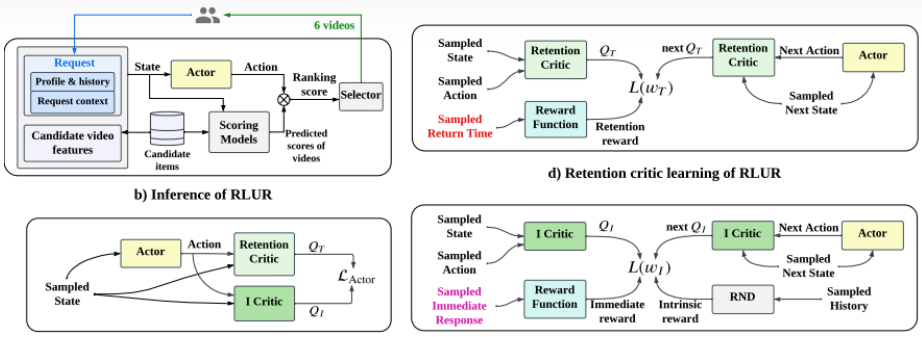
知乎



快手2023：利用强化学习提升短视频推荐系统，攻克用户留存难题

SmartMindAI
专注搜索、广告、推荐、大模型和人工智能最新技术，欢迎关注我

已关注

11 人赞同了该文章

Introduction

短视频应用如抖音、YouTube Shorts和快手吸引了数十亿用户，并通过推荐有趣的内容吸引他们。学术界和业界对短视频推荐越来越感兴趣。推荐者通过多轮交互与系统交换信息，包括显性和隐性的反馈（如观看时间、点赞、关注、评论等），然后系统返回每轮请求的一个短视频列表。最终目标是提高留存率，即用户再次访问该系统的比率，通常称为第二天留存率。这直接关系到DAU（短视频应用的核心价值）。
当前推荐系统通常使用点对点模型或列表模型来预测用户对项目的即时奖励或项目组合的奖励。然而，保留奖励是由用户多次与系统交互产生的长期反馈，类似于围棋游戏中的中间反馈。由于保留奖励与中间反馈之间的关系复杂且难以分解，因此现有模型无法优化保留奖励。我们采用强化学习方法优化用户留存。将视频推荐系统中的用户留存问题建模为无限期的基于请求的马尔科夫决策过程，代理是推荐器，用户是环境。每次用户请求，代理使用评分模型的预测反馈分数生成排名权重，并以此组合出得分最高的视频。用户会立即给出反馈，如观看时间和互动情况。会话结束后，下一个会话从用户重新打开应用程序开始。目标是最小化累计返回时间（定义为上一次请求到下一次请求之间的时间间隔）。这种方法与之前最大化累积即时反馈的强化学习不同，它是短视屏推荐系统中直接优化用户留存的方法。
1）不确定性：系统外的不确定因素影响保留率，例如社交事件噪音。
2）偏差：保留率随时间（平日/周末）和用户活动水平变化，高峰期活跃用户保留率高。
3）长延迟时间：保留奖励返回时间长，训练策略不稳定，需要处理分布变化。

解决方案：
- 提出归一化技术预测返回时间，减小不确定性。
- 对不同用户组训练不同策略，防止高活跃用户主导学习。
- 提出软正则化方法平衡样本效率和稳定性。
-使用即时反馈作为启发式回报和内在动机方法优化保留。

本文将短视频推荐问题建模为无限时长的有请求基础的MDP，旨在减少累计返回时间；同时提出解决用户留存率不确定性和偏见的方法；实验证明RLUR在离线和直播环境下显著提高用户留存率；最后，RLUR算法已在Kuaishou应用中发布，证明其持续提高用户留存率和DAU。

Problem Definition

状态 s_{it} 由用户档案、行为历史u_{i_t}、请求上下文和候选视频特征组成。
行动 动作是$[0, C]^n$中的连续向量，其中n是评分模型的数量。
排名函数 我们使用线性排名函数，即$f(a_{i_t}, x_j) = \sum_{k=1}^{n} a_{i_t k} x_{jk}$。
即时奖励 即时奖励$I(s_{i_t}, a_{i_t})$包括观看时间和互动，定义为该请求的观看时间和交互次数之和。
返回时间 返回时间$T(s_i)$是会话s_i的最后一个请求与会话s_{i+1}的第一个请求之间的时间间隔。

如果一个项目有超过 d 次交互，它会直接分裂并单独形成一个集群。

动作a_{i_t}。我们的目标是最小化累计回报时间$\sum_{i=1}^{\infty}\gamma^{i-1}T(s_i)$，其中$\gamma$（0\< γ \<1）为折扣因子。

Method

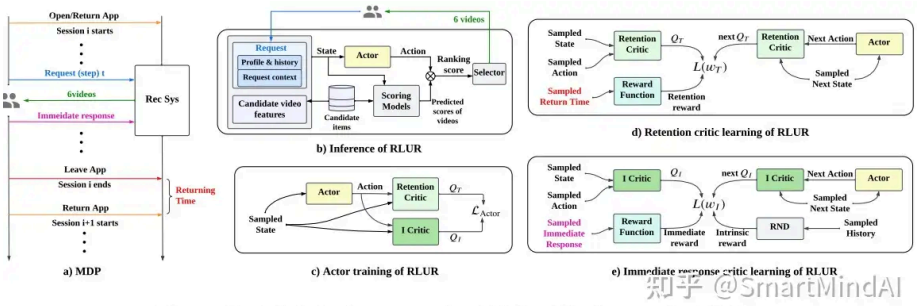


Figure 1: The infinite horizon request-based MDP and the framework of RLUR

Retention Critic Learning

我们使用参数化⁺的评估函数
$Q_T(s_{i_t}, a_{i_t}|w_T)$
来估计从当前状态 s_{i_t} 和动作 a_{i_t} 的累积返回时间。我们通过深度确定性策略梯度⁺方法来估计累积返回时间的奖励。学习过程可以用图表示为：$L(w_T)$。

$$\sum_{s_{i_t},a_{i_t}\in D}(Q_T(s_{i_t},a_{i_t}|w_T)-(r(s_{i_t},a_{i_t})+\gamma_{i_t}Q_T(s_{i_{t+1}},\pi(s_{i_{t+1}}|\theta)|w_T)))^2,$$

当$\gamma_{i_t}=1$时，对于非最后一个请求的样本，折扣因子设置为1；对于最后一个请求，设置为γ。D是数据集。非终端样本的折扣因子设置为1，以防止通过指数衰减⁺机制使返回时间奖励消失。如果将折扣因子设置为小于1的数字，则未来会话中返回时间的重要性将非常小。优化损失等价于估计

$$\sum_{i=1}^{\infty}\gamma^{i-1}T(s_i)。$$

$$\begin{cases}(Q_T^{w_1}(s_{i_t},a_{i_t})-(r(s_{i_t},a_{i_t})+Q_T^{w_1}(s_{i_{t+1}},\pi(s_{i_{t+1}}|\theta))))^2 & \text{if } i_t \text{is the last request of session i}\\ -(n+1)/2 & \text{Otherwise}\end{cases}$$

Methods for Delayed Reward

使用随机网络蒸馏（RND）方法可以有效地驱动策略探索新状态，同时保持即时奖励与保留奖励的独立性。为了解决即时奖励与保留奖励之间的干扰，我们还学习了一个独立的即时奖励批评器。

$$\sum_{s_{i_t},a_{i_t}\in D}(Q_I(s_{i_t},a_{i_t}|w_I)-(I(s_{i_t},a_{i_t})+||E(u_{it}|w_e)-E(u_{it}|w_e^*)||_2^2 + \gamma Q_I(s_{i_{t+1}},\pi(s_{i_{t+1}}|\theta)|w_I)))^2.$$

Uncertainty

利用归一化技术，将真实返回时间与预测返回时间的比例作为归一化⁺的保留奖励。对于预测返回时间，使用$\beta\%$分位数⁺T_β来确定样本x是否为正。如果返回时间小于T_β，则样本x的标签为正，因为较短的时间意味着更好。$T'(x)$的损失函数⁺为

$$y*logT'(x)+(1-y)*log(1-T'(x))$$

其中$T'(x)$预测返回时间小于T_β的概率。通过Markov不等式得到期望返回时间的下界，

$$ET(x)\geq P(T(x)\geq T_\beta)*T_\beta\sim(1-T'(x))*T_\beta$$

最后，我们得到了归一化的保留奖励，

$$r(s_{i_t},a_{i_t})=clip\{0,\frac{T(s_i)}{(1-T'(x))*T_\beta},\alpha\}。$$

为了在短时间内提高回报率并最大化即时奖励，对于高活跃用户和低活跃用户分别采用了不同的策略

$\pi(\cdot|\theta_{\text{high}})$
和 $\pi(\cdot|\theta_{\text{low}})$。

为此，使用了一种加权损失函数来评估策略

$\pi(\cdot|\theta_{\text{high}})$

的效果，该损失包括即时批评家+和保持批评家的影响。

$$L(\theta_{\text{high}}) = \lambda_T Q_T(s_{i_t}, \pi(s_{i_t}|\theta_{\text{high}})|w_T) - \lambda_I Q_I(s_{i_t}, \pi(s_{i_t}|\theta_{\text{high}})|w_I),$$

给定正权重λ_T, λ_I，其他策略的学习损失相似。Actor的学习结果见图(c)。

Tackling the Unstable Training Problem

我们提出了一种软正则化的改进方法，以提高在线学习的稳定性和样本效率。我们的方法基于行为克隆损失的正则化，但避免了其可能带来的问题。具体的行动损失定义如下：

$$\exp(\max\{\lambda * (log(p(a_{i_t}|s_{i_t})) - log(p_b(a_{i_t}|s_{i_t}))), 0\})L(\theta)$$

$$\exp(\max\{\lambda * (log(p(a_{i_t}|s_{i_t})) - log(p_b(a_{i_t}|s_{i_t}))), 0\})L(\theta_{\text{high}}),$$

结合上述技术，我们提出了用户保留的强化学习（RLUR）算法，框架如图所示。

Offline Experiments

我们验证了RLUR在公共短视频推荐数据集《快赚》上的有效性。我们使用模拟器构建了三个部分：预测用户立即反馈模块、预测用户是否会离开会话的模块以及预测用户在第k天返回应用的概率模块。我们将K设为10。我们将RLUR与其他常用参数搜索黑盒优化方法+（CEM）和最先进强化学习方法（TD3）进行比较。评估各项指标，如所有用户会话中返回平均天数（返回时间）和第一天平均保留率（用户保留率）。通过训练直到收敛，报告最后50个episode+的平均性能。

Table 1: Offline Results

Algorithm	Returning time↓	User retention↑
CEM	2.036	0.587
TD3	2.009	0.592
RLUR (naive, $\gamma = 0$)	2.001	0.596
RLUR (naive, $\gamma = 0.9$)	1.961	0.601
RLUR	**1.892**	**0.618**

结果表明，TD3在各项指标上均优于CEM。而RLUR则显著优于TD3和CEM。我们还研究了一种仅关注学习返回时间部分的RLUR变体（RLUR（Naive,γ=0.9）），并在各项指标上均优于RLUR（Naive,γ=0）。

Live Experiments

(a) 用户发出请求后，其状态会被发送给行为者，然后返回平均值μ和方差σ。接着从高斯分布+ N(μ, σ)中抽取一个动作。随后排名函数会计算动作与每部视频预测评分的线性乘积，并向用户推荐前6个评分最高的视频。

知乎

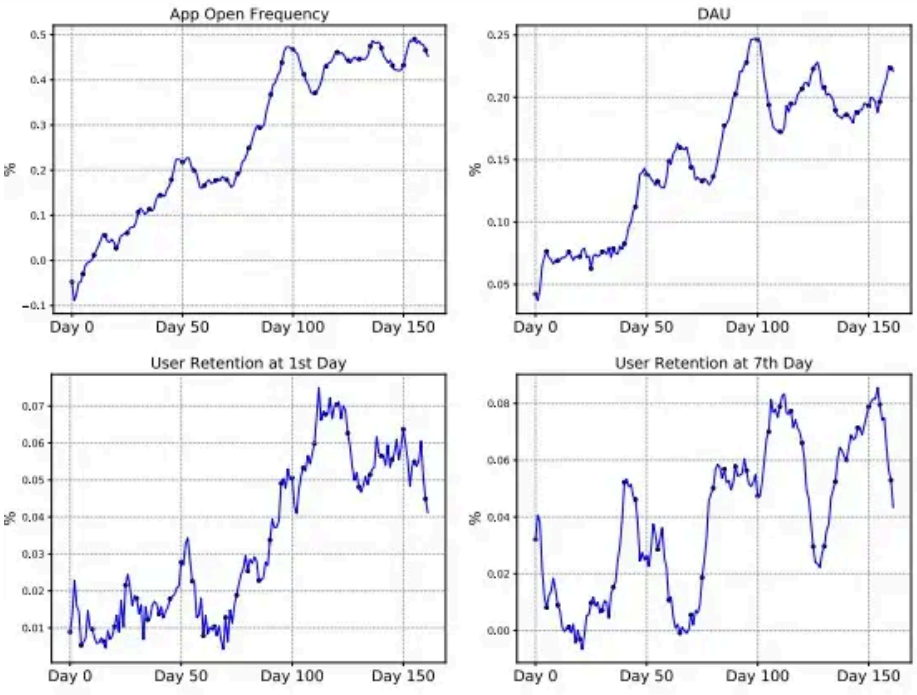


Figure 2: Live performance gap of each day.

状态 状态是用户档案、行为历史（用户统计信息以及前3个请求中用户的视频ID和相应的反馈）、请求上下文和候选视频特征的向量。用户档案包括年龄、性别和位置。用户统计信息包括各种反馈的统计信息。至于请求上下文，我们使用时间（小时和分钟）和此请求的会话深度。候选视频特征包括评分模型预测的统计数据和通过率。

行动 我们选择一个8维连续向量，在$[0,4]^8$范围内，策略输出预测主要反馈（观看时间、短视图、长视图、喜欢、关注、转发、评论和个人页面进入）的8个评分模型的参数。

即时奖励 每个请求的即时奖励$I(s_{it}, a_{it})$被设计为6个视频的观看时间（以秒为单位）和互动（包括喜欢、关注、评论和分享）之和。

参数设置：折扣因子为0.95，actor loss权重为1.0,1.0，保留模型为第60百分位，$\alpha = 3$，正则化系数$^+\lambda = 1.5$。 图2显示了RLUR和CEM的比较结果，x轴表示部署后天数，y轴表示性能差距的百分比。app open频率从Day 0到Day 80持续增加，然后在Day 80到Day 100急剧增加，最后收敛到0.450%。DAU和留存率在Day 0到Day 80期间增长缓慢，从Day 80到Day 100急剧增长，最后分别收敛到0.2%和0.053%。需要注意的是，0.01%的用户留存率和0.1%的DAU改善在短视频平台上具有统计学意义。

Conclusion

本文研究了优化短视频推荐系统用户留存的问题。将此问题形式化为一个基于请求的马尔可夫决策过程$^+$，该过程在无限时域内运行。目标是学习使累积回访时间最小化的策略。提出了一种新的解决留存挑战的技术，即RLUR算法，并介绍了其工作原理。离线和实时实验验证了RLUR的有效性，并证明了其在亿级别短视频系统中的部署可以显著提高用户留存率和每日活跃用户数（DAU），并且具有良好的一致性。

关注我，追踪最新技术不迷路

发布于 2023-12-01 10:28 · IP 属地北京

强化学习 (Reinforcement Learning)　快手　短视频