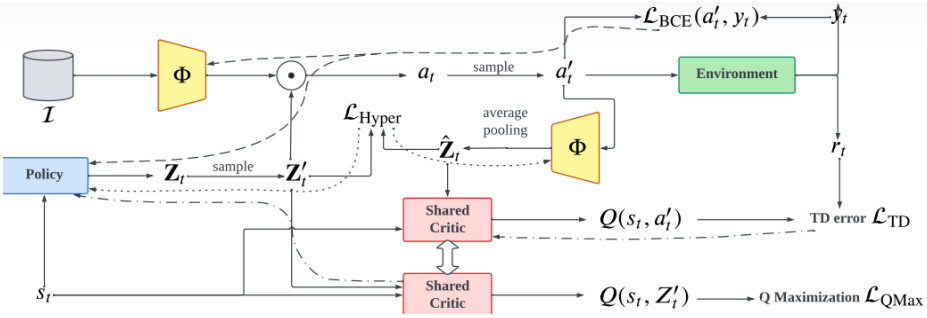


快手-2023：推荐系统中的强化学习-深入探索潜在行为空间的秘密

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

Introduction

在推荐系统中，额外的隐含推断步骤给RL带来了新的挑战：两种不同动作空间之间的不一致性（高效一端到端训练中的高效超动作与实际效果的动作，提出的最佳离散效应动作的潜在表示可能不同于拟合的超动作），不确定的推理准确性，以及额外的探索阶段及是否应在潜在空间上探索超动作或离散效应动作空间，增加了学习的不稳定性和不确定性，需要一个能调节两种动作空间并稳定RL过程的方案。

本研究提出了一种名为 Hyper-Actor Critic (HAC) 的通用学习框架，它包括四个部分：用户状态和超动作生成器；评价函数，将超动作映射到效果动作（推荐列表）的评分；批评网络，评估超动作空间和效果动作空间；反向映射模块，根据效果行动推测超动作。 在训练过程中，采用backbone actor-critic学习方法，加入对齐模块以保证两个操作空间的一致性，并添加了监督模块以提高稳定性和有效性。实验结果显示，该框架适用于多种推荐任务解决方案，如DDPG和在线/离线监督学习（SL）。

- 提出一种基于潜在超行动的高效RL框架用于大型物品池推荐。
- 我们使用公共数据集构建在线模拟器并通过实验验证了其优于标准RL和SL解决方案的性能。
- 一致性协调有助于提高效率和精度。

Method

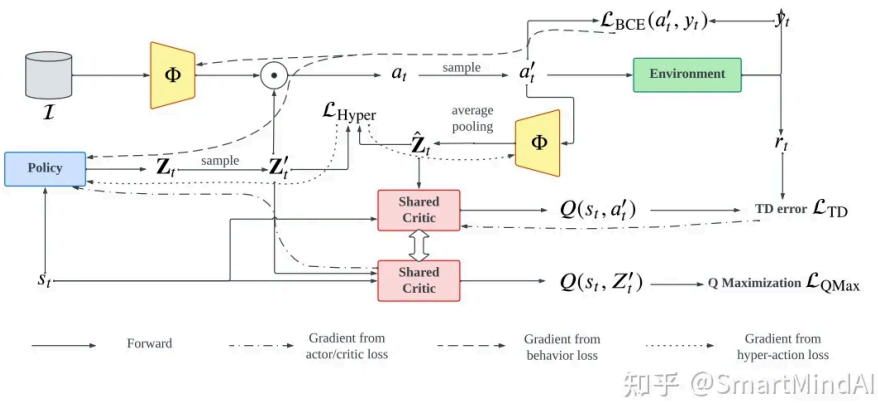


Figure 2: Hyper-Actor Critic (HAC) learning framework. ⊙ represents the scoring function that selects items from $\mathcal{I}$,

Problem Formulation

考虑基于会话的推荐场景，系统需处理包含N个物品的集合$\mathcal{I}$及用户的静态特征$\boldsymbol{u}$和会话历史$\boldsymbol{x}_{1:t}$。目标是在每次交互中选择增加累积奖励最多的项。我们将推荐问题建模为修改后的马尔可夫决策过程（MDP），如图所示。在这个过程中，潜在的动作空间被强调。

- $\mathcal{S}$：用户状态的连续表示空间。

知乎

- $\mathcal{R}(s,a)=r(s,a)$代表了在状态s采取行动a时的单步奖励。

RS中的隐式状态转换模型为用户提供了一个概率估计，即在采取特定动作后从一个状态移动到另一个状态的可能性。这个模型被融入到用户状态编码器中，并通过序列模型来建模。每次用户与系统的交互中，推荐策略都会生成一系列可能的超行动，然后选择最有可能的结果。用户的反馈以及更新后的用户状态会被送回系统，并且奖励函数被视为已知的。目标是找到能够使整个会话中总回报最大的最优推荐策略。

$$\mathbb{E}_{\tau\sim\pi}[\mathcal{R}(\tau)]=\mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{|\tau|}\gamma^t r(s_t,a_t)\right]$$

τ代表了轨迹，其中s_0, a_0, r_0,...表示一系列的轨迹点，γ为折扣因子，衡量未来奖励的价值。

Overall Framework

图1所示的框架被称为"超演员-批评家"学习方法。推荐策略可以分解为超行动网络，用于生成向量化超行动，以及排名评分者，用于根据超行动选择最终的推荐列表。此外，我们提出共享批评家网络，以评估超行动或最终效果动作（即推荐列表）。DDPG是基础，但我们解决了使用不同动作空间进行演员学习和批评家学习的问题。具体来说，我们优化效果动作以确保准确评估，使用超行动优化演员，实现端到端有效训练和探索。我们学习了反池化模块，使用项核函数从效果动作推断出超行动，并确保两个动作空间的评价共享同一批评家，从而从效果动作中传递知识到超行动。最后，为了稳定学习过程，我们包括了即时用户响应的监督作用于效果动作。

User State and Hyper-Actor

在RS中，用户可见的信息包括$\boldsymbol{x}_{1:t}$和$\boldsymbol{u}$。这些信息可以被编码并用来预测动态用户状态。

$$\boldsymbol{s}_t=\mathrm{StateEnc}(\Phi(x_1),\ldots,\Phi(x_t),\boldsymbol{u})$$

采用密集嵌入的方式在内核空间中表示项目，并利用用户函数将用户特征和历史记录转换为同一内核空间的表示。随后，作者使用SASRec的状态编码器作为基础，对用户特征和历史记录进行编码，并通过多层感知机（MLP）模块生成矢量化表示（即超动作）。

$$\boldsymbol{Z}_t=\mathrm{MLP}(\boldsymbol{s}_t)$$

假设分布服从标准高斯

$$\mathcal{N}(\boldsymbol{Z}_t,\sigma_Z^2),$$

采用重参数化技巧进行端到端训练。

Scoring Functions and Effect Action

$$P(a_t|\boldsymbol{s}_t,\boldsymbol{Z}_t)=P(a_t|\boldsymbol{Z}_t)$$

$$\mathrm{score}(i|Z_t)=\Phi(i)^\top\boldsymbol{Z}_t$$

推荐列表分解类似于此。我们可以通过softmax函数来定义每个项目的选取概率$P(i|Z_t)$，其中$\mathcal{I}$表示项目集合。

Shared Critic and The Inverse Module

批评的目标是评估状态-动作对或预测状态的价值，以便有效地指导学习和探索。 相比于传统的RL框架，新的问题设定允许我们评估具有$Q(\boldsymbol{s}_t,\boldsymbol{Z}_t)$的超动作或具有$Q(\boldsymbol{s}_t,\boldsymbol{a}_t)$的实际动作或两者。为了保证对不同空间的动作进行一致的评估，我们提出了共享批评网络，可以从$Q(\boldsymbol{s}_t,\boldsymbol{a}_t)$转移到$Q(\boldsymbol{s}_t,\boldsymbol{Z}_t)$。 共享批评是一个映射函数

$$g\colon\ \mathcal{S}\times\mathcal{Z}\to\mathbb{R},$$

它接收用户状态$\boldsymbol{s}_t$和在核空间$\boldsymbol{Z}_t$中嵌入的动作，或等价地说，

此外，还引入了一个反向模块h来推断超动作回去。

$$\hat{Z}_t = h(a_t) = \text{pooling}(\Phi(i)|i \in a_t)$$

评估方式为

$$Q(s_t, a_t) = g(s_t, \hat{Z}_t)。$$

在实践中，我们发现将物品嵌入在核空间的平均池化[+]生成的结果最稳定，尽管存在无穷多个可以生成相同列表的潜在Z。相比于PG-RA等现有的使用相邻状态进行隐含动作推断的方法，我们认为推荐任务中的效果动作有足够的信息来恢复超动作。此外，我们还使用一个重构损失，通过对齐损失函数进一步调节超动作和效果动作的一致性。正如我们在下一段中所描述的那样，它确保生成的超动作Z_t处于核空间的有效区域，靠近候选物品。

Overall Learning Framework

整体优化过程采用修改的Actor-Critic[+]框架，包含批评损失、演员损失、超演员损失和监督损失。经验重放缓冲区$\mathcal{D}$用于收集$(s_t, a_t, r(s_t, a_t), s_{t+1}, d)$格式的数据，其中$d$表示用户离线的终止信号。批评损失目标是训练一个准确评估器来捕捉行动质量模式。

$$\mathcal{L}_{\text{TD}} = \mathbb{E}_{\mathcal{D}}\left[(r(s_t, a_t) + \gamma(1-d)Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2\right]$$

$$\mathcal{L}_{\text{QMax}} = \mathbb{E}_{\mathcal{D}}\left[Q(s_t, Z_t)\right]$$

给定Z_t,我们将学习批评家和演员的动作空间，为了保证生成的超动作模式不崩溃，我们需要使两个空间对齐，为此我们采用L2正则化策略。

$$\mathcal{L}_{\text{Hyper}} = \mathbb{E}_{D}\left[\|Z_t - \hat{Z}_t\|^2\right]$$

其中Z_t由状态s_t生成，$hat Z_t$是首先使用Z_t贪婪地选择效果动作，然后通过反向模块（如第描述部分所述）恢复超行动。同时，为了稳定RL并利用每个项目用户的详细响应信号，我们还添加了一个基于效果动作的监督学习目标。

$$\mathcal{L}_{\text{BCE}} = \mathbb{E}\left[\sum_{i \in a_t} y_{t,i} \log P(i|Z_t) + (1 - y_{t,i}) \log(1 - P(i|Z_t))\right]$$

提醒读者，有其他先进的监督和正则化方法[+]，如在线训练监督效果动作空间或适合超行动空间距离控制。将学习范式总结为算法，注意反向模块参数来自项核。

procedure HAC 初始化演员、评论家和项目内核函数中的所有可训练参数。初始化重播缓冲区$\mathcal{B}$. 在运行的剧集中应用当前策略，收集并存储样本到$\mathcal{B}$.从

$$(s_t, a_t, r(s_t, a_t), s_{t+1}, d) \sim \mathcal{B}$$

采样小批量样本。

Exploration in Hyper-Action Space and Effect-Action Space

读者可能会注意到，引入潜在的超动作引入了额外的动作采样步骤，如图所示。因此，产生的框架允许在超动作空间上进行采样（例如添加高斯噪声[+]）以及在效果-行动空间上进行采样（例如基于排名分数的类别采样）。从理论上讲，这表明效应动作的采样概率应描述为：

$$P(s) = P(e(s)) * P(a(s|e(s)))$$

其中，s表示状态，$e(s)$表示影响该状态的所有可能超动作，$a(s|e(s))$表示在该状态下执行的每个超动作的概率，而$P(e(s))$则表示所有可能超动作发生的概率。

$$P(a_t|s_t) = \int_{Z_t} P(a_t|Z_t)P(Z_t|s_t)$$

知乎

探索不足，过大会导致不稳定。我们需要将Z_t和在核空间中嵌入的项目分布对齐，如在第;中所述。通过项核函数折叠可能更有效的探索行动空间可以增强策略表达能力，见图。

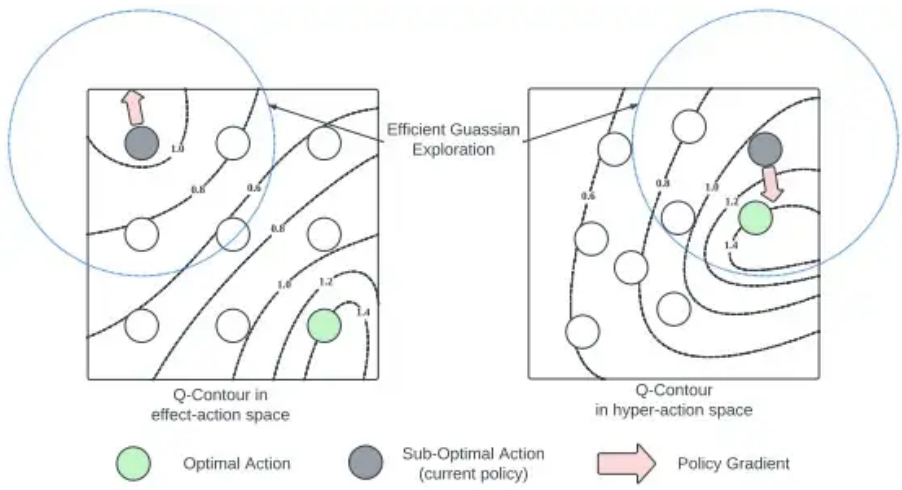


Figure 3: Exploration in Different Action Spaces

虽然我们不确定所有RL解都能探索到潜在行动空间，但我们会展示将用户和项目编码到相同的核空间，并使用反卷积$^+$模块调节动作有效性的效果。

Algorithm 1 Hyper-Actor Critic Training

procedure HAC
Initialize all trainable parameters in the actors, critics, and the item kernel function.
Initialize replay buffer $\mathcal{B}$.
while Not Converged, in each iteration **do**
  Apply current policy in running episodes, collect and store samples to $\mathcal{B}$.
  Sample mini-batch of $(s_t, a_t, r(s_t, a_t), s_{t+1}, d) \sim \mathcal{B}$.
  Update actor and critic with loss Eq.(8) and Eq.(7).
  Update actor and the kernel with loss Eq.(9), if any action alignment.
  Update actor and the kernel with loss Eq.(10), if any supervision.
end while

Experimental Settings

Datasets:

我们使用了三个公开的数据集：RL4RS、ML1M和KuaiRand1K，它们分别是会话型数据集、电影列表数据集和最近的连续短视频推荐数据集。我们首先将这三组数据集全部预处理成统一格式，以便后续的研究。在这个过程中，我们确保每条记录都包含了（用户特征，用户历史，曝光物品，用户反馈和时间戳）的顺序排列，并且详细的信息可以在附录中找到，而得到的结果数据集统计信息可以在表中提供。

RL4RS	-	283	781,367	9
MovieLens-1M	6400	3706	1,000,208	10
KuaiRand	986	11,643	969,831	10

Online Environment Simulator

为每个数据集训练一个用户响应模型$\Psi(s,a) \rightarrow R$, 其中s是用户状态，a是用户动作，R是项目奖励。用户状态基于静态用户特征和动态历史交互。Ψ输出在推荐项a_t上用户对正面反馈的可能性的概率，然后通过均匀采样得到响应$y_t \in \{0,1\}^k$。环境的详细信息请参见附录。

Models and Baselines

- 线性SL：SASRec模型通过即时用户反馈而非长期奖励进行学习。
- A2C: A3C的同步版本，它在效果动作空间上应用策略梯度。
- DDPG：超动作空间的深度DPPG框架，供演员和批评家使用。等价于无监督的HAC模型。
- TD3使用双重Q学习来提高策略训练的稳定性。
- 这是一种采用 方法中的动作表示学习的DDPG框架。这种方法最接近我们的工作，它调节效应行动，而我们的HAC模型则对超行动进行对齐

我们引入了离线SL方法，利用等式进行优化，并基于离线数据而非在线环境。模型架构和规格在附录中给出。

Evaluation

将数据集分为训练集80%和评估集20%，按时间戳记录。在训练集上预训练在线环境，然后在所有数据集上对另一个在线环境进行预训练，用于评估。训练过程中的奖励折扣设置为0.9，限制交互深度不超过20次。大部分RL方法在50k迭代内收敛并稳定。考虑总回报和深度作为长期评估指标，较高的值表示更好性能。通过奖励方差指标评估学习到的政策稳定性，较低的值表示更稳定的政策。注意，此指标描述的是状态的方差而非随机种子的方差。每种实验都评估所有上述指标，并报告不同用户会话的平均值。

Effectiveness

对于所有数据集，我们将其分为训练集的前80%和评估集的后20%，根据记录的时间戳。然后，我们在训练集中预训练在线环境，并在所有数据集上对另一个在线环境进行预训练，以便稍后进行评估。 我们在第一个环境中训练我们的推荐策略，并在第二个环境中对其进行评估。 在训练过程中，我们将奖励折扣设置为$\gamma = 0.9$，并限制所有实验的交互深度为≤ 20。

Model	RL4RS		ML1M		KuaiRand	
	Total Reward	Depth	Total Reward	Depth	Total Reward	Depth
Offline SL	6.721	8.163	18.559	18.717	14.394	14.982
Online SL	9.502	10.571	18.629	18.780	13.456	14.147
A2C	7.789	9.140	16.158	16.556	12.460	13.250
DDPG	8.337	9.588	17.205	17.508	11.394	12.313
TD3	8.553	9.791	17.545	17.814	11.777	12.664
PG-RA	8.561	9.728	18.466	18.633	10.859	11.814
HAC	**10.059**	**11.102**	**18.863**	**18.988**	**14.789**	**15.335**

Table 2: Online Performance. The best performances in bold and second best in Underline

我们发现大多数RL方法都在50,000个迭代内收敛并稳定。 对于长期评估指标，我们考虑总回报（代表用户会话中所有奖励的总和）和深度（代表用户和每个会话如何使用模拟的在线环境与学习的政策互动）。对于这两个指标，更高的值意味着更好的性能。 为了评估学习到的政策的稳定性，我们包括一个奖励方差指标，该指标估计一个政策如何处理不同的用户状态，因此较低的值表示更稳定的政策。 请注意，此指标描述的是状态的方差而不是随机种子的方差。 在每个实验中，我们评估所有上述指标，并报告不同用户会话的平均值。

Learning HAC

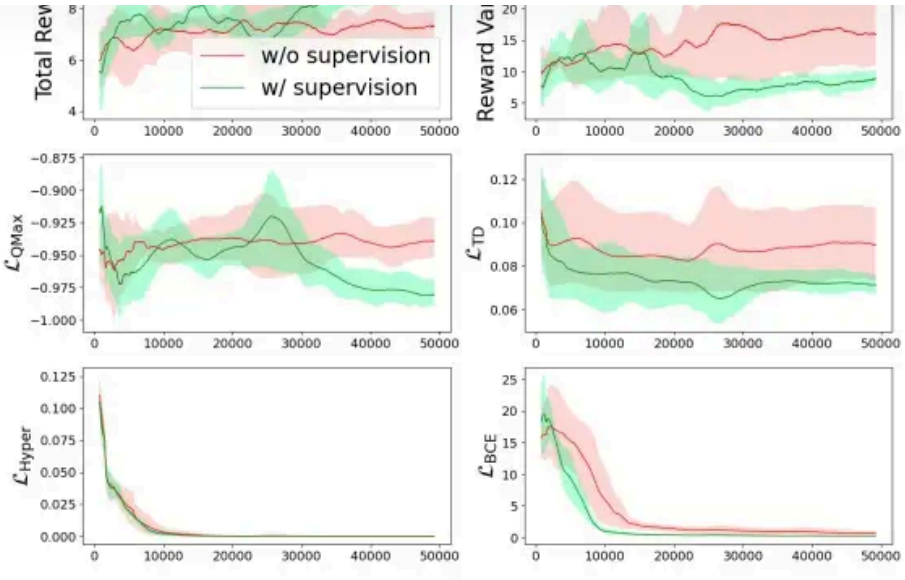


Figure 4: Training curves of HAC without supervision on RL4RS. X-axis corresponds to the number of iteration. The four losses are presented in log scales.

图显示了HAC的学习曲线，以及它与没有监督损失梯度的"HAC无监督"方法的比较。所有的方法都达到了性能饱和。 总奖励 作为惩罚的总体表现良好，而"HAC无监督"也能成功降低四个在第3节中提到的损失函数。值得注意的是，尽管"HAC无监督"能有效减少每个项目的BCE损失，但基于奖励的方法对于每个项目自身的信号也能有效地改进。

此外，引入监督机制可以显著提高模型性能，并减少有助于探索更好动作的Qmax损失。从图1可以看出，HAC的TD损失低于"HAC无监督"，表明学习过程更加稳定。我们可以通过观察不同用户状态下总奖励的方差来验证这一结论，因为方差越高，表示学习策略在不同用户状态下不能提供良好的动作。例如，在Kuai Rand环境下，增加对监督损失的重要性可以帮助改善推荐性能并减少方差。然而，如果过度使用监督模块的学习率，可能会导致过于依赖用户的意图，从而损害性能。在其他两个数据集中也观察到了类似的现象，这些结果在附录中给出。

通过比较不同数量的超行动对损失方程的影响，我们发现HAC框架在总回报和奖励方差的收敛方面比只使用监督的HAC（$\lambda_h = 0$）慢，因为包括了超行动对齐（$\lambda_h = 0.1$和$\lambda_h = 1$）。相反，更一致的动作空间可以帮助模型更好地学习和探索更好的动作策略。同时，增加这个对齐模块的重要性导致$\mathcal{L}_{\text{QMax}}$更差，$\mathcal{L}_{\text{TD}}$更好，这表明批评家更能捕捉到动作的质量。需要注意的是，$\lambda_h = 1$比$\lambda_h = 0.1$更稳定，但可能不如$\lambda_h = 0.1$有效。为了验证这一点，我们在图中展示了评估结果，结果显示最佳点为$\lambda_h = 0.1$，推荐更高且更稳定的回报。

Ablation Study

比较不同学习模块的工作方式：DDPG（无监督和动作对齐的HAC，使用超行动空间进行演员学习和评论学习）、HAC w/o $\mathcal{L}_{\text{BCE}}$（排除HAC的监督）、HAC w/o $\mathcal{L}_{\text{Hyper}}$（不使用$\mathcal{L}_{\text{Hyper}}$）。

在HAC模型中，我们通过探索超动作空间和效果动作空间来提高性能。通过改变高斯噪声的方差并在其他超参数保持不变的情况下，我们可以比较不同大小的超行动探索的效果。结果显示了在一个特定搜索空间的最优点，这强调了需要精心设计探索以避免采样方差过小或过大。根据第3章的讨论，小方差可能会限制新行动的探索，而大方差会导致不稳定探索，难以收敛。实验也发现，使用top-k贪婪选择可以取得最佳结果，而添加分类采样则可能使学到的策略无法优化。

Conclusion

我们提出一种演员-批评学习框架，在离散效果动作空间和向量化的超动作空间中优化推荐策略。实验验证了我们的方法的有效性和稳定性。设计的目的是通用且可容纳多种现有解决方案框架，如Actor-Critic、DDPG和监督学习。此外，我们也尝试将超行动视为应用更复杂的评分函数的初步尝试。

知乎

推荐系统 强化学习 (Reinforcement Learning) 快手

理性发言，友善互动

还没有评论，发表第一个评论吧

推荐阅读

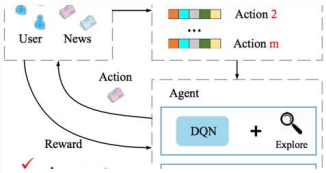
快手强化学习与多任务推荐

DataF... 发表于人工智能算...

增量学习概览

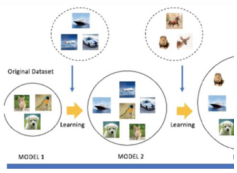
增量学习是一种机器学习方法，模型逐步学习和增强其知识，而不会忘记以前获取的信息。从本质上讲，它通过随着时间的推移获取新信息来模仿人类的学习模式，同时维护和建立以前的知识。在数据...

爱吃牛油果的璐璐



强化学习系列一——基于深度强化学习的新闻推荐模型DRN

yymWa... 发表于Beyon...



增量学习（新样本出现时忆学习）

海布里啦 发表于机器学