一种基于深度强化学习的直播推荐方法*

王潇，刘红岩，车尚锟
（清华大学经济管理学院，北京 100084）

摘　要　近年来，在线直播行业快速兴起，而给用户推荐其感兴趣的直播是提升用户体验的关键。直播推荐有着更强的动态性，直播内容和用户偏好时刻在变化中。现有推荐算法没有针对此特点进行建模。本文基于强化学习理论，提出了一种新型的直播推荐模型。该模型从三个角度构建用户的状态表示。同时，将基于排序的有监督学习策略引入强化学习模型，使得模型在探索学习的同时保证推荐质量。在真实的数据集上的实验评估结果验证了所提模型的有效性。

关键词　推荐系统，深度强化学习，在线直播，有监督学习

中图分类号　TP391.3

1　引言

近年来，在线直播行业快速兴起，观看在线直播成为大众娱乐的重要方式之一。根据艾媒咨询的研究报告[1]，2019 年中国在线直播市场用户规模达到 5.04 亿人，增长率为 10.5%；2020 年中国在线直播市场用户规模达到 5.87 亿人，增长率为 16.5%；2021 年用户预计达 6.35 亿人，增长率为 8.2%。一方面，用户规模的快速提升促进了主播数量的增加、直播内容的丰富，同时也给用户选择感兴趣的直播带来可能；另一方面，用户规模虽逐年提升，但增速已经逐步放缓，当直播平台的用户量趋于稳定后，提升用户体验、增强用户黏性成为平台运营的关键环节。推荐系统的引入可以有效减少用户的搜索时间，帮助用户发现感兴趣的内容，从而提升用户体验。

但是在线直播的推荐面临着诸多挑战：①在线直播具有实时性，主播直播的内容在动态变化，用户也会在主播间随时切换以观看其最喜欢的内容。直播推荐系统必须有能力捕捉用户和主播动态变化的状态。②直播推荐面对的是主播、直播内容和用户的三元组，其中两两之间都具有关联关系，同一个主播不同时段的直播具有共性和特性，同时用户的兴趣既有针对特定主播的也有针对某类内容的，因此合理表示这三者及其之间的关系是直播推荐的重要步骤。已有推荐系统的相关研究没有针对直播的这些特点进行建模，性能上存在着改进的空间。

为解决上述问题，本文将直播领域中的推荐建模转化为一个强化学习问题，提出了一种用于直播推荐的深度强化学习（deep reinforcement learning，DRL）模型。该模型将推荐系统作为智能体，通过系统与用户不断交互的过程探索用户的真实偏好，最大化用户的长期收益。模型采用深度确定性策略梯度算法解决推荐系统问题中动作空间维度高、计算量过大的难题，同时解决了一般的强化学习策略只能推荐一个物品的问题[2]。另外，强化学习通过探索过程最大化用户长期收益，短期通过探索策略会推荐一些与当前偏好不完全吻合的目标，因此会牺牲部分短期收益，可能造成短期用户体验差的问

* 基金项目：国家自然科学基金面上项目（编号：71771131）、国家自然科学基金重大项目（编号：71490724）。
通信作者：刘红岩，清华大学经济管理学院教授、博士生导师，E-mail：liuhy@sem.tsinghua.edu.cn。

题。为此，本文提出了将基于排序的有监督学习策略引入强化学习模型，使得推荐列表能在原有基础上使用监督策略进行改进，缓解这一问题。同时，提出了对用户状态的建模方法，通过静态、动态和实时特征三个角度进行建模，全面反映用户的偏好特征。

本文内容安排如下：第 2 部分总结分析相关研究；第 3 部分定义研究问题；第 4 部分描述所提出的模型；第 5 部分通过实验评估所提模型的性能；最后第 6 部分总结全文。

2 相关研究

推荐模型可以分为协同过滤（collaborative filtering，CF）[3]、基于内容的推荐（content-based recommendation）[4]及混合方法。下面对经典的通用推荐模型以及与本文工作相关的视频推荐、直播推荐和基于强化学习的推荐模型进行总结分析。

2.1 通用推荐方法

在通用推荐模型中，比较经典的方法是协同过滤和基于内容的推荐。

协同过滤的基本思想是向用户推荐与其有着相似喜好的用户所喜欢的物品[3]。该方法基于用户的历史行为信息，如用户购买哪些商品或者用户对已有商品的打分信息计算用户之间的相似度。然后将相似用户购买过的商品推荐给目标用户。

基于内容的推荐是向用户推荐其喜好物品的相似物品[4]。如果用户喜欢某个物品，则与该物品类似的物品也会被推荐给用户。该方法的难点在于需要找到准确的特征以描述物品，同时该方法可能推荐的都是与已消费物品类似的物品。

经典的协同过滤方法和基于内容的推荐方法没有考虑用户行为的时序特点，因此，考虑用户行为的序列模式的推荐方法即序列推荐（sequential recommendation）吸引了很多研究者进行研究。早期的序列推荐方法利用马尔科夫链建模用户的序列行为[5]，建立在较强的马尔科夫性质的假设前提下。随着深度学习的发展，研究者提出了许多基于神经网络模型的推荐方法，例如，Hidasi 等研究者[6]采用循环神经网络（recurrent neural networks，RNN）模型来建模会话中的用户点击序列。Wu 等研究者[7]采用图神经网络建模，将全局偏好和当前会话偏好结合，对下一物品进行预测。Ying 等研究者[8]采用基于层级的注意力网络，结合了用户长短期的偏好进行预测，将用户长期的偏好的变化加入考虑。

近年来，强化学习在游戏领域和自动控制领域取得了良好的效果[9~11]。因此深度强化如何应用于推荐系统也成为研究热点。Zhao 等学者提出基于深度强化学习的用于电商平台的商品推荐算法，让推荐系统智能地学习最优推荐策略[12~14]。区别于其他应用中智能体每一步与环境的交互都可以得到反馈[15]，在推荐系统中，获得任意一步动作的反馈的代价是较高的。因此已有深度强化推荐工作中采用了环境模拟器来根据协同过滤的思想来预测反馈值。但该方法的不足之处在于，若环境模拟器的预测值与真实反馈值有偏差，则会影响到强化学习的学习效果。在同样的框架下，Zhao 等学者研究在电商平台下如何为用户推荐多个商品并排版成网页的问题，但该研究主要侧重于使用 Encoder-Decoder 模型进行商品页面的生成[13]。Zheng 等构建了基于深度强化学习的框架进行新闻的推荐，侧重解决推荐物品过于相似和重复的问题[16]。

Wang 等学者的研究结合了监督学习和强化学习，采用 Actor-Critic 模型来克服仅采用强化学习模型时在探索时期推荐不准确的缺陷[17]。但该方法的不足之处在于，其定义的动作空间维度与物品维度相同，在应用于直播领域时，由于待推荐的物品数量很多，会造成计算复杂度很高，同时网络的参数也会随动作维度的增大而增大，使得模型求解复杂。Liu 等同样采用了深度确定性策略梯度算法，提出了

三种状态表示模型来建模物品之间的联系以及用户和物品之间的联系[18]。Chen 等学者提出两种算法来缓解由用户、物品分布变化引起的反馈不准确的问题，提出分层抽样回放和近似悔恨反馈法来有效地估计反馈值[19]。

2.2 视频推荐

在视频推荐领域，基于协同过滤的思想根据用户的行为进行分析，从相似用户的角度对用户进行推荐。基于内容的推荐则利用了视频的一些元信息，如标题和风格；或者视频中的信息，如色彩和明暗。

Davidson 等分析了 YouTube 的视频推荐系统，该系统采用的输入包括了内容相关信息和用户相关信息，后者包括了直接和间接的回馈。直接回馈包括喜爱和厌恶等行为，间接回馈包括浏览和观看等行为[20]。随着深度学习的发展，Covington 等应用了深度学习进行视频推荐，深度神经网络的优点在于可以方便地处理离散和连续变量，可将用户观看历史、搜索记录、场景信息及用户画像共同作为输入，并输出用户的向量表示[21]。基于内容的推荐还可以利用视频信息，如 Mei 等在研究中利用了视频的文本信息如描述和标签等[22]。Deldjoo 等同样采用基于内容的推荐，提出了一种能分析视频风格特征的推荐方法[23]。

2.3 直播推荐

由于直播行业近些年才流行起来，目前直播推荐相关的研究还很少。根据 Yang 等的研究[24]，直播平台 Twitch 采取了最多观看（most viewed）的推荐手段，该方法的缺陷是没有考虑到用户的个人偏好，即每个用户的不同点。由此该文章提出了 HyPAR（hybrid preference-aware recommendation）算法，加入了对用户历史观看记录信息的利用，包括观看频道、观看时长，以此来分析用户的喜好。Liu 等的研究[25]着重考虑了直播平台中的关注列表信息，该方法首先对用户观看记录进行分析并用 k-means 方法聚类，而后基于用户群进行推荐。

由此可见，目前已有的针对直播的推荐方法没有充分利用直播推荐的特点。如果采用已有的视频推荐算法做直播推荐，则忽略了直播内容实时变化的特点，直播视频的动态变化影响着用户的选择，但是在视频推荐中没有考虑这点。此外，直播视频都是实时的，这使得直播推荐算法无法利用完整的视频信息。另外，深度强化推荐模型这类新方法仍然主要应用于商品推荐，没有针对直播场景进行优化，同时，已有模型采用的离散动作空间的定义方式使得在推荐的场景下计算复杂度很高[19]。此外，部分已有强化推荐系统采用深度 Q 学习在每一步只能推荐一个物品，以及采用基于分类的监督学习算法[17]不能很好地解决本质上是多个物品排序的推荐问题。因此，本文针对直播推荐的应用场景，研究上述问题的解决方法。

3 问题定义

3.1 直播场景下的推荐问题

假设在推荐系统中有 M 个用户，用集合 U 表示；有 N 个主播用集合 V 表示。令 U_t、V_t 分别表示 t 时刻在线的用户和主播集合。对于任何一个用户 $u \in U$，给定其历史观看行为记录，推荐问题是预测其下一时刻可能感兴趣的直播，为其生成一个长度为 K 的主播推荐列表。

3.2 直播推荐的强化学习建模

强化学习的目的是教会智能体（agent）如何去决策（action），每一个决策会影响智能体未来的状态（state），智能体采取的每一步决策会产生反馈（reward），反馈值越高即表示收益越高。在直播推荐领域中，推荐系统可以看作类似的一个智能体，可以由图 1 来表示，推荐系统收到用户当前的状态表示 s_t，并根据策略做出决策，推荐用户喜好的直播列表（在图中对应动作 a_t），用户将对推荐系统的每个决策做出反馈 r_t，此时，用户达到下一个状态 s_{t+1}。

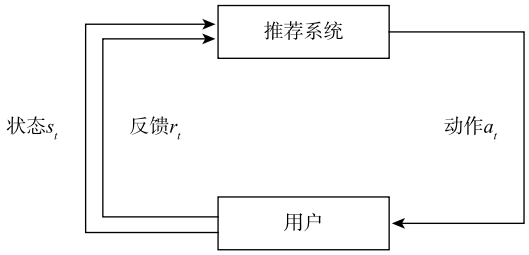


图 1 强化学习与推荐系统交互模型

给定某用户 t 时刻的状态 s_t 后，假设其未来的状态与过去的状态是独立的，当前时刻的推荐行为 a_t 只与当前状态 s_t 有关，而与之前的状态无关，则此过程符合马尔科夫决策过程（Markov decision process）的定义，因此我们可以将直播推荐问题建模为一个马尔科夫决策过程，由状态、动作和反馈的序列组成，可以由五元组（S，A，P，R，γ）表示，定义如下。

状态空间 S：用户当前状态的向量表示，用户在时刻 t 的状态为 s_t。

动作空间 A：推荐系统在时刻 t 的动作记为 a_t。在本文中，为了提升计算效率，将动作空间建模为连续空间。为了得到推荐列表，将 a_t 建模为由稠密向量表达的用户当前偏好。基于该向量与各个主播偏好向量的匹配可以得到推荐列表。本文采用的连续动作空间有着计算效率的优势，如果将动作空间定义为离散空间，计算复杂度很高[22]。

反馈 R：$S \times A \to R$ 表示反馈函数 $r(s,a)$，表示在状态 s 下采用动作 a 得到的反馈。推荐系统根据动作 a 推荐一个主播列表后，若用户观看了列表中的主播，则反馈取值为正。

状态转移概率 P：$p(s_{t+1} \mid s_t, a_t)$ 定义了由状态 s_t 采取动作 a_t，达到状态 s_{t+1} 的概率。

折现因子（discount factor）γ：γ 是[0,1]区间的实数，表示未来收益的折现率。特别地，若 $\gamma = 1$，意味着未来得到的收益与当前价值等同。若 $\gamma = 0$，则意味着未来得到的回报在现在毫无价值，智能体可被看作“短视”的。

为了衡量推荐系统在一段较长时间内的推荐效果，定义模型的总期望收益为 Q 函数（state action value function）：$Q^{\pi}(s,a)$ 表示在时刻 t 状态 s 下执行动作 a，并在接下来采取策略 π 的总期望收益。γ 表示折现因子，则

$$Q^{\pi}(s,a) = E\left(r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \cdots | s,a\right) \tag{1}$$

假设状态 s 下执行动作 a 得到的反馈为 r，则 Bellman 方程可给出 Q 函数的迭代形式：

$$Q^{\pi}(s,a) = E_{s'}\left(r + \gamma Q^{\pi}(s',a') \mid s,a\right) \tag{2}$$

采用值迭代（value iteration）的方法可以求出最优 Q 函数 $Q^{*}(s,a)$：

$$Q^{*}(s,a) = E_{s'}\left(r + \gamma \max_{a'} Q^{*}(s',a') \mid s,a\right) \tag{3}$$

3.3 强化学习的模型选择

由于上述的强化学习模型中对于 Q 函数的求解，以及策略函数 π 的求解比较困难，而深度强化学习则结合了强化学习与深度神经网络，采用了深度神经网络来表示 Q 函数或策略，并通过梯度下降的方法完成端到端的 Q 函数或策略的优化，有效解决了上述问题，因此，本研究中采用深度强化学习的模型来求解直播推荐问题。具体来说，本文采用基于策略梯度学习的深度确定性策略梯度模型[26]，并在此基础上进行改进。深度确定性策略梯度模型可以分为两部分：Actor 模型和 Critic 模型。Actor 模型学习策略而 Critic 模型学习 Q 函数。采用神经网络来进行策略近似和 Q 函数近似可以进行高维非线性的近似，能够有效处理连续动作空间。

本文采用参数为 w 的深度神经网络模型 Q-network 来表示 Q 函数，采用深度 Q 学习的方法估计最优的 Q 函数值，即

$$Q(s,a,w)\approx Q^{*}(s,a) \tag{4}$$

目标是用深度 Q-network 来近似最优 Q 函数，所以令最优 Q 函数的 Bellman 方程[式（2）]的右半部分为 Q-network 模型的学习目标 Q-target，即

$$Q\text{-target}=r+\gamma\max_{a'}Q(s',a',w) \tag{5}$$

本文用深度策略网络来近似表示策略，考虑确定性策略（deterministic policy gradient）的情况[27]。用含有参数 θ 的深度网络 μ 来表示策略，则在状态 s 时的动作为

$$a=\mu(s,\theta) \tag{6}$$

假设时刻 t 的收益为 r_t，定义收益函数为

$$J(\theta)=E\left[r_1+\gamma r_2+\gamma^2 r_3\cdots\mid\mu(s,\theta)\right] \tag{7}$$

本文使用的主要变量符号如表 1 所示。

表 1 直播推荐问题中的主要变量符号

符号	含义
T	时刻的数目，推荐问题总共有 T 个时刻
t	时刻的取值，从集合 $\{1,2,\cdots,T\}$ 中取值
M	直播平台中的用户数目
U	用户集合
U_t	t 时刻在线的用户集合
N	直播平台中的主播数目
V	主播集合
V_t	t 时刻在线的主播集合
h	表示每个用户和主播所用的向量空间（嵌入向量）的维度
H_u	用户 u 的嵌入向量，维度为 h
H_v	主播 v 的嵌入向量，维度为 h
$v_{u,t}$	用户 u 在时刻 t 观看的主播
o_t	t 时刻的观测信息，指 t 时刻起始时已知的推荐系统中与用户相关的所有信息
s_t	用户在 t 时刻起始时的状态表示，从观测信息 o_t 中提取得到
$f(\bullet)$	$s_t=f(o_t)$，指将 t 时刻的观测信息映射到用户状态表示的函数

续表

符号	含义
a_t	在 t 时刻推荐系统给用户做出的推荐行为
$\pi(\bullet)$	$a=\pi(s)$，表示一个从状态 s 到推荐动作 a 的策略函数
r_t	在 t 时刻推荐行为 a_t 给用户带来的收益，反映推荐行为 a 是否准确

4 SDRIV 模型

针对直播推荐应用场景中直播的实时特点，本文在深度确定性策略梯度的 Actor 模型部分引入反映直播实时特点的特征。同时，为了更有效地对网络进行学习，本文引入了基于排序的有监督的学习策略。结合这两点，本文提出了基于有监督深度强化学习的直播推荐模型 SDRIV（supervised deep reinforcement based live video recommendation）。

4.1 模型概述

由于强化学习探索的特点，其产生的相对随机的动作可能会影响推荐系统的表现，带来负面的用户体验[19]，因此本文采用了融合有监督学习和强化学习的方法，来确保模型在探索的同时也保证了推荐的质量。受基于排序的推荐模型的启发[28]，本文提出了如何将基于排序的有监督学习与强化学习相结合的方法。同时，本文认为用户状态的表示（state）是强化学习推荐模型具有良好推荐效果的关键因素，提出从用户的静态特征、动态特征和实时特征三个角度来对用户建模。这些信息将从用户的历史观看记录和当前观看直播的情况来提炼。

本文提出的 SDRIV 模型的框架由图 2 表示，图中 Actor 模型中的两个全连接层（FC layer）由下至上分别为 ReLU 和 tanh；Critic 模型中的两个全连接层（FC layer）都为 ReLU。根据上述的介绍，先提炼用户的状态表示（state s）作为 Actor 模型的输入。图中 B_t 代表了 t 时刻用户的三元组信息集合，每个用户的三元组中包含用户编号、用户观看过的主播编号、用户未观看过的主播编号，将 B_t 引入模型中，是为了实现基于排序的有监督学习。Actor 网络的输出为动作 a（Action a），a 将由策略近似和基于排序的有监督学习[28]共同学习得到。推荐系统根据动作 a 为用户推荐一个主播列表，收到反馈 r（reward r），用于 Critic 网络的学习。Critic 网络用于实现 Q 函数的近似，其输入包括了用户状态表示 s、动作 a 和收集到的反馈 r。

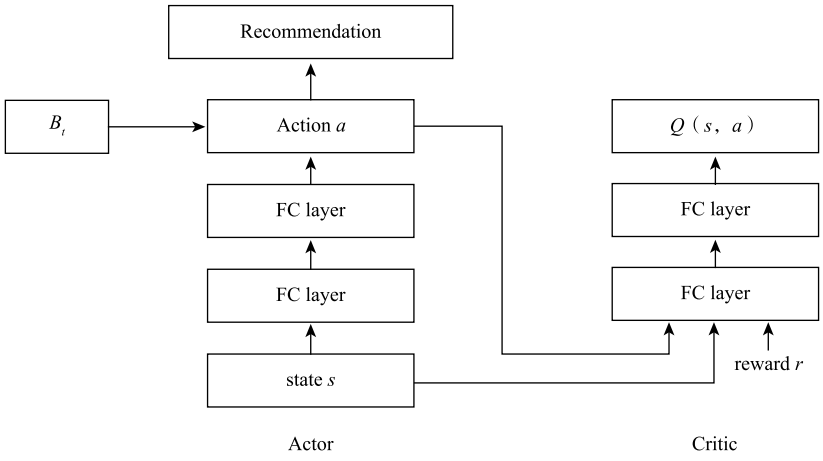


图 2 SDRIV 模型

4.2 用户状态表示模型

用户状态表示是 Actor 模型的输入部分。本文提出从用户的静态特征、动态特征和实时特征三个方面共同构建状态表示。静态特征反映的是用户长期的、稳定的兴趣，动态特征代表用户的动态偏好。两类特征通过所提模型学习得到，这种定义和处理方式与已有文献的处理方式一致[29, 30]。实时特征将当前在线的相似用户的偏好信息加以考虑，反映推荐时刻的实时信息。这三个方面分别从对时间敏感度不同的三个角度来提取特征，用以表示当前的状态。图 3 展示了用户 u 的状态表示模型。在本文中，无论是静态特征还是动态特征，都不是通过显式的特征反映，而是通过学习得到隐式向量（latent vector）来表达，且是与整个模型中的其他参数一起通过学习得到的。

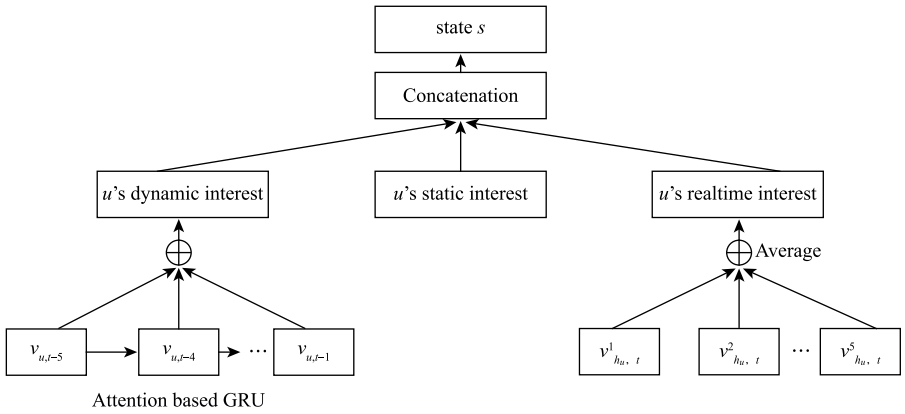


图 3 用户状态表示模型

第一部分是用户的静态特征，由嵌入向量（embedding vector）表示。通过模型的嵌入层（embedding layer），将用户 u 的编号（ID）作为输入，映射到维度为 h 的向量空间，得到向量 H_u。同时，每个主播 v 也通过模型的嵌入层得到其嵌入向量表示，记为 H_v。

第二部分是用户的动态特征，是基于用户的历史行为序列学习到的用户行为的序列模式特征，反映用户偏好的动态变化趋势。本文采用了用户最近 m 次观看的主播作为本部分的输入，用户 u 的观看历史序列记为 $<v_{u,\ t-1},\cdots,v_{u,\ t-m}>$。在网络中，这 m 个主播将首先进行嵌入学习，得到其向量表示，作为门循环网络（gated recurrent network，GRU）部分的输入。循环神经网络可以有效学习序列前后元素相互依赖的特征。且由于本文处理的序列较短，GRU 参数较少，训练速度快，比 LSTM（long short-term memory，长短期记忆网络）更适合于本问题。在此之上，本文采用了注意力机制的方法[31]，通过赋予输入序列中每个元素不同的权重，可以进一步学习到输出对输入的每个部分的不同依赖程度。

第三部分是用户的实时特征，通过相似用户当前的观看行为获取。本文基于用户观看直播的历史行为，通过预训练一个基于排序的有监督学习模型得到用户的特征向量，在此之上通过内积的方式确定用户的相似用户。设 h_u 为用户 u 的相似用户的有序列表（根据相似度由高到低的顺序进行排序），$v_{h_u,t}$ 为在线的每个相似用户正在观看的主播，取前 n 个主播记为 $v^1_{h_u,t},v^2_{h_u,t},\cdots,v^n_{h_u,t}$。对主播进行嵌入学习，得到其向量表示后，取平均作为用户的实时特征表示。

将三部分得到的特征向量拼接，即用户 u 当前的状态表示，而这也将作为 Actor 网络的输入。用户表示模型的参数与 Actor 网络中的参数共同训练学习。

4.3 Actor 模型

Actor 模型用参数为 θ 的神经网络 μ 来表示策略，结合了策略近似和基于排序的有监督学习两个部分，定义 $J_{\mathrm{RL}}(\theta)$ 为强化学习部分的目标函数，$J_{\mathrm{RS}}(\theta)$ 为有监督学习的目标函数，$\epsilon \in [0,1]$ 用来调节两者的权重。则 Actor 模型的目标函数为

$$J(\theta) = \epsilon \bullet J_{\mathrm{RL}}(\theta) + (1-\epsilon) \bullet J_{\mathrm{RS}}(\theta) \tag{8}$$

本文采用梯度上升的方法来最大化 Actor 目标函数 $J(\theta)$。由于 $J(\theta)$ 可表示为 $J_{\mathrm{RL}}(\theta)$ 和 $J_{\mathrm{RS}}(\theta)$ 的线性组合，下面将分为强化学习和有监督学习两个部分，分别求对参数 θ 的偏导，最后得出整体的更新公式。

对于强化学习的部分，Actor 网络通过参数 θ 来更改策略 $\mu_\theta(s)$，使得 $Q(s,a)$ 的期望值达到最大。由式（6）知：$a = \mu(s,\theta)$，假设 w 是由 Critic 模型给出的深度 Q-network 的参数，则目标函数 $J_{\mathrm{RL}}(\theta)$ 表示如下：

$$\max J_{\mathrm{RL}}(\theta) = E\left[Q\left(s,\mu_\theta(s),w\right)\right] \tag{9}$$

采用链式法则对 θ 求偏导后得到梯度公式：

$$\nabla_\theta J_{\mathrm{RL}}(\theta) = E\left[\nabla_\theta \mu_\theta(s) \bullet \nabla_a Q(s,a,w)\right] \tag{10}$$

模型的有监督学习部分采用的是基于排序的有监督学习方法。在本文中，我们认为用户对正在观看的直播的偏好高于随机抽取的未观看过的直播。因此，对于一个用户，其在时刻 t 正在观看的主播被当作正例，在其他主播中随机采样 5 个主播作为负例，则有监督学习的目的是最大化观看正例与观看负例的概率之差。对于每个用户 u，可以构建三元组的集合 $B_{u,t}$，以及 t 时刻全部用户的三元组的集合 B_t，具体定义为

$$\begin{aligned} B_{u,t} &= \left\{(u,i,j) | i = v_{u,t}, j \in V_t \setminus i\right\} \\ B_t &= U_{u \in U_t} B_{u,t} \end{aligned} \tag{11}$$

则有监督学习的目标函数可以表示为

$$\begin{aligned} \max J_{\mathrm{RS}}(\theta) &= \ln p(\theta | >_u) = \Sigma_{(u,i,j) \in B_t} \ \ln \sigma\left(\hat{x}_{uij}\right) - \lambda_\theta \left\|\theta\right\|^2 \\ \hat{x}_{uij} &= x_{ui} - x_{uj} = a^T \bullet H_i - a^T \bullet H_j \end{aligned} \tag{12}$$

采用链式法则对 θ 求偏导后得到梯度公式：

$$\nabla_\theta J_{\mathrm{RS}}(\theta) = E\left[\Sigma_{(u,i,j) \in B_t} \frac{e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \bullet \nabla_a \hat{x}_{uij} \bullet \nabla_\theta \mu_\theta(s)\right] \tag{13}$$

将强化学习与有监督学习的公式合并，采用梯度上升更新 Actor 模型部分参数 θ，假设学习率为 α_θ，梯度更新公式为

$$\theta \leftarrow \theta + \alpha_\theta \bullet \left[\epsilon \bullet \nabla_\theta J_{\mathrm{RL}}(\theta) + (1-\epsilon) \bullet \nabla_\theta J_{\mathrm{RS}}(\theta)\right] \tag{14}$$

对于每个用户 u，Actor 模型的输出是动作 a，用户当前对主播的偏好向量与表示主播的向量有着相同的维度。通过做内积的方式可以求得对每个主播的偏好分数。对于主播 v，其分数计算方式为

$$\mathrm{score}_v = a^T \bullet H_v \tag{15}$$

分数在前 K 的主播构成推荐列表 G，推荐给用户，若 G 中包含了用户下一时刻观看的主播 $v_{u,t}$，则 $\mathrm{rank}\left(v_{u,t}\right)$ 表示 $v_{u,t}$ 在推荐列表中的排位序号。本文的反馈机制表示如下：

$$r=\begin{cases}1-\dfrac{\mathrm{rank}\left(v_{u,\ t}\right)}{K}, & \text{if } v_{u,t}\in G\\ -1, & \text{otherwise}\end{cases} \tag{16}$$

4.4 Critic 模型

Critic 网络的目标是实现 Q 函数近似，其输入包括了用户状态表示 s、动作 a 和收集到的反馈 r。设神经网络的参数为 w，采用 $Q(s,a,w)$ 对 $Q^*(s,a)$ 进行近似。

则 Critic 网络的损失函数为

$$L(w)=\frac{1}{2}\bullet E_{s,a,r,s'\sim D}\left[\left(r+\gamma\max_{a'}Q(s',a',w')-Q(s,a,w)\right)^2\right] \tag{17}$$
$$a'=\mu(s'\mid\theta')$$

其中，D 是智能体的经验 $e_t=(s_t,a_t,r_t,s_{t+1})$ 的集合。本文采用了基于经验回放的训练方式[30]。将智能体的每条经验 $e_t=(s_t,a_t,r_t,s_{t+1})$ 存储到记忆 D 之中。模型训练过程中，每次从 D 中抽取一个随机的小批次 $\{(s,a,r,s')\}$ 进行模型训练。

本文采用固定 target 网络的方法[32]以减小更新 Q 函数时造成的策略的大幅度波动，并采用了与 Lillicrap 等的研究中相一致的一种缓速更新（soft update）[26]。对于深度确定性策略梯度算法的两个近似目标 $Q(s,\ a\mid w)$ 和 $\mu(s\mid u)$，在某时刻复制 Critic 和 Actor 模型作为 target 网络 $Q(s,a\mid w')$ 和 $\mu(s\mid\theta')$ 并缓慢地更新，如式（18）所示，其中 $\tau\ll 1$。

$$w'\leftarrow\tau w+(1-\tau)w';\ \ \theta'\leftarrow\tau\theta+(1-\tau)\theta' \tag{18}$$

Critic 网络的目标是最小化 $L(w)$，故而采用梯度下降的方法，采用链式法则对 w 求偏导可得到梯度公式：

$$\frac{\partial L(w)}{\partial w}=E_{s,a,r,s'\sim D}\left[-\left(r+\gamma\max_{a'}Q(s',a',w')-Q(s,a,w)\right)\frac{\partial Q(s,a,w)}{\partial w}\right] \tag{19}$$

令 α_w 为 Critic 网络的学习率，参数 w 的梯度更新公式为

$$w\leftarrow w-\alpha_w\bullet\frac{\partial L(w)}{\partial w} \tag{20}$$

4.5 SDRIV 算法

对于强化学习的探索部分，随机的动作所生成的推荐列表可能会影响到用户的体验。本文利用了深度确定性策略梯度算法的优势，即探索与模型的学习可以分开独立处理。本文采取的方法是在已有的动作上添加一个高斯随机变量 $\zeta,\zeta\sim N(0,\sigma^2 I)$，其中 σ 为随训练次数指数衰退的参数。

$$\mu'(s)=\mu(s|\theta)+\zeta \tag{21}$$

根据上述策略，模型会在训练初期进行较为大幅的探索，而随着训练的进行，策略趋近最优策略，探索幅度减小。$\mu'(s)$ 代表了带有衰退探索机制的策略，在模型测试时，策略将完全由 $\mu(s|\theta)$ 给出，并不会加入随机变量 ζ。

区别于传统的强化学习训练方法，本文采用的是结合监督学习与强化学习的训练手段，对于每一条训练样本都需要有标签与其对应，故而每一条经验的产生都要基于一个真实的用户记录。设 o_t 为

一个 t 时刻的用户记录，则 o_t 包含了用户的 ID 信息、观看记录和时刻 t 下推荐系统的一些特征信息。用户的状态 s_t 可由 $s_t = f(o_t)$ 得到。令 O 表示全部用户记录 o 的集合。表 2 给出了本文提出的 SDRIV 算法的主要步骤。

表 2 SDRIV 算法

算法：SDRIV

Randomly initialize critic network $Q(s,a,w)$ and actor network $\mu(s,\theta)$ with w, θ

Initialize target network Q, μ with parameters $w' \leftarrow w$, $\theta' \leftarrow \theta$

Initialize replay buffer D

For epoch = 1, M **do**

  For observation o_t in O **do**

    Obtain $s_t = f(o_t)$

    Select action $a_t = \mu(s_t|\theta) + \zeta$

    Execute action a_t and obtain reward r_t and new state s_{t+1}

    Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer D

    Sample a random minibatch (s_i, a_i, r_i, s_{i+1}) from D

    Set $y_i = r_i + \gamma Q(s_{i+1}, \mu'(s_{i+1}, \theta'), w')$

    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i, w))^2$

    Update the actor policy using the sampled policy gradient:

$$\nabla_\theta J \approx \frac{1}{N}\sum_i \left(\epsilon \bullet \nabla_a Q(s,a,w) + (1-\epsilon) \bullet \Sigma_{(u,i,j)\in B} \frac{e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \nabla_a \hat{x}_{uij} \right) \nabla_\theta \mu(s,\theta)$$

    Update the target networks:

$$w' \leftarrow \tau w + (1-\tau) w'$$

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta'$$

  end for

end for

5 实验

5.1 数据集与预处理

本文采用的数据集是某在线直播平台 14 天的直播数据，数据包括了用户 ID、主播 ID、直播 ID、观看直播的起始时间及终止时间。在清洗数据时，本文剔除了平均每天观看时长大于 12 小时的用户，最后保留了观看时长超过 300 秒的用户。经过上述操作后，本文选取了观看直播次数较多的 1 781 位用户，以及直播次数较多的 1 012 位主播，在这 14 天共 191 112 条观看记录。将前 11 天数据作为训练集，第 12 天数据作为验证集，最后 2 天的数据作为测试集。

5.2 评价指标

本文采用了与基于会话推荐的文章相一致且广泛应用的两个指标，Recall@K 和 MRR@K。令 m 表示测试用例的个数，G 表示推荐列表。

Recall@K：召回率衡量了用户观看的主播在推荐列表中的测试用例占全部测试用例的比例。其计算方法如下：

$$\text{Recall@}\left(K=\frac{n_{\text{hit}}}{m}\right) \tag{22}$$

其中，n_{hit} 为用户观看的主播在推荐列表中的测试用例的个数。

MRR@K：MRR 计算了用户观看的主播在推荐列表中排名的倒数的平均值，若排名超出 K，则被设为 0。该评价指标将用户真正观看的主播在推荐列表中的排序纳入了考量。其计算方法如下，假设用户 t 时刻观看的主播为 v_t，$\text{Rank}(v_t)$ 表示其在推荐列表中的排名，则

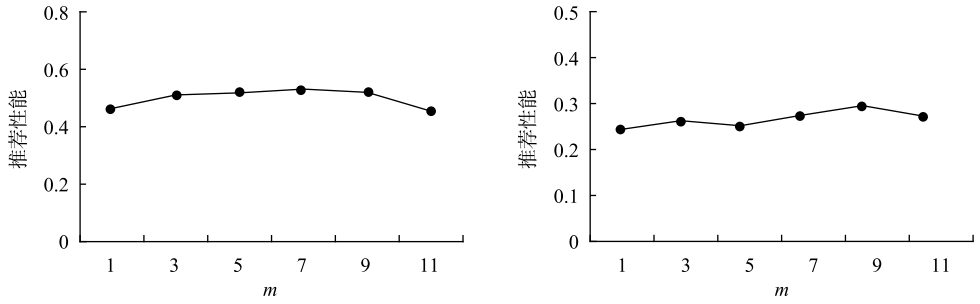
$$\text{MRR@}K=\frac{1}{m}\sum_{v_t\in G}\frac{1}{\text{Rank}(v_t)} \tag{23}$$

本文基于两个不同的角度测试推荐效果，一个是在全部测试用例上衡量推荐的 Recall 和 MRR 值，另一个则是在用户观看新主播的测试用例上衡量上述指标。这里，用户观看新主播的定义是，该用户在测试集中观看了其在训练集中未曾观看过的主播。这可以代表用户的探索行为，在推荐领域中，如何发掘用户新的喜好是个重要的问题。

5.3 动态特征和实时特征的参数设置

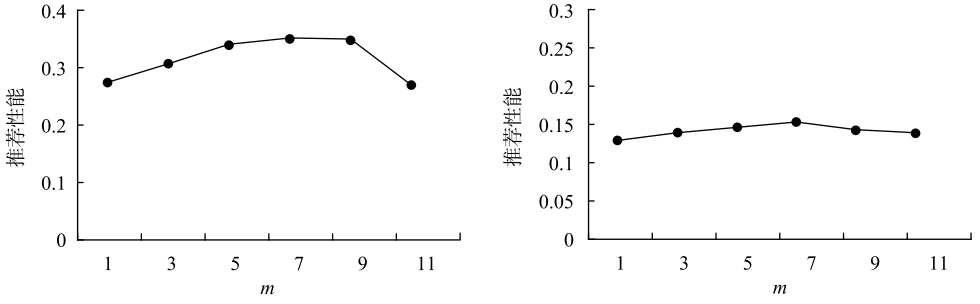
本文所提模型在动态特征和实时特征部分分别选取当前用户的最近 m 次观看的主播和与其最相似的用户当前观看的 n 个主播的嵌入向量作为输入。本节讨论这两个参数的设置。

对于 m 的设置，本文比较了其不同取值对推荐效果的影响。以 Recall@10 和 MRR@10 为例，图 4 给出了 m=1,3,5,7,9,11 六种不同取值情况下在全部测试用例上的推荐性能，图 5 则给出了 m 在六种不同取值情况下在新主播测试用例上的推荐性能。



（a）m 取不同值情况下的 Recall@10　　（b）m 取不同值情况下的 MRR@10

图 4　m 的不同取值在全部测试用例上的推荐性能变化

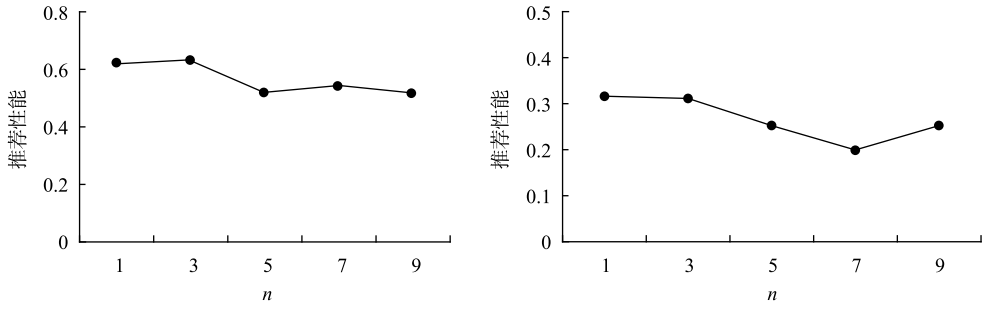


（a）m 取不同值情况下的 Recall@10　　（b）m 取不同值情况下的 MRR@10

图 5　m 的不同取值在新主播测试用例上的推荐性能变化

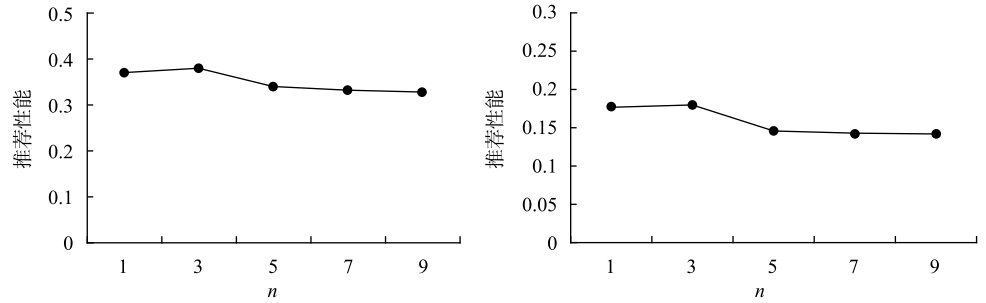
从图 4 和图 5 可以看出，当取到的最近观看次数 m 不大于 7 时，随着 m 的增大，推荐性能有上升趋势，这是因为在一定时间内能够利用的观看记录更多，可以让系统更好地获得用户的动态特征。但当 m 更大时，随着 m 的增大，推荐性能有下降趋势，这是因为过早的观看记录已经无法反映用户目前的兴趣特征，反而会为模型带来噪声。从图中可知当 m 取 5 到 9 之间时模型的性能较好且相对稳定，我们在后续实验中设置 m=5，与其他模型进行对比。

对于 n 的取值，本文比较了其取不同值对推荐效果的影响。以 Recall@10 和 MRR@10 为例，图 6 给出了 n=1,3,5,7,9 五种不同取值情况下在全部测试用例上的推荐性能，图 7 则给出了 n 在五种不同取值情况下在新主播测试用例上的推荐性能。



（a）n 取不同值情况下的 Recall@10　　（b）n 取不同值情况下的 MRR@10

图 6　n 的不同取值在全部测试用例上的推荐性能变化



（a）n 取不同值情况下的 Recall@10　　（b）n 取不同值情况下的 MRR@10

图 7　n 的不同取值在新主播测试用例上的推荐性能变化

从图 6 和图 7 不难看出，随着 n 取值变大，推荐性能有下降趋势，这也是比较容易理解的，当 n 取值较大时，可能会引入不相似用户的观看偏好，因而为模型的输入带来噪声，降低模型的准确度。从图中看，n 取 3 相对有比较好的表现，n 取 5 之后表现相对稳定，因而在后续的实验中设置 n=5，与其他模型进行对比。

5.4　与其他模型对比

本节将对比本文提出的 SDRIV 模型与其他已有模型的推荐效果，包括了经典的推荐模型、基于会话的推荐模型以及基于强化学习的模型。衡量指标为 Recall 和 MRR 值，为了全面地比较推荐效果，推荐列表包含的主播数量 K 选取了 1，5，10，20 四个值进行比较。

下面为本文对比的算法的简介。

POPULAR：根据训练集中被观看的次数对主播进行排序，推荐训练集中最受欢迎的 K 个主播。

U-POP：对于每个用户，推荐其在训练集中观看次数最多的K个主播。该算法在推荐新主播的测试用例下不适用，因为其推荐的都是用户已经观看过的主播。

ITEM-KNN[33]：该方法基于物品的相似程度进行推荐，物品的相似程度根据 cosine 相似度进行计算。

BPR[28]：该方法应用矩阵分解进行推荐。区别于传统矩阵分解，BPR 利用随机梯度下降对一个基于物品偏序关系的目标函数进行优化。

GRU4REC[6]：采用循环神经网络模型来建模会话中的用户点击序列。

SRGNN[7]：采用图神经网络，将全局偏好和当前会话偏好结合，对下一物品进行预测。

SHAN[8]：采用基于层级的注意力网络，结合用户长短期的偏好进行预测，将用户长期的偏好的变化加入考量。

SRLDTR[17]：采用结合监督学习和深度强化学习的方法，用循环神经网络处理 POMDP 问题。该算法在原文中用于药品推荐，由于该算法也利用了强化学习结合有监督学习的方法，因此，本文将其应用到直播推荐中，与本文方法进行对比。

表 3 给出了全部测试用例下，各模型在推荐列表长度 K=1,5,10,20 下的 Recall 值和 MRR 值。

表 3　模型效果对比（全部测试用例）

模型	Top 1	Top 5	Top 10	Top 20
POPULAR	0.044/0.044	0.132/0.067	0.232/0.079	0.310/0.084
U-POP	**0.149/0.149**	0.363/0.223	0.479/0.238	0.576/0.245
BPR	0.094/0.094	0.359/0.192	**0.522**/0.213	0.625/0.220
ITEM-KNN	0.126/0.126	0.306/0.190	0.395/0.202	0.506/0.209
SRGNN	0.127/0.127	0.358/0.214	0.462/0.228	0.532/0.233
SHAN	0.106/0.106	0.273/0.161	0.385/0.176	0.495/0.184
GRU4REC	0.146/0.146	0.296/0.202	0.375/0.212	0.463/0.218
SRLDTR	0.127/0.127	0.262/0.177	0.367/0.190	0.642/0.210
SDRIV	0.148/0.148	**0.387/0.236**	0.520/**0.253**	**0.644/0.262**

表 4 给出了新主播测试用例下，各模型在 K=1,5,10,20 下的 Recall 值和 MRR 值。

表 4　模型效果对比（新主播测试用例）

模型	Top1	Top5	Top10	Top20
POPULAR	0.002/0.002	0.026/0.008	0.053/0.011	0.110/0.015
BPR	0.006/0.006	0.105/0.039	0.324/0.066	0.459/0.076
ITEM-KNN	0.023/0.023	0.056/0.035	0.077/0.038	0.139/0.042
SRGNN	0.023/0.023	0.097/0.049	0.143/0.054	0.204/0.059
SHAN	0/0	0.010/0.003	0.020/0.004	0.141/0.011
GRU4REC	0.062/0.062	0.137/0.091	0.176/0.096	0.223/0.099
SRLDTR	0.048/0.048	0.186/0.090	0.231/0.097	0.457/0.120
SDRIV	**0.087/0.087**	**0.207/0.128**	**0.341/0.146**	**0.526/0.159**

在全部测试用例中的实验结果中，SDRIV 算法相较其他算法有着一定的提升。从 U-POP 的结果可以看出，仅推荐用户历史中喜好的主播就可以取得相对较好的效果。并且在排序上，由于 U-POP 是按照用户对主播的观看次数进行排序，故而其 MRR 值也很高。基于会话的推荐中，采用图神经网络的 SRGNN 也取得了较好的效果，不过相比于本文的 SDRIV 算法有一定的差距。对于 Recall 值所反映出的算法的召回能力，BPR 算法在 Top1 中推荐效果较差，而 Top5、Top10 和 Top20 中的结果均较好，与 SDRIV 算法持平。但对于 MRR 值所反映出的算法的排序能力，SDRIV 算法相比于 BPR 算法有着较大的提升。SDRIV 算法合理应对了直播推荐的特点，有效建模用户表示，并采用监督学习确保了推荐质量，在 Top1 的 Recall 和 MRR 以及 Top10 的 Recall 上与表现最好的结果非常接近，而在其他情况下均表现最优。

在对新主播的推荐上，可以看出，本文所提模型优于其他所有模型，其推荐策略的探索机制为用户带来对新主播的推荐，且基于排序的有监督学习的融入也使得模型保持了较高的推荐准确率。

5.5 用户状态建模的性能测试

4.2 节介绍了本文提出的用户状态表示模型。用户状态表示模型由三部分构成，即用户的静态特征、动态特征和实时特征。本小节将分模块验证各个部分的效果，即模型其他部分不变，但是在用户表示部分仅采用一个或两个特征作为输入。

表 5 给出推荐列表长度 K=20 时的实验结果。在只采用单独一个特征作为用户状态建模的情况下，静态特征忽略了其动态性和实时性，使得其在预测全部测试用例或观看新主播测试用例上均表现较差。在全部测试用例情况下，动态特征的 Recall 值最高，实时特征的 MRR 值最高。在新主播测试集上，实时特征性能最优。观看新主播的测试结果说明，由于在该测试用例下，用户之前未曾观看过该主播，故而其静态特征和历史特征均不如实时特征对观看新主播的预测和排序准确。

表 5 用户状态建模的实验结果

用户状态	全部测试用例		观看新主播测试用例	
	Recall@20	MRR@20	Recall@20	MRR@20
静态特征	0.623	0.214	0.508	0.089
动态特征	0.638	0.224	0.511	0.093
实时特征	0.624	0.246	0.522	0.147
静态+动态特征	0.641	0.218	0.518	0.090
静态+实时特征	**0.645**	0.252	0.512	0.151
动态+实时特征	0.636	0.253	0.521	0.142
SDRIV	0.644	**0.262**	**0.526**	**0.159**

根据融合了两个特征的建模用户表示的结果，采用实时特征可以提高 MRR 值，即将用户观看的主播排在前列。包含用户静态特征会使全部测试用例的 Recall 值有一定提高，说明用户的静态特征也是预测用户下一个观看的主播所必不可少的一部分。动态特征模块则更好地把控了用户近期的偏好，相对来说要比静态特征更为敏感，更能拟合用户当前喜好。将三种用户特征结合起来共同建模用户状态，模型在全部测试用例的 MRR 值和观看新主播测试用例的 Recall 和 MRR 值上总体表现最优。此部分实验在推荐列表长度 K=1,5,10 时具有相同结论。

5.6 基于排序的有监督学习

为了测试本文提出的将基于排序的有监督学习与强化学习相结合的有效性，将 SDRIV 与其简化版本 DRIV 进行比较。DRIV 为本文提出的算法中仅基于强化学习的推荐策略。具体地说，在 Actor 模型中，令强化学习策略的权重占比$\epsilon=1$。这样，模型的学习将不依赖于基于排序的有监督学习。其结果对比显示于表 6 中。从中可以看到，SDRIV 算法由于基于排序的监督学习的融入，其性能无论是在全部测试用例上还是在新主播测试用例上，Recall 和 MRR 均有提升，说明了引入基于排序的有监督学习的有效性。此部分实验在推荐列表长度 K=1,5,10 时具有相同结论。

表 6 基于排序的有监督学习性能对比

模型	全部测试用例		观看新主播测试用例	
	Recall@20	MRR@20	Recall@20	MRR@20
DRIV	0.626	0.248	0.419	0.101
SDRIV	**0.644**	**0.262**	**0.526**	**0.159**

6 结论与展望

本文采用了深度强化学习的方式建模直播推荐问题。强化学习可以有效地建模用户的动态性，权衡用户长短期收益，学习最优推荐策略。本文根据直播推荐场景，在深度确定性策略梯度模型基础上，对模型进行改进并提出 SDRIV 模型。SDRIV 模型融合了基于排序的有监督学习策略，使得模型在探索用户真实偏好过程中保证推荐质量。同时，本文提出了融合用户静态、动态和实时特征的用户状态表示模型，有效建模用户当前状态。实验结果表明，SDRIV 算法相较其他方法在直播推荐任务中取得了更好的推荐效果。

未来，对直播推荐的研究可以尝试对视频特征的提炼。主播、用户和视频存在着三元关系，本文在推荐系统的设计主要基于用户和主播的角度进行建模。提取视频特征相对来说成本较高，不过视频的特征更能有效地反映直播内容的实时状态。如何有效融入视频特征是可以进一步研究的方向。

参 考 文 献

[1] 艾媒咨询. 2020-2021 中国在线直播行业年度研究报告[EB/OL]. https://www.iimedia.cn/c460/77452.html，2021-03-15.

[2] Swaminathan A，Krishnamurthy A，Agarwal A，et al. Off-policy evaluation for slate recommendation[EB/OL]. https://arxiv.org/abs/1605.04812，2017-11-06.

[3] 冷亚军，陆青，梁昌勇. 协同过滤推荐技术综述[J]. 模式识别与人工智能，2014，27（8）：720-734.

[4] 刘建国，周涛，汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展，2009，19（1）：1-15.

[5] He R，McAuley J. Fusing similarity models with markov chains for sparse sequential recommendation[C]. 2016 IEEE 16th International Conference on Data Mining. Barcelona，Spain，2016：191-200.

[6] Hidasi B，Karatzoglou A，Baltrunas L，et al. Session-based recommendations with recurrent neural networks[C]. International Conference on Learning Representations. Puerto Rico，2016.

[7] Wu S，Tang Y Y，Zhu Y Q，et al. Session-based recommendation with graph neural networks[EB/OL]. https://arxiv.org/abs/1811. 00855，2019-01-24.

[8] Ying H C，Zhuang F Z，Zhang F Z，et al. Sequential recommender system based on hierarchical attention networks[C]. The 27th International Joint Conference on Artificial Intelligence. Stockholm，Sweden，2018：3926-3932.

[9] Silver D，Google DeepMind. Tutorial：deep reinforcement learning[EB/OL]. https://icml.cc/2016/tutorials/deep_rl_tutorial.pdf，2016.

[10] Kober J，Bagnell J A，Peters J. Reinforcement learning in robotics：a survey[J]. The International Journal of Robotics Research，2013，32（11）：1238-1274.

[11] Levine S，Finn C，Darrell T，et al. End-to-end training of deep visuomotor policies[J]. Journal of Machine Learning Research，2015，17（1）：1334-1373.

[12] Zhao X Y，Zhang L，Xia L，et al. Deep reinforcement learning for list-wise recommendations[EB/OL]. https://arxiv.org/abs/1801.00209，2017-12-30.

[13] Zhao X Y，Xia L，Zhang L，et al. Deep reinforcement learning for page-wise recommendations[C]. The 12th ACM Conference on Recommender Systems. Vancouver，Canada，2018：95-103.

[14] Zhao X Y，Zhang L，Ding Z Y，et al. Recommendations with negative feedback via pairwise deep reinforcement learning[C]. The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London，United Kingdom，2018：1040-1048.

[15] Volodymyr M，Koray K，David S，et al. Human-level control through deep reinforcement learning[J]. Nature，2015，518（7540）：529-533.

[16] Zheng G J，Zhang F Z，Zheng Z H，et al. DRN：a deep reinforcement learning framework for news recommendation[C]. The 2018 World Wide Web Conference. Lyon，France，2018：167-176.

[17] Wang L，Zhang W，He X F，et al. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation[C]. The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London，United Kingdom，2018：2447-2456.

[18] Liu F，Tang R M，Li X T，et al. Deep reinforcement learning based recommendation with explicit user-item interactions modeling[EB/OL]. https://arxiv.org/abs/1810.12027，2019-10-29.

[19] Chen S Y，Yu Y，Da Q，et al. Stabilizing reinforcement learning in dynamic environment with application to online recommendation[C]. The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London，United Kingdom，2018：1187-1196.

[20] Davidson J，Liebald B，Liu J N，et al. The YouTube video recommendation system[C]. The 4th ACM Conference on Recommender Systems. Barcelona，Spain，2010：293-296.

[21] Covington P，Adams J，Sargin E. Deep neural networks for YouTube recommendations[C]. The 10th ACM Conference on Recommender Systems. Boston，Massachusetts，USA，2016：191-198.

[22] Mei T，Yang B，Hua X S，et al. VideoReach：an online video recommendation system[C]. SIGIR 2007：Proceedings of the 30th Annual International ACM SIGIR Conference on Research & Development in Information Retrieval. Amsterdam，Netherland，2007：767-768.

[23] Deldjoo Y，Elahi M，Cremonesi P，et al. Content-based video recommendation system based on stylistic visual features[J]. Journal on Data Semantics，2016，5（2）：99-113.

[24] Yang T W，Shih W Y，Huang J L，et al. A hybrid preference-aware recommendation algorithm for live streaming channels[C]. Conference on Technologies & Applications of Artificial Intelligence. Taipei，China，2013.

[25] Liu Y W，Lin C Y，Huang J L. Live streaming channel recommendation using HITS algorithm[C]. IEEE International Conference on Consumer Electronics-Taiwan. Taipei，China，2015.

[26] Lillicrap T P，Hunt J J，Pritzel A，et al. Continuous control with deep reinforcement learning[EB/OL]. https://arxiv.org/abs/1509.02971，2015-09-09.

[27] Silver D，Lever G，Heess N，et al. Deterministic policy gradient algorithms[C]. International Conference on Machine Learning. Beijing，China，2014.

[28] Rendle S，Freudenthaler C，Gantner Z，et al. BPR：bayesian personalized ranking from implicit feedback[C]. Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. Montreal，Canada，2009：452-461.

[29] Wu C Y，Ahmed A，Beutel A，et al. Recurrent recommender networks[C]. Tenth ACM International Conference on Web Search & Data Mining. Cambridge，United Kingdom，2017：495-503.

[30] Kumar S，Zhang X K，Leskovec J. Predicting dynamic embedding trajectory in temporal interaction networks[C]. The 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Anchorage，Alaska，USA，2019：1269-1278.

[31] Bahdanau D，Cho K，Bengio Y. Neural machine translation by jointly learning to align and translate[J]. Computer Science，2014.

[32] Mnih V，Kavukcuoglu K，Silver D，et al. Human-level control through deep reinforcement learning[J]. Nature，2015，518（7540）：529-533.

[33] Sarwar B，Karypis G，Konstan J，et al. Item-based collaborative filtering recommendation algorithms[C]. The 10th International World Wide Web Conference. Hong Kong，China，2001：285-295.

A Supervised Deep Reinforcement Learning Based Live Streaming Recommendation Method

WANG Xiao，LIU Hongyan，CHE Shangkun

（School of Economics and Management，Tsinghua University，Beijing 100084，China）

Abstract With the rapid rise of live streaming industry，recommending interested live streaming to users is the key to enhance user experience. Live streaming is dynamic，as its content changes dynamically and user's preference changes rapidly. Existing recommendation algorithms fail to model the dynamic nature of live streaming. In this paper，we propose a novel supervised deep reinforcement learning based recommendation model，SDRIV. In this model，we model user's state from three angles to reflect the dynamics. Meanwhile，we introduce a ranking based supervised learning strategy to the deep reinforcement learning model，which can make the model conduct exploration while guaranteeing recommendation accuracy and user experience. The experimental results on real-world dataset demonstrate its advantage over benchmark models.

Keywords recommendation system，deep reinforcement learning，live streaming，supervised learning

作者简介

王潇（1995—），男，获清华大学经济管理学院管理学硕士学位，研究方向为强化学习、推荐系统等，E-mail：wx950724@126.com。

刘红岩（1968—），女，博士，清华大学经济管理学院教授、博士生导师，主要研究方向为机器学习、商务智能、社会计算、个性化推荐系统、计算机视觉、医疗和金融数据分析等。在国际学术期刊和国内外学术会议上发表论文百余篇，包括国际一流学术期刊 ISR、MIS Quarterly、INFORMS JOC、TOIS、TODS、TKDE 以及一流国际学术会议 VLDB、IEEE ICDE、ACM SIGKDD、IEEE ICDM、SDM、CIKM、ICIS 等。获得 11 项国家发明专利授权。主持和参与多项国家自然科学基金面

上、重大和重点项目及国家社会科学基金重大项目，E-mail：liuhy@sem.tsinghua.edu.cn。

车尚锟（1997—），男，清华大学经济管理学院 2019 级博士研究生，研究方向为个性化推荐系统，E-mail：csk19@mails.tsinghua.edu.cn。