i^1,a_i^2,\cdots,a_i^n]^{\text{T}}$;所有用户的特征向量表示集为$\boldsymbol{U} = \{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_N\}$,$N$为用户总数,用户$i$的特征向量$\boldsymbol{u}_i = [u_i^1,u_i^2,\cdots,u_i^n]^{\text{T}}$;所有商品的特征向量表示集为$C = \{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_M\}$,$M$为商品总数,商品$i$的特征向量$\boldsymbol{c}_i = [c_i^1,c_i^2,\cdots,c_i^n]^{\text{T}}$;$n$为所有特征向量的维度数。已知用户的实时偏好属性集$\boldsymbol{A}_{\text{u}}$,用户对商品的评分表示为:

$$\hat{f}\left(\boldsymbol{u},\boldsymbol{c}\,\middle|\,\boldsymbol{A}_{\text{u}}\right) = \boldsymbol{u}^{\text{T}}\boldsymbol{c} + \sum_{\boldsymbol{a}_i\in\boldsymbol{A}_{\text{u}}} \boldsymbol{c}^{\text{T}}\boldsymbol{a}_i \tag{4}$$

其中:$\boldsymbol{u},\boldsymbol{c},\boldsymbol{a}_i$分别为用户、商品和商品属性$i$的特征向量表示;$\boldsymbol{u}^{\text{T}}\boldsymbol{c}$表示用户对商品的喜爱程度,$\sum_{\boldsymbol{a}_i\in\boldsymbol{A}_{\text{u}}} \boldsymbol{c}^{\text{T}}\boldsymbol{a}_i$将用户已接受的商品属性与目标商品作内积运算,表示用户偏好属性集与商品的契合程度,两项之和表示在考虑用户实时偏好的情况下用户对商品的综合评分。

用户-属性亲和度表示在已知用户实时偏好属性集$\boldsymbol{A}_{\text{u}}$的

情况下，用户对其他商品属性的亲近程度。例如某用户的偏好属性集中有“周杰伦”，那么该用户对“音乐”这一属性的亲近程度也应该较高。用户-属性亲和度表示如下：

$$\hat{h}\left(\boldsymbol{a}|\boldsymbol{c},\boldsymbol{A}_{\mathrm{u}}\right)=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{a}+\sum_{\boldsymbol{a}_i\in\boldsymbol{A}_{\mathrm{u}}}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}_i \tag{5}$$

其中：$\boldsymbol{u}^{\mathrm{T}}\boldsymbol{a}$表示用户对商品属性的评分，$\sum_{\boldsymbol{a}_i\in\boldsymbol{A}_{\mathrm{u}}}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{a}_i$将用户已接受的商品属性集与目标属性作内积运算，表示用户已接受商品集合中每个商品属性与$\boldsymbol{a}$向量所对应商品属性的亲近程度，两者之和表示在考虑用户实时偏好的情况下，对$\boldsymbol{a}$向量所对应属性的评分。

本文使用贝叶斯个性化排序（Bayesian Personalized Ranking，BPR）[17]优化反馈模型。为了有效区分用户拒绝的商品与待推荐商品，期望用户对拒绝商品的评分尽可能低，对待推荐商品的评分尽可能高，将商品损失L_{item}定义为：

$$L_{\mathrm{item}}=\sum_{(\boldsymbol{u},\boldsymbol{c},\boldsymbol{c}')\in\mathbb{D}_1}-\ln\sigma\left(\hat{f}\left(\boldsymbol{u},\boldsymbol{c}|\boldsymbol{A}_{\mathrm{u}}\right)-\hat{f}\left(\boldsymbol{u},\boldsymbol{c}'|\boldsymbol{A}_{\mathrm{u}}\right)\right)+\lambda_{\Theta_1}\left\|\boldsymbol{\Theta}_1\right\|^2 \tag{6}$$

其中：$\hat{f}\left(\boldsymbol{u},\boldsymbol{c}|\boldsymbol{A}_{\mathrm{u}}\right)$表示用户对待推荐商品的评分，$\hat{f}\left(\boldsymbol{u},\boldsymbol{c}'|\boldsymbol{A}_{\mathrm{u}}\right)$表示用户对已拒绝商品的评分；$\boldsymbol{A}_{\mathrm{u}}$为实时偏好属性特征向量集；$\boldsymbol{C}_{\mathrm{rej}}$为所有拒绝商品的集合；$\boldsymbol{\Theta}_1$为模型参数，$\mathbb{D}_1=\left\{(\boldsymbol{u},\boldsymbol{c},\boldsymbol{c}')|\boldsymbol{c}\in\boldsymbol{C}/\boldsymbol{C}_{\mathrm{rej}};\ \boldsymbol{c}'\in\boldsymbol{C}_{\mathrm{rej}}\right\}$。

同样地，为了有效区分用户拒绝商品属性与待询问商品属性，期望用户对于拒绝商品属性的亲和度尽可能低，对于待询问商品属性亲和度尽可能高，属性损失$L_{\mathrm{attribute}}$定义为：

$$L_{\mathrm{attribute}}=\sum_{(\boldsymbol{u},\boldsymbol{a},\boldsymbol{a}')\in\mathbb{D}_2}-\ln\sigma\left(\hat{h}\left(\boldsymbol{a}|\boldsymbol{u},\boldsymbol{A}_{\mathrm{u}}\right)-\hat{h}\left(\boldsymbol{a}'|\boldsymbol{u},\boldsymbol{A}_{\mathrm{u}}\right)\right)+\lambda_{\Theta_2}\left\|\boldsymbol{\Theta}_2\right\|^2 \tag{7}$$

其中：$\hat{h}\left(\boldsymbol{a}|\boldsymbol{u},\boldsymbol{A}_{\mathrm{u}}\right)$表示用户对待询问属性的亲和度，$\hat{h}\left(\boldsymbol{a}'|\boldsymbol{u},\boldsymbol{A}_{\mathrm{u}}\right)$表示用户对已拒绝属性的亲和度。$\boldsymbol{A}_{\mathrm{rej}}$为用户反馈中所有拒绝属性的集合，$\boldsymbol{\Theta}_2$为模型参数，$\mathbb{D}_2=\left\{(\boldsymbol{u},\boldsymbol{a},\boldsymbol{a}')|\boldsymbol{a}\in\boldsymbol{A}/\boldsymbol{A}_{\mathrm{rej}};\ \boldsymbol{a}'\in\boldsymbol{A}_{\mathrm{rej}}\right\}$。

模型最终的优化函数为：

$$L=L_{\mathrm{item}}+L_{\mathrm{attribute}} \tag{8}$$

本文使用梯度下降法求解优化函数，最小化L以区分用户拒绝的商品集和用户拒绝商品属性集。在每次交互后，用户的特征表示根据两种粒度的反馈信息在动态地变化，即系统对用户有了更进一步的了解，因此在下一轮推荐或询问时，能够选择更加符合用户偏好的商品和商品属性，即以迭代的方式动态学习用户的购买偏好。

1.4 算法流程

本文算法流程如下。

输入 UI表示用户交互记录集，以成对形式表示，例如(bob，cake)；Q表示初始化动作-价值网络；D表示初始化经验回放存储器（Replay buffer）；M表示最大迭代次数；N表示最大对话轮数；$thr\in(0,1)$：阈值；$randomNum\in(0,1)$表示随机值；r表示奖励；

输出 更新后的动作-价值网络Q。

```
for episode ← 1 to M do
    从交互记录 UI 中随机采样一条交互记录(u, i)，随机选取物品 i 所包含的一个商品属性的特征表示 p 作为用户接受的初始商品属性
    for t ← 1 to N do
        计算观测向量 s_t
        生成随机值 randomNum
        a_t = { 随机选取决策动作，        randomNum < thr
              { 根据式(2)选取决策动作，   randomNum ≥ thr
        if a_t = 属性询问 then
            根据式(4)选择 K 个商品进行推荐
        else
            根据式(5)选择询问属性
        end if
        收集用户反馈信息
        根据用户反馈信息得到奖励 r_t
        D ← (s_t, a_t, r_t, s_{t+1})
        从 D 中随机采样 minibatch 个元组(s_j, a_j, r_j, s_{j+1})
        根据式(3)更新 θ
        根据式(8)更新用户特征 u、商品特征 c 和商品属性特征 a、Θ_1、Θ_2
    end for
end for
```

特别要说明的是，决策模型（1.2节）根据收集的多粒度反馈信息获得相应的奖励，同时多粒度反馈模型（1.3节）根据收集的多粒度反馈信息挖掘用户的实时偏好，即决策模型和多粒度反馈模型在得到用户反馈信息后根据对应的损失函数同时进行参数的更新。

在本文算法中，决策模块解决何时推荐、何时询问这一问题，多粒度反馈模型根据商品评分（式(4)）和用户-属性亲和度（式(5)）解决询问什么商品属性、推荐什么商品的问题，并根据用户的多粒度反馈信息（接受/拒绝商品属性、拒绝商品）对用户特征等在线更新。当用户接受系统推荐的商品时，根据用户在对话中的反馈信息以推理用户购买该商品的原因。例如，用户在音乐对话推荐场景下接受了属性“周杰伦”和“流行音乐”，拒绝了属性“摇滚”和“乡村”，最后接受了歌曲“稻香”。由整个对话过程，系统推测用户并不喜欢“摇滚”和“乡村”类型的音乐，而是因为喜欢“周杰伦”并且爱听“流行音乐”才接受了歌曲“稻香”。

2 实验与结果分析

2.1 数据集

本文使用2个真实世界数据集对提出的算法进行验证。Last. fm是关于用户听歌序列的数据集，数据集中包含每个用户的ID、最受他们欢迎的艺术家的列表、音乐类型以及播放次数。Yelp是美国著名商户点评网站，其数据集包含了餐馆、购物、酒店等多领域的数据。为了降低稀疏性，本文删除了交互次数少于10的记录，经过清洗的数据信息如表2所示。Yelp数据集中的属性数目（590个）较多，对其进行二级分类（例如父属性“衣服”的子属性有：“长袖”“T恤”“衬衫”等）。系统在交互过程中询问父属性，并罗列子属性供用户选择。系统采用多值问答策略收集用户的商品属性反馈。在Last. fm数据集中属性数为33，系统在交互过程中只询问

单个属性,用户进行二值应答(喜欢/拒绝该属性)。

表2　数据集介绍

Tab. 2　Introduction of datasets

数据集	用户数	商品数	交互次数	属性数
Yelp	27 675	70 311	1 368 606	590
Last. fm	1 801	7 432	76 693	33

2.2　参数设置

本节介绍模型相关超参数等的设定。本文算法利用Pytorch深度学习框架实现,在显存为12 GB的NVIDIA GPU上进行模型训练。决策模型使用了n个输入、m个输出的双隐层全连接网络学习动作价值函数,n为观测状态向量的维度,m为决策网络的动作总数(询问属性或推荐商品),网络结构中隐藏层单元为64,折扣因子(discount factor)为0.995,使用Adam(Adaptive moment estimation)优化器[18]优化模型,批大小设置为128。系统动作获得的奖惩如表1所示。在多粒度反馈模型中,商品损失优化的学习率λ_{item}设置为0.01,属性损失优化的学习率$\lambda_{\text{attribute}}$设置为0.001,归一化项的权重衰减系数统一设置为0.005。

2.3　基线算法

本文选择以下算法作为基线:

绝对贪心(Absolute Greedy,AG)算法[10]:该算法根据贪心原则,在每轮对话中选择评分最高的k个商品进行推荐。

最大熵(Max Entropy,ME)算法[12]:该算法根据最大熵理论,基于一定规则进行商品推荐和属性询问,其中询问部分选择熵最大的商品属性。

CRM[13]:该算法引入深度强化学习网络帮助系统作出恰当的决策动作,进行多次询问操作,一次推荐操作。

EAR[14]:EAR设计"估计-动作-反射"的三阶段框架以解决对话推荐问题。

SCPR(Simple Conversational Path Reasoning)[19]:该算法根据历史交互记录构建图,在决策网络作出决策后,根据图中当前用户节点所关联的节点,进行推荐和询问操作。

2.4　实验结果对比分析

本文分别选择5轮、10轮以及15轮作为最长对话轮数进行测试。定义指示函数$I(a_i)$(式(9)),a_i为样本i推荐成功时的对话总轮数。算法的评价指标为:SR@k(Success Rate)表示在k轮内,系统在测试集上的推荐成功率;AT@k(Average Turn)表示k轮数内推荐成功的平均轮次,具体计算方式见式(10)~(11)。

$$I(a_i)=\begin{cases}a_i, & a_i \leqslant maxturn\\ n+1, & a_i > maxturn\end{cases} \tag{9}$$

$$I_{\text{SR@}maxturn}=successcount/count \tag{10}$$

$$I_{\text{AT@}maxturn}=\frac{1}{count}\sum_{i=1}^{count}I(a_i) \tag{11}$$

其中:*successcount*为测试集中推荐成功的总数;*count*为测试集样本总数。本文算法期望在尽可能少的轮数内为用户成功推荐商品,平均轮数反映了算法在推荐过程中推荐的精准程度,推荐成功所经过的轮数越短,系统对用户的刻画越准确。

由表3可知,本文算法在Yelp数据集上的表现优于Last. fm数据集。随着最大对话轮数的增加,Last. fm数据集上本文算法的推荐成功率SR与Yelp数据集相比,差距不断减小,由5轮中的48.9个百分点降低至15轮的28.5个百分点;而推荐成功的平均轮数AT正好相反,由5轮相差1.412提高至15轮相差8.086。造成这种现象的原因有两个:1)由于对话轮数的增加,系统能够获取更多与用户相关的信息和偏好,提升推荐成功率,缩小了两个数据集之间推荐成功率的差距;2)Yelp数据集上用户与商品的交互次数和属性数量都远大于Last. fm数据集(见表2),与Last. fm数据集相比,在每轮对话中能够询问多个属性,获得更多的用户信息,因此推荐成功所需的平均轮数更少。

表3　不同算法的比较结果

Tab. 3　Comparison results of different algorithms

数据集	算法	SR@5	AT@5	SR@10	AT@10	SR@15	AT@15
Last. fm	AG	0.039	4.918	0.125	9.470	0.219	13.561
	ME	0.006	4.990	0.113	9.700	0.288	13.607
	CRM	0.014	4.978	0.139	9.562	0.308	13.348
	EAR	0.020	4.976	0.245	9.252	0.432	12.432
	SCPR	0.075	4.850	0.270	9.070	0.465	12.860
	本文算法	**0.182**	**4.422**	**0.459**	**8.246**	**0.681**	**12.546**
Yelp	AG	0.214	4.442	0.301	8.136	0.377	11.345
	ME	0.521	3.915	0.840	5.248	0.919	5.780
	CRM	0.478	3.913	0.789	5.422	0.884	6.115
	EAR	0.623	3.566	0.886	4.606	0.965	5.870
	SCPR	0.650	3.211	0.895	4.520	**0.969**	4.970
	本文算法	**0.671**	**3.010**	**0.917**	**3.933**	0.966	**4.460**

绝对贪心算法、最大熵算法和CRM这三类算法并未考虑用户的反馈信息,因此难以捕获用户的偏好,在两个数据集上的表现较差;EAR考虑到了商品粒度的反馈信息,但是忽略了用户对商品属性粒度的反馈信息;SCPR根据用户历史记录构建静态图,捕获用户、商品和商品属性三者的关系,在决策时根据当前用户节点所连接的邻居节点选择推荐商品和询问属性,在对话轮数较长时相较于其他对比算法有一定提升。本文算法考虑多粒度的用户反馈信息,相较于其他算法能够更加清晰地把握用户购买商品的原因,从实验结果来看,与不考虑反馈信息或只考虑单粒度反馈信息的算法相比,本文算法有一定的优势。

与对话路径推理(SCPR)算法相比,在Last. fm数据集上,算法在SR@5、SR@10和SR@15指标上分别提升了142.7%、70%和86.4%,在AT@5、AT@10和AT@15指标上分别缩短了0.428轮、0.824轮和0.314轮;在Yelp真实数据集上,算法在SR@5和SR@10指标上分别提升了3.2%和2.5%,在AT@5、AT@10和AT@15指标上缩短了0.201轮、0.578轮和0.51轮。针对对话轮数较少时成功率有一定提升的现象,可能是本文算法考虑了多粒度的用户反馈信息,不仅考虑用户对于商品的反馈信息,而且考虑更加细致的用户对于商品属性的反馈信息。这使本文算法在前5轮中,能够捕获到更加精确的用户实时偏好,并且询问的属性更加符

合用户需求，从而在较少对话轮次下能够超越其他算法。本文算法的推荐效果在多数情况下，相较于其他算法取得了一定提升，原因是在每一轮交互后，模型根据用户的多粒度反馈信息对用户特征进行在线更新，以迭代的方式不断学习用户的购买偏好，在区分用户心仪商品的同时进一步分析用户购买商品的原因；同时，文本算法分析利用用户的多粒度反馈信息，提升决策的准确性，缩短对话轮次。

2.5 观测状态对推荐结果的影响

为了研究不同状态向量对决策网络的影响，将单个状态、多个状态的组合作为决策网络的观测状态向量（输入），研究不同状态在5轮、10轮和15轮作为最大对话轮数时对系统的影响。由表4知，在Last. fm数据集中，$\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}$作为输入向量得到的实验效果最佳；$\boldsymbol{s}_{\text{attribute}}$和$\boldsymbol{s}_{\text{conver}}$两种状态单独作为输入向量也取得了较为优异的实验结果；三种与四种状态组合的输入向量在实验结果上不及$\boldsymbol{s}_{\text{attribute}}$和$\boldsymbol{s}_{\text{conver}}$单独作为输入向量。在Yelp数据集上状态的组合数和推荐成功率SR在整体上呈现正比关系，由表2可知在Yelp数据集上，商品属性数为590，与Last. fm数据集相比增加了16.9倍，商品数量为70 311，与Last. fm数据集相比增加了8.5倍，在如此庞大的数据集上进行推荐，需要考虑的环境也更为复杂。实验结果表明，将$\boldsymbol{s}_{\text{user}}$、$\boldsymbol{s}_{\text{attribute}}$、$\boldsymbol{s}_{\text{conver}}$和$\boldsymbol{s}_{\text{items}}$一同作为输入时，推荐准确率SR和平均推荐轮次AT指标均优于其他状态向量作为输入的情况。单独的$\boldsymbol{s}_{\text{user}}$作为决策网络的输入，在两个数据集上模型均无法拟合，这是由于用户与用户之间存在着差异，仅将$\boldsymbol{s}_{\text{user}}$作为输入向量而不给予决策网络关于对话的其他信息，决策网络无法作出恰当的判断（例如，系统根据用户特征了解到用户A与用户B之间存在着差异，但并不了解两者在何处存在差异，因此需要引入其他状态帮助系统进行决策）。

表4 决策网络中不同状态组合作为输入的系统表现

Tab. 4 System performance of taking different state combinations as inputs in decision network

数据集	状态组合	SR@5	AT@5	SR@10	AT@10	SR@15	AT@15
Last. fm	$\boldsymbol{s}_{\text{attribute}}$	0.175	4.422	0.435	8.356	0.655	12.153
	$\boldsymbol{s}_{\text{conver}}$	0.138	4.496	**0.459**	8.483	0.670	12.165
	$\boldsymbol{s}_{\text{items}}$	0.138	4.644	0.385	8.470	0.591	12.622
	$\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}$	**0.182**	4.458	0.412	8.367	**0.681**	**11.546**
	$\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}+\boldsymbol{s}_{\text{attribute}}$	0.144	4.623	0.426	**8.246**	0.632	11.985
	$\boldsymbol{s}_{\text{user}}+\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}+\boldsymbol{s}_{\text{attribute}}$	0.176	**4.390**	0.378	8.335	0.620	12.071
Yelp	$\boldsymbol{s}_{\text{attribute}}$	0.633	3.249	0.886	4.404	0.923	5.519
	$\boldsymbol{s}_{\text{conver}}$	0.562	3.225	0.833	4.723	0.925	5.876
	$\boldsymbol{s}_{\text{items}}$	0.643	3.049	0.865	4.392	0.902	5.340
	$\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}$	0.658	3.063	0.898	4.180	0.955	4.570
	$\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}+\boldsymbol{s}_{\text{attribute}}$	0.665	**3.010**	0.913	4.125	0.962	4.533
	$\boldsymbol{s}_{\text{user}}+\boldsymbol{s}_{\text{items}}+\boldsymbol{s}_{\text{conver}}+\boldsymbol{s}_{\text{attribute}}$	**0.671**	3.080	**0.917**	**3.933**	**0.966**	**4.460**

2.6 多粒度反馈模型对推荐结果的影响

如表5所示，商品反馈在两个数据集上的表现均优于属性反馈，商品反馈与属性反馈的SR、AT指标的差距随着最大对话轮数的增加逐渐变大。在Yelp数据集上，由5轮中SR和AT相差1.6个百分点和0.309轮增加至15轮相差9.4个百分点和0.469轮；在Last. fm数据集上，SR和AT由5轮中相差0.6个百分点和0.126轮增加至15轮相差18.3个百分点和0.662轮。从直观上看，商品反馈直接反映用户的潜在偏好，而描述商品的更细粒度的商品属性反馈通过商品与用户建立关联，并未直接与用户联系，潜在地反映用户喜好。因此随着对话轮数的增加，商品反馈相较于属性反馈，实验表现更为优异。多粒度反馈模型与商品反馈模型相比，在Last. fm数据集上，模型在SR@5、SR@10和SR@15指标上分别提升了29.5%、7.2%和9.7%，在AT@5、AT@10和AT@15指标上分别缩短了0.249轮、0.27轮和0.633轮；在Yelp真实数据集上，模型在SR@5、SR@10和SR@15指标上分别提升了28.4%、13%和3.3%，在AT@5、AT@10和AT@15指标上缩短了0.26轮、0.615轮和1.2轮。与属性反馈模型相比，在Last. fm数据集上，模型在SR@5、SR@10和SR@15指标上分别提升了36.2%、18.5%和55.5%，在AT@5、AT@10和AT@15指标上分别缩短了0.375轮、0.783轮和1.295轮；在Yelp真实数据集上，模型在SR@5、SR@10和SR@15指标上分别提升了32.4%、24%和14.9%，在AT@5、AT@10和AT@15指标上缩短了0.569轮、1.512轮和1.669轮。实验结果表明商品反馈和属性反馈叠加的多粒度反馈模型在提升准确率的同时缩短了对话轮数。

表5 多粒度反馈模型的有效性

Tab. 5 Effectiveness of multi-granularity feedback model

数据集	反馈类型	SR@5	AT@5	SR@10	AT@10	SR@15	AT@15
Last. fm	商品反馈	0.122	4.755	0.388	8.702	0.621	13.179
	属性反馈	0.116	4.881	0.351	9.215	0.438	13.841
	多粒度反馈	**0.158**	**4.506**	**0.416**	**8.432**	**0.681**	**12.546**
Yelp	商品反馈	0.529	3.270	0.792	4.845	0.935	5.660
	属性反馈	0.513	3.579	0.722	5.742	0.841	6.129
	多粒度反馈	**0.679**	**3.010**	**0.895**	**4.230**	**0.966**	**4.460**

2.7 决策模型对推荐结果的影响

为了研究决策模型在系统中的影响，使用随机策略代替决策模型中的Q网络，即在决策时，系统随机作出推荐决策和询问决策。由表6所示，使用随机策略替代Q网络之后，系统在每轮无法作出较好的决策以捕获用户的购买偏好，在两个数据集上推荐成功率大幅下降，推荐成功所需的平均对话轮数也更长，说明了本文决策模型中Q网络的有效性。

表6　消融实验中决策模型的有效性

Tab. 6　Effectiveness of decision-making models in ablation study

数据集	策略	SR@5	AT@5	SR@10	AT@10	SR@15	AT@15
Last. fm	随机策略	0. 004	4. 994	0. 137	9. 612	0. 294	13. 838
	本文策略	**0. 158**	**4. 506**	**0. 416**	**8. 432**	**0. 681**	**12. 546**
Yelp	随机策略	0. 528	4. 236	0. 830	4. 724	0. 902	5. 948
	本文策略	**0. 679**	**3. 010**	**0. 895**	**4. 230**	**0. 966**	**4. 460**

3　结语

本文探讨了如何较为准确地挖掘用户偏好以提升推荐准确率以及缩短对话轮数提升推荐效率的问题,提出利用多粒度的用户反馈信息以更准确地学习用户的购买偏好。引入适用于对话推荐场景的DQN算法充分利用多粒度的反馈信息实现决策网络,提高决策速度,一定程度上解决了以往的工作挖掘用户兴趣不准确的问题,在提高推荐成功率的同时缩短了对话轮次,并通过实验验证了算法的有效性。

未来,本研究考虑利用用户的评论、点赞、位置和社交等其他信息,同时尝试更加有效的网络代替决策网络中的多层感知机模型,以提升推荐准确率和速度;此外,动作空间过大是强化学习研究中不可避免的一大难点。在今后的工作中,本研究将尝试将属性和商品作为系统动作选择,并引入分层强化学习对该问题进行探索和研究。

参考文献 (References)

[1] RESNICK P, VARIAN H R. Recommender systems [J]. Communications of the ACM, 1997, 40(3): 56-58.

[2] GOLDBERG D, NICHOLS D, OKI B M, et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70.

[3] PAZZANI M J, BILLSUS D. Content-based recommendation systems [M]// BRUSILOVSKY P, KOBSA A, NEJDL W. The Adaptive Web: Methods and Strategies of Web Personalization, LNCS 4321. Berlin: Springer, 2007: 325-341.

[4] WANG W, BENBASAT I. Research Note - a contingency approach to investigating the effects of user-system interaction modes of online decision aids [J]. Information Systems Research, 2013, 24(3): 861-876.

[5] JANNACH D, MANZOOR A, CAI W, et al. A survey on conversational recommender systems [J]. ACM Computing Surveys, 2021, 54(5): No. 105.

[6] GAO C, LEI W, HE X, et al. Advances and challenges in conversational recommender systems: A survey [J]. AI Open, 2021, 2: 100-126.

[7] LI R, EBRAHIMI KAHOU S, SCHULZ H, et al. Towards deep conversational recommendations [C]// Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc., 2018: 9748-9758.

[8] ZHANG Y, CHEN X, AI Q, et al. Towards conversational search and recommendation: System ask, user respond [C]// Proceedings of the 27th ACM International Conference on Information and Knowledge Management. New York: ACM, 2018: 177-186.

[9] LIAO L, MA Y, HE X, et al. Knowledge-aware multimodal dialogue systems [C]// Proceedings of the 26th ACM International Conference on Multimedia. New York: ACM, 2018: 801-809.

[10] CHRISTAKOPOULOU K, RADLINSKI F, HOFMANN K. Towards conversational recommender systems [C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 815-824.

[11] ZHOU K, ZHOU Y, ZHAO W X, et al. Towards topic-guided conversational recommender system [EB/OL]. [2020-10-08]. https://arxiv. org/pdf/2010. 04125.

[12] DHINGRA B, LI L, LI X, et al. Towards end-to-end reinforcement learning of dialogue agents for information access [EB/OL]. [2021-09-13]. https://arxiv. org/pdf/1609. 00777.

[13] SUN Y, ZHANG Y. Conversational recommender system [C]// Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2018: 235-244.

[14] LEI W, HE X, MIAO Y, et al. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems [C]// Proceedings of the 13th International Conference on Web Search and Data Mining. New York: ACM, 2020: 304-312.

[15] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning [EB/OL]. (2013-12-19) [2021-07-22]. https://arxiv. org/pdf/1312. 5602.

[16] HOSU I A, REBEDEA T. Playing atari games with deep reinforcement learning and human checkpoint replay [EB/OL]. (2016-07-18) [2021-09-02]. https://arxiv. org/pdf/1607. 05077.

[17] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback [C]// Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. Arlington, VA: AUAI Press, 2009: 452-461.

[18] RUDER S. An overview of gradient descent optimization algorithms [EB/OL]. (2017-06-15) [2021-08-17]. https://arxiv. org/pdf/1609. 04747. pdf.

[19] LEI W, ZHANG G, HE X, et al. Interactive path reasoning on graph for conversational recommendation [C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2020: 2073-2083.

This work is partially supported by Fujian Provincial Science and Technology Project (2018H6010).

YAO Huayong, born in 1998, M. S. candidate. His research interests include recommendation algorithm.

YE Dongyi, born in 1964, Ph. D., professor. His research interests include machine learning.

CHEN Zhaojiong, born in 1964, M. S., professor. Her research interests include machine learning.