中国科学: 信息科学 2024 年 第 54 卷 第 8 期: 1983–2000
SCIENTIA SINICA Informationis
论文

预测资源分配: 马尔可夫决策过程的无监督学习

吴佳骏, 赵剑羽, 孙乘坚, 杨晨阳*

北京航空航天大学电子信息工程学院, 北京 100191

* 通信作者. E-mail: cyyang@buaa.edu.cn

收稿日期: 2024-01–08; 修回日期: 2024–04–25; 接受日期: 2024–06–04; 网络出版日期: 2024–08–01

国家重点研发计划 (批准号: 2022YFB2902002)、国家自然科学基金重点项目 (批准号: 61731002) 和国家自然科学基金面上项目 (批准号: 62271024) 资助

摘要 当已知未来的移动轨迹等信息时, 面向视频点播业务的预测资源分配可以在满足用户体验的前提下降低基站能耗或提高网络吞吐量. 传统的预测资源分配方法采用先预测用户轨迹等信息再优化功率等资源分配的方法, 在预测窗较长时预测误差大, 导致预测所带来的增益降低. 为了解决这个问题, 近期已有文献把预测资源分配建模为马尔可夫决策过程, 采用深度强化学习进行在线决策. 然而, 对于这类适于采用强化学习的马尔可夫决策过程, 现有文献往往以试错的方式对状态进行设计. 此外, 对于有约束的优化问题, 现有利用强化学习解决无线问题的方法大多通过在奖励函数上加入包含需要手动调节超参数的惩罚项满足约束. 本文以移动用户视频播放不卡顿约束下使基站发射能耗最小的问题为例, 提出在线求解预测资源分配的无监督深度学习方法对信息预测和资源分配进行联合优化, 并建立这种方法与深度强化学习的联系. 所提出的方法可以通过在线端到端无监督深度学习提高预测资源分配的性能, 能以系统化而非试错式的方式设计状态, 可以自动而非通过引入超参来满足复杂的约束. 仿真结果表明, 所提出的在线无监督深度学习与深度强化学习所达到的发射能耗相近, 但能够简化状态的设计, 验证了理论分析结果.

关键词 预测资源分配, 马尔可夫决策过程, 无监督深度学习, 深度强化学习, 状态设计, 复杂约束

1 引言

当已知未来的移动轨迹等用户行为信息时, 面向视频点播业务和文件下载等延时不敏感业务的预测资源分配可以在保证用户体验的前提下降低基站能耗或提高网络吞吐量. 这种通过预测用户行为信息提高无线资源利用率的基本思想, 是根据一个预测窗内多个帧的未来信道信息或未来数据率 (其中一个帧内信道基本不变), 在用户信道条件好时多传数据, 在信道条件差时少传数据. 已有的研究结果表明, 基于分钟级大尺度信道预测的预测资源分配可以在满足请求视频点播业务用户服务质量 (quality of service, QoS), 即在视频播放不卡顿的前提下有效提升网络的能效或吞吐量 [1~3].

引用格式: 吴佳骏, 赵剑羽, 孙乘坚, 等. 预测资源分配: 马尔可夫决策过程的无监督学习. 中国科学: 信息科学, 2024, 54: 1983–2000, doi: 10.1360/SSI-2024-0011
Wu J J, Zhao J Y, Sun C J, et al. Predictive resource allocation: unsupervised learning of Markov decision processes (in Chinese). Sci Sin Inform, 2024, 54: 1983–2000, doi: 10.1360/SSI-2024-0011

为了研究预测资源分配的潜力, 早期文献在假设未来一个分钟级预测窗内的信道或数据率已知的前提下优化满足用户 QoS 所需分配的未来资源 [1,2]. 文献 [1] 假设未来的瞬时数据率已知, 在保证视频点播业务不卡顿的前提下以最小化系统能耗为目标优化了预测窗内的未来数据率. 文献 [2] 假设未来平均信道增益已知, 在保证视频播放不卡顿的前提下以最大化能效为目标优化了预测窗内的功率和带宽分配. 在实际应用时, 未来的信息需要预测. 考虑到预测存在不确定性, 文献 [3] 通过把预测误差建模为随机变量, 研究了预测资源分配的鲁棒优化方法.

由于人工智能 (artificial intelligence, AI) 在自然语言处理和计算机视觉等领域的成功应用, 近几年机器学习技术已被广泛用于设计无线通信系统 [4]. 近期文献开始研究信息预测问题, 首先考虑了采用先预测再优化的方法实现预测资源分配, 即通过循环神经网络 (recurrent neural network, RNN) 等机器学习模型先预测未来的用户轨迹或平均数据率, 然后根据预测值优化在预测窗内多个帧之间的资源分配 [5,6]. 为了分析预测资源分配何时相对于非预测资源分配具有较大的增益, 文献 [7] 推导了相对增益的表达式, 分析了影响增益的关键因素. 由于先预测再优化方法没有利用新观测到的数据, 故当预测窗较长时性能较差. 为了解决这个问题, 文献 [8] 把预测资源分配建模为马尔可夫决策过程 (Markov decision processes, MDP), 采用深度强化学习 (deep reinforcement learning, DRL) 根据当前和过去观测的环境信息进行了在线决策.

利用 AI 技术的确能解决很多采用传统优化理论难以解决的复杂无线问题, 但很多在无线 AI 领域中的研究也沿袭了其他 AI 领域的经验性设计方法. 在无线 AI 文献中有大量的工作采用 DRL 来优化无线问题, 动机是强化学习具有以在线、端到端的方式从无模型或非凸问题中学习策略的能力. 状态设计是应用 DRL 解决各种问题的关键. 很多无线问题并非 MDP, 其状态设计与深度神经网络 (deep neural network, DNN) 的输入特征设计类似. 对于建模为 MDP 的问题, 状态设计非常困难, 一般采用启发式方法. 这是因为, 状态是强化学习用于决策的充分统计量, 而对于 MDP, 如在线预测资源分配这种目标函数与多时间步决策有关的问题, 很难确定充分统计量. 对于适于采用强化学习的 MDP 问题, 现有文献往往直接给出通过试错方式设计出的状态 [9~11], 却不提及设计过程. 此外, 与其他领域不同, 很多无线任务往往需要满足复杂的瞬时约束. 对于有约束的优化问题, 现有无线 AI 文献大多通过在奖励函数上加入包含可手动调节超参数的惩罚项来启发式地满足约束 [11,12]; 尽管可以采用针对约束下的马尔可夫决策过程 [13] 设计的强化学习方法, 但这些方法只适于解决长期约束问题, 无法满足预测资源分配问题中的瞬时约束. 这种试错式的经验性设计方法极为繁琐, 且无法保证根据所选择的状态或超参学习得到的解最优.

实际上, 无监督深度学习也能以在线、端到端的方式从无模型或非凸问题中学习策略 [14], 而且能同时满足瞬时和长期约束. 与自编码器和聚类等经典的无监督机器学习不同, 文献 [15] 针对约束优化问题提出的无监督深度学习以优化问题的拉格朗日函数为损失函数训练 DNN, 无需加入包含超参的惩罚项来启发式地满足约束. 然而, 现有无监督深度学习方法只能解决非 MDP 问题 [14].

为了解决强化学习方法和无监督深度学习存在的这些问题, 本文以预测资源分配问题为例, 研究能联合优化信息预测和资源分配, 并自动满足瞬时 QoS 约束的无监督深度学习方法, 可以解决 MDP 问题并能对 MDP 的状态进行系统化设计 (即无需试错). 最近, 无线 AI 领域已有文献针对信道估计、信道预测和预编码进行端到端优化 [16~18]. 例如, 文献 [16] 采用一个 RNN 进行信道预测和一个神经网络学习功率分配策略而后通过和数据率最大预编码矩阵的结构计算预编码, 通过把损失函数设计为信道预测的均方误差 (mean square error, MSE), 学习功率分配的 MSE 与和数据率的加权和, 以监督学习的方式联合训练两个神经网络. 文献 [17] 设计了两层 DRL 的结构来进行信道预测和预编码优化, 其中第 1 层利用一个 RNN 进行信道预测, 第 2 层把第 1 层的预测结果作为输入学习预编码策略. 文

献 [18] 则设计了 DNN 直接学习从上行导频到下行预编码之间的映射, 以预编码的目标函数 (即和数据率最大) 的负值为损失函数以无监督的方式训练 DNN. 以上这些问题都不是 MDP 问题, 即使采用强化学习其状态设计也不困难. 另外, 现有文献都并没有对端到端优化问题进行严格的数学描述, 因此采用了启发式的设计方法.

为了系统化, 而非启发式或试错式地设计深度学习方法解决无线问题, 本文首先对于联合信息预测和资源分配建模了主动优化问题, 从而能采用无监督深度学习以端到端的方式学习从历史数据到未来最优决策之间的映射. 而后, 提出在每个时间步利用所观测到的环境变量在线优化决策变量的方法. 为了符号简单且便于理解, 考虑单用户视频点播业务中的预测资源分配, 但本文的分析不难扩展到多用户问题.

本文的主要贡献如下:

• 建模了主动优化问题对信息预测和资源分配进行联合优化, 对端到端优化问题进行了严谨的数学描述, 提出了将其转化为多时间步和单时间步在线主动优化问题的方法, 从而能够利用最新观测的数据并通过递推式单步预测提高多步预测的性能.

• 通过无监督深度学习在线求解单时间步主动优化问题, 满足了瞬时约束, 无需引入对不同场景都要手动调参的惩罚项来启发式地满足约束. 我们的研究表明, 采用无监督深度学习同样可以在线学习 MDP 问题, 并能够系统化地设计状态, 无需以试错的方式对状态进行设计. 通过指出所提出无监督学习与深度确定性策略梯度 (deep deterministic policy gradient, DDPG) 算法间的联系, 解释了通过强化学习进行联合预测与优化的机理.

• 通过仿真验证了理论分析结果, 表明所提出方法能达到与 DRL 方法几乎相同的系统性能.

本文的后续内容安排如下, 第 2 小节介绍系统模型和未来信道已知时的预测资源分配问题, 并建模主动优化问题, 第 3 小节介绍在线主动优化的无监督学习, 指出在线单步决策的无监督学习与 DDPG 的关系, 第 4 小节给出仿真结果, 第 5 小节总结全文.

2 预测资源分配

本节首先介绍系统模型, 而后描述已知未来信道时的预测资源分配问题, 最后, 建模能同时进行信道预测和资源分配的主动优化问题.

2.1 系统模型

考虑一个由多个基站和一个中心处理器组成的网络, 服务一个请求视频点播业务的移动用户. 用户就近接入基站, 即由大尺度信道增益最大的基站服务. 中心处理器可以通过基站获取用户的信息, 并根据这些信息优化基站的发射功率.

一个视频文件包含 N_v 个片段, 每个片段的播放时间包含 L_v 个时间步, 每个时间步包含 N_s 个时隙. 大尺度信道增益在每个时间步内基本不变, 在不同时间步之间可能会改变; 小尺度信道增益在每个时隙内基本不变. 每个时间步的持续时间为 ΔT, 每个时隙的持续时间为 τ. 因此, $\tau \leqslant \Delta T$. 令 $g^{ti}\alpha^t$ 表示用户与接入基站之间在第 t 个时间步内第 i 个时隙的信道增益, 其中 g^{ti} 和 α^t 分别是小尺度和大尺度信道增益. 用户在第 t 个时间步第 i 个时隙的可达数据率为 $R^{ti} = W\log_2(1+\frac{\alpha^t}{\sigma^2}p^{ti}g^{ti})$, 其中 W 是带宽, σ^2 是噪声功率, p^{ti} 是基站在第 t 个时间步第 i 个时隙的发射功率. 为了简化符号且不失一性, 假设 $L_v = 1$.

为了避免播放视频时发生卡顿, 视频片段在播放之前应下载到用户的缓冲区. 因此, 用户 QoS 约

束为 $\sum_{l=0}^{t}\bar{R}^l \geqslant \frac{1}{\Delta T}\sum_{l=1}^{t+1}B_0^l, t=0,\ldots,N_v-1$, 其中 $\bar{R}^t=\mathbb{E}_{g^{ti}}\{R^{ti}\}$ 为第 t 个时间步的平均数据率, $\mathbb{E}_{g^{ti}}\{\cdot\}$ 表示对小尺度信道取均值, B_0^l 表示第 l 个视频片段的大小, 即包含的比特数.

2.2 已知未来信道时的预测资源分配优化问题

为了收集历史数据进行预测资源分配, 在用户刚发起请求时, 即第 $t=0$ 个时间步, 用户所接入的基站首先以尽力而为的方式把第一个视频片段传输给用户. 在第 $t=1$ 个时间步, 中心处理器根据所收集的信息对预测窗内未来 N_v-1 个时间步的资源提前进行分配使传输的总能耗最小.

在第 t 个时间步内基站为传输视频所消耗的能量为 $\frac{1}{\rho}\sum_{i=1}^{N_s}\tau p^{ti}+\Delta T P_c$ [19], 其中 ρ 是功放效率, P_c 是静态功耗. 为了以文献 [2] 提出的能达到全局最优解的数值方法作为学习性能的可达上界, 假设小尺度信道增益在不同时隙之间统计独立同分布 (independent and identically distributed, i.i.d.), 则瞬时功率关于小尺度信道增益的时间平均等于其集平均, 即 $\mathbb{E}_{g^{ti}}\{\sum_{i=1}^{N_s}\tau p^{ti}\}=\Delta T\bar{P}^t$, 其中 $\bar{P}^t=\mathbb{E}_{g^{ti}}\{p^{ti}\}$ 表示第 t 个时间步的平均发射功率. 由于最小化总能耗 $\sum_{t=1}^{N_v-1}\Delta T\ (\frac{1}{\rho}\bar{P}^t+P_c)$ 等价于最小化总功耗 $\sum_{t=1}^{N_v-1}\bar{P}^t$, 故若假设未来 $(N_v-1)N_s$ 个时隙内的小尺度信道已知, 则在保证用户播放视频不卡顿的前提下使基站总能耗最小的优化问题可以建模为

$$\min_{p^{ti},\forall t,i}\quad \sum_{t=1}^{N_v-1}\bar{P}^t, \tag{1}$$

$$\text{s.t.}\quad \sum_{l=1}^{t}\bar{R}^l \geqslant \frac{1}{\Delta T}\sum_{l=2}^{t+1}B_0^l,\quad t=1,\ldots,N_v-1, \tag{1a}$$

$$0\leqslant p^{ti}\leqslant P_{\max},\quad t=1,\ldots,N_v-1, i=1,\ldots,N_s, \tag{1b}$$

其中 $P_{\max}$ 是基站的最大发射功率.

若通过求解式 (1) 的问题来优化未来每个时隙的功率分配, 则中心处理器需要在每个时隙 (通常持续时间在毫秒级别) 内从基站收集小尺度信道并向基站发出资源分配的指令, 会导致大量的信令开销. 更重要的是, 对预测窗内的 $(N_v-1)N_s$ 个小尺度信道增益进行预测非常困难, 一般不可能. 因此, 可以把式 (1) 中的问题等效地转化为如下问题 [20]:

$$\min_{\bar{R}^t,t=1,\ldots,N_v-1}\quad \sum_{t=1}^{N_v-1}\bar{P}^t, \tag{2}$$

$$\text{s.t.}\quad \text{式 (1a)},$$

$$0\leqslant \bar{R}^t\leqslant R_{\max},\ t=1,\ldots,N_v-1, \tag{2a}$$

其中 $R_{\max}=\mathbb{E}_{g^{ti}}\{W\log_2(1+\frac{\alpha^t}{\sigma^2}P_{\max}g^{ti})\}$ 是用户在第 t 个时间步的最大可达平均数据率, 可以表示为 $R_{\max}(\alpha^t)$. 式 (2a) 为最大发射功率约束所对应的最大数据率约束. 这个优化问题目标函数的时间尺度为 N_v-1 个时间步, 而优化变量和约束的时间尺度为单个时间步, 属于瞬时约束下的优化问题.

通过求解式 (2) 的问题找到最优的 $\bar{R}^t$ 后, 可以根据如下公式得到最优功率分配 [20]:

$$p^{\text{opt}}=\begin{cases}\dfrac{\sigma^2}{\alpha^t}\left(\dfrac{1}{g_{\text{th}}^t}-\dfrac{1}{g^{ti}}\right), & g^{ti}\geqslant g_{\text{th}}^t,\\ 0, & g^{ti}<g_{\text{th}}^t.\end{cases} \tag{3}$$

式 (3) 中的 g_{th}^t 和式 (2) 中的 $\bar{P}^t$ 都可以表示为如下与 $\bar{R}^t$ 有关的形式 [8]:

$$g_{\text{th}}^t=-\text{Ei}^{-1}\left(-\frac{\bar{R}^t\ln 2}{W}\right), \tag{4}$$

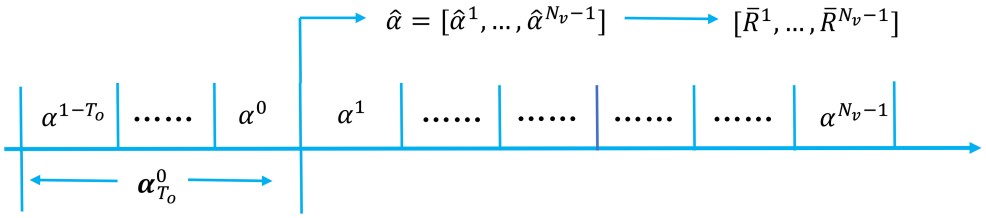

图 1 (网络版彩图) 先预测再优化过程

Figure 1 (Color online) Procedure of first prediction and then optimization

$$\bar{P}^t = \int_{g_{\text{th}}^t}^{\infty} \frac{\sigma^2}{\alpha^t} \left(\frac{1}{g_{\text{th}}^t} - \frac{1}{g^{ti}} \right) \mathrm{e}^{-g^{ti}} \mathrm{d}g^{ti} \triangleq \bar{P}^t(\alpha^t, \bar{R}^t), \tag{5}$$

其中 $\mathrm{Ei}(-x) \triangleq -\int_{-x}^{\infty} \frac{\mathrm{e}^{-t}}{t} \mathrm{d}t$.

此时中心处理器只需要在每个时间步 (通常持续时间在秒级别) 内与基站进行信息交互, 且只需预测未来 $N_v - 1$ 个时间步的大尺度信道增益.

2.3 先预测再优化与主动优化的问题建模

为了通过求解式 (2) 中问题和式 (3) 得到未来每个时隙的功率分配, 需要对预测窗内的大尺度信道增益 $\alpha^t, t = 1, \ldots, N_v - 1$ 进行预测.

2.3.1 先预测再优化

传统的解决方法是先预测再优化, 如图 1 所示.

这种方法包含两个阶段. 第一个阶段通过一个在长度为 T_o 个时间步的观测窗内采集的大尺度信道增益时间序列 $\boldsymbol{\alpha}_{T_o}^0 \triangleq [\alpha^{1-T_o}, \ldots, \alpha^0]$ 得到未来多个时间步的大尺度信道预测 $\hat{\boldsymbol{\alpha}} \triangleq [\hat{\alpha}^1, \ldots, \hat{\alpha}^{N_v-1}]$. 信息预测通常以最小化预测值和未来真值之间的均方误差为目标, 即未来第 t 个时间步的大尺度信道增益 α^t 可以通过求解如下使 MSE 最小的优化问题进行预测:

$$\min_{\hat{\alpha}^t} \mathbb{E}_{\alpha^t | \boldsymbol{\alpha}_{T_o}^0} \left\{ \left(\hat{\alpha}^t - \alpha^t \right)^2 \right\}, \tag{6}$$

其中 $\hat{\alpha}^t, t = 1, \ldots, N_v - 1$ 为预测窗内第 t 个时间步大尺度信道增益的预测值. 第二个阶段则通过把预测值当作真值来优化预测窗内未来每个时间步的平均数据率, 即

$$\min_{\bar{R}^t, t=1,\ldots,N_v-1} \quad \sum_{t=1}^{N_v-1} \bar{P}^t(\hat{\alpha}^t, \bar{R}^t), \tag{7}$$

$$\text{s.t.} \quad \text{式 (1a)},$$

$$0 \leqslant \bar{R}^t \leqslant R_{\max}(\hat{\alpha}^t), \ t = 1, \ldots, N_v - 1. \tag{7a}$$

2.3.2 主动优化

由式 (6) 和 (7) 可见, 信息预测与资源优化的目标函数不同, 使 MSE 最小的预测值未必能使预测资源分配在满足约束的前提下达到最小的总功耗. 为了解决这一问题, 下面提出联合进行信息预测和资源分配优化的方法, 即根据在观测窗内采集的大尺度信道增益来优化预测窗内多个时间步的决策变量, 称为主动优化方法, 如图 2 所示.

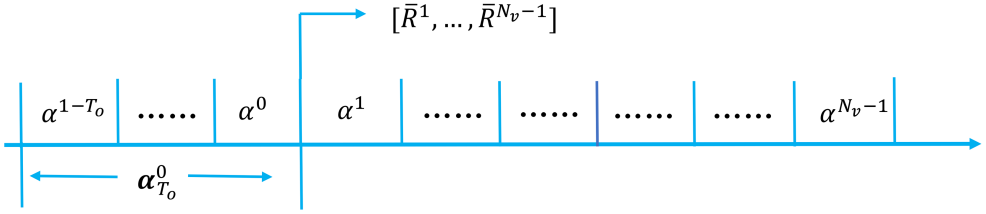

图 2　(网络版彩图) 主动优化过程

Figure 2　(Color online) Procedure of proactive optimization

由于预测的目的是通过进行预测资源分配满足 QoS 约束并使所需的总功耗最小, 故隐含信息预测的预测资源分配问题 (称为主动优化问题) 可建模为

$$\min_{\bar{R}^t, t=1,\ldots,N_v-1} \quad \sum_{t=1}^{N_v-1} \mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\bar{P}^t(\alpha^t, \bar{R}^t)\}, \tag{8}$$

$$\text{s.t.} \quad \text{式 (1a)},$$

$$0 \leqslant \bar{R}^t \leqslant \mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{R_{\max}(\alpha^t)\},\ t = 1, \ldots, N_v - 1, \tag{8a}$$

其中, 由于 $\bar{P}^t(\alpha^t, \bar{R}^t)$ 和 $R_{\max}(\alpha^t)$ 与需要预测的未来大尺度信道增益 α^t 有关, 所以在目标函数和约束 (8a) 上引入了关于 α^t 的条件均值, 即 $\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\cdot\}$.

2.3.3　先预测再优化与主动优化间的关系

由于主动优化的优化目标和约束与原始优化问题相同, 与文献 [18] 中的联合信道估计与预编码优化问题类似, 主动优化的性能不劣于先预测再优化的性能. 下面给出二者性能相同的条件.

当先预测再优化方法中的预测以 MSE 最小为目标时, 对大尺度信道增益得到的预测值为

$$\hat{\alpha}^t = \mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\alpha^t\}, \tag{9}$$

把式 (9) 代入式 (7) 中的目标函数和式 (7a) 中的约束, 则式 (7) 中的问题变为

$$\min_{\bar{R}^t, t=1,\ldots,N_v-1} \quad \sum_{t=1}^{N_v-1} \bar{P}^t(\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\alpha^t\}, \bar{R}^t), \tag{10}$$

$$\text{s.t.} \quad \text{式 (1a)},$$

$$0 \leqslant \bar{R}^t \leqslant R_{\max}(\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\alpha^t\}),\ t = 1, \ldots, N_v - 1. \tag{10a}$$

如果 $\bar{P}^t(\alpha^t, \bar{R}^t)$ 和 $R_{\max}(\alpha^t)$ 关于 α^t 是线性的, 那么可以推出

$$\sum_{t=1}^{N_v-1} \mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\bar{P}^t(\alpha^t, \bar{R}^t)\} = \sum_{t=1}^{N_v-1} \bar{P}^t(\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\alpha^t\}, \bar{R}^t), \tag{11}$$

$$\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{R_{\max}(\alpha^t)\} = R_{\max}(\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}^0_{T_o}}\{\alpha^t\}),\ t = 1, \ldots, N_v - 1. \tag{12}$$

把式 (11) 和 (12) 分别代入式 (8) 中的目标函数和式 (8a) 中的约束, 则主动优化问题与式 (10) 中的问题完全相同. 这意味着当先预测再优化中的预测问题以 MSE 最小为目标时, 若 $\bar{P}^t(\alpha^t, \bar{R}^t)$ 和 $R_{\max}(\alpha^t)$ 是 α^t 的线性函数, 则先预测再优化与主动优化等价.

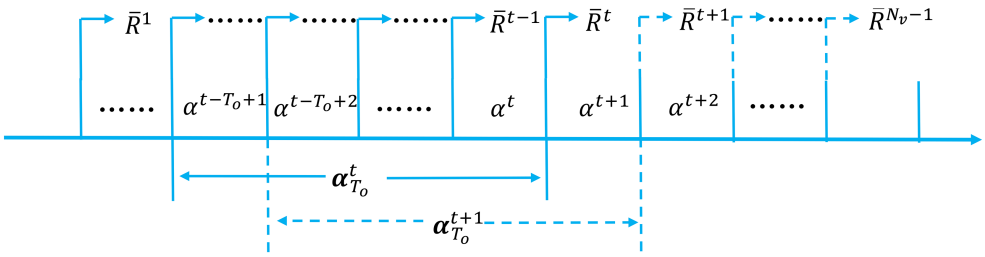

图 3 (网络版彩图) 在线单步决策的主动优化过程

Figure 3 (Color online) Procedure of on-line and single step proactive optimization

3 在线主动优化的无监督学习

由于在一般情况下难以推导条件均值 $\mathbb{E}_{\alpha^t|\boldsymbol{\alpha}_{T_o}^0}\{\cdot\}$, 式 (8) 中主动优化问题的目标函数和约束 (8a) 都没有显式表达式, 因此难以用传统数值方法求解. 可以采用文献 [15] 中无监督深度学习的方法, 根据主动优化问题的目标函数和约束构造损失函数, 训练 DNN 来学习从观测窗内数据到预测窗内多个决策变量的映射. 不过, 这种方法存在以下两个问题: (1) 仅利用了第 $t=0$ 个时间步的观测数据 $\boldsymbol{\alpha}_{T_o}^0$, 没有利用最新观测到的大尺度信道数据, 因此预测性能差; (2) 由于需要进行多步决策, 距离初始时刻越远, 相应时间步大尺度信道增益的预测性能越差, 所对应的资源分配性能也越差.

为了解决第一个问题, 可以采用在线主动优化, 在每个时间步不仅利用历史数据也利用新观测的数据对预测窗内的优化变量进行决策, 即学习从一个滑动的观测窗内数据 $\boldsymbol{\alpha}_{T_o}^t \triangleq [\alpha^t, \ldots, \alpha^{t-T_o+1}]$ 到一个滑动的预测窗内多个决策变量的映射. 为了解决第二个问题, 可以采用在线单步决策的主动优化, 即学习从滑动观测窗内数据 $\boldsymbol{\alpha}_{T_o}^t$ 到预测窗内单个决策变量的映射, 如图 3 所示.

为此, 本节首先建模在线多步决策的主动优化问题, 然后建模在线单步决策的主动优化问题, 最后通过无监督学习进行求解, 并指出与强化学习的关系.

3.1 在线多步决策的主动优化问题

对于在线多步决策的主动优化, 需要在第 t 个时间步根据第 $1\sim t-1$ 个时间步的决策 $\bar{R}^1\sim\bar{R}^{t-1}$ 和第 t 个时间步观测窗内的数据 $\boldsymbol{\alpha}_{T_o}^t$ 优化未来多个时间步的决策 $\bar{R}^t\sim\bar{R}^{N_v-1}$.

由于第 $1\sim t-1$ 个时间步的决策已完成, 即 $\bar{R}^1\sim\bar{R}^{t-1}$ 的值已知, 故在目标函数和约束中无需再考虑与这些时间步有关的项. 因此, 式 (8) 问题中的约束 (1a) 变为

$$\sum_{l=1}^{i}\bar{R}^l \geqslant \frac{1}{\Delta T}\sum_{l=2}^{i+1}B_0^l, \quad i=t,\ldots,N_v-1. \tag{13}$$

由于第 t 个时间步, 历史决策 $\bar{R}^1,\ldots,\bar{R}^{t-1}$ 已知, 为了简化符号进行如下定义:

$$B^t \triangleq \sum_{l=1}^{t-1}\bar{R}^l, \tag{14}$$

其中 B^t 为到第 t 个时间步为止基站所传输给用户的总数据量. 把式 (14) 代入式 (13), 并进一步用第 t 个时间步的观测 $\boldsymbol{\alpha}_{T_o}^t$ 替代 $\boldsymbol{\alpha}_{T_o}^0$, 则根据式 (8), 在线多步决策的主动优化问题为

$$\min_{\bar{R}^i,i=t,\ldots,N_v-1}\ \bar{P}^t(\alpha^t,\bar{R}^t)+\sum_{i=t+1}^{N_v-1}\mathbb{E}_{\alpha^i|\boldsymbol{\alpha}_{T_o}^t}\{\bar{P}^i(\alpha^i,\bar{R}^i)\}, \tag{15}$$

$$\text{s.t.}\quad B^t+\sum_{l=t}^{i}\bar{R}^l\geqslant\frac{1}{\Delta T}\sum_{l=2}^{i+1}B_0^l,\quad i=t,\ldots,N_v-1, \tag{15a}$$

$$0\leqslant\bar{R}^t\leqslant R_{\max}(\alpha^t), \tag{15b}$$

$$0\leqslant\bar{R}^i\leqslant\mathbb{E}_{\alpha^i|\boldsymbol{\alpha}_{T_o}^t}\{R_{\max}(\alpha^i)\},\ i=t+1,\ldots,N_v-1, \tag{15c}$$

其中式 (15a) 为当前的第 t 个时间步和未来第 $t+1\sim N_v-1$ 个时间步用户的 QoS 约束, 式 (15b) 和 (15c) 分别为当前时间步和未来第 $t+1\sim N_v-1$ 个时间步的最大可达数据率约束.

3.2 在线单步决策的主动优化问题

式 (15) 中问题的优化变量包含多个时间步的决策, 且其目标函数和式 (15c) 中有条件均值. 为了对这类多步决策问题通过递推的方式进行求解, 可以将其转化为多阶段随机优化问题 [21], 从而得到最优决策序列 $\{\bar{R}^i\}_{i=t}^{N_v-1}$. 为此, 下面首先分析第 t 个时间步的状态, 即系统在当前时间步所能获取的所有影响当前决策的环境变量.

由式 (15a) 可知, 第 t 个时间步的决策 $\bar{R}^t$ 与 B^t, 未来第 $t+1\sim N_v-1$ 个时间步的决策 $\bar{R}^{t+1}\sim\bar{R}^{N_v-1}$ 以及 $B_0^2\sim B_0^{N_v-1}$ 有关; 而由目标函数和式 (15c) 可以看出, 未来第 $t+1\sim N_v-1$ 个时间步的决策依赖于第 t 个时间步的观测 $\boldsymbol{\alpha}_{T_o}^t$, 因此 $\bar{R}^t$ 也与 $\boldsymbol{\alpha}_{T_o}^t$ 有关. 因此, 所有影响 $\bar{R}^t$ 的可观测量可以表示为

$$\boldsymbol{s}^t\triangleq[\boldsymbol{\alpha}_{T_o}^t,B^t,B_0^2,\ldots,B_0^{N_v-1}]. \tag{16}$$

可以把式 (14) 写成如下的递推形式:

$$B^t=\sum_{l=1}^{t-1}\bar{R}^l=\sum_{l=1}^{t-2}\bar{R}^l+\bar{R}^{t-1}=B^{t-1}+\bar{R}^{t-1}. \tag{17}$$

由于 $\boldsymbol{s}^t$ 中包含 B^t, 故 $\boldsymbol{s}^t$ 受到 $\boldsymbol{s}^{t-1}$ 中的元素 B^{t-1} 和决策 $\bar{R}^{t-1}$ 的影响, 可见对预测资源分配进行在线单步决策的主动优化问题属于 MDP. 以下称式 (16) 中的 $\boldsymbol{s}^t$ 为第 t 个时间步的状态.

为了建模在线单步决策问题, 首先根据 (17) 把第 i, $i>t$ 个时间步基站已经传输给用户的数据量表示成如下形式:

$$B^t+\sum_{l=t}^{i}\bar{R}^l=\sum_{l=1}^{t-1}\bar{R}^l+\sum_{l=t}^{i}\bar{R}^l=\sum_{l=1}^{i-1}\bar{R}^l+\bar{R}^i=B^i+\bar{R}^i. \tag{18}$$

而后把式 (18) 代入式 (15a), 即可得到如下对每个时间步具有统一形式的 QoS 约束:

$$B^i+\bar{R}^i\geqslant\frac{1}{\Delta T}\sum_{l=2}^{i+1}B_0^l,\quad i=t,\ldots,N_v-1. \tag{19}$$

根据针对 MDP 的多阶段随机优化理论 [21], 式 (15) 中的问题可转化为如下的递推形式:

$$\begin{aligned}\min_{\bar{R}^t}\ &\bar{P}^t(\alpha^t,\bar{R}^t)+\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\Big\{\min_{\bar{R}^{t+1}}\bar{P}^{t+1}(\alpha^{t+1},\bar{R}^{t+1})\\&+\cdots+\mathbb{E}_{\boldsymbol{s}^{N_v-1}|(\boldsymbol{s}^{N_v-2},\bar{R}^{N_v-2})}\Big\{\min_{\bar{R}^{N_v-1}}\bar{P}^{N_v-1}(\alpha^{N_v-1},\bar{R}^{N_v-1})\Big\}\Big\},\\\text{s.t.}\ &\text{式 (15b), (15c), (19),}\end{aligned} \tag{20}$$

其中式 (19) 和 (15c) 中的约束限制了 $\bar{R}^i, i>t$ 的可行域. 第 i, $i>t$ 个时间步决策的可行域可以表示为

$$\mathbb{D}^i \triangleq \left\{ \bar{R}^i | B^i + \bar{R}^i \geqslant \frac{1}{\Delta T}\sum_{l=2}^{i+1} B_0^l, 0 \leqslant \bar{R}^i \leqslant \mathbb{E}_{\alpha^i|\boldsymbol{\alpha}_{T_o}^t}\{R_{\max}(\alpha^t)\} \right\},\ i>t, \tag{21}$$

则式 (20) 中的问题可以重新写为如下形式:

$$\begin{aligned} \min_{\bar{R}^t}\ & \bar{P}^t(\alpha^t, \bar{R}^t) + \mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\Big\{ \min_{\bar{R}^{t+1}\in\mathbb{D}^{t+1}} \bar{P}^{t+1}(\alpha^{t+1}, \bar{R}^{t+1}) \\ & + \cdots + \mathbb{E}_{\boldsymbol{s}^{N_v-1}|(\boldsymbol{s}^{N_v-2},\bar{R}^{N_v-2})}\Big\{ \min_{\bar{R}^{N_v-1}\in\mathbb{D}^{N_v-1}} \bar{P}^{N_v-1}(\alpha^{N_v-1}, \bar{R}^{N_v-1}) \Big\}\Big\}, \\ \text{s.t.}\ & B^t + \bar{R}^t \geqslant \frac{1}{\Delta T}\sum_{l=2}^{t+1} B_0^l, \ \text{式 (15b)}. \end{aligned} \tag{22}$$

与式 (20) 中的问题不同, 这个问题中的约束只与第 t 个时间步的 B^t 和 $\bar{R}^t$ 有关. 其中, 除第一项以外目标函数中的其他项为对未来多个时间步功耗预测值的求和. 根据 $\boldsymbol{s}^t$ 的定义, 这个求和项是 $\boldsymbol{s}^t$ 与 $\bar{R}^t$ 的函数, 故可引入如下符号来表示对未来多个时间步发射功耗预测值之和 (称为值函数):

$$\begin{aligned} Q(\boldsymbol{s}^t, \bar{R}^t) \triangleq & \mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\Big\{ \min_{\bar{R}^{t+1}\in\mathbb{D}^{t+1}} \bar{P}^{t+1}(\alpha^{t+1}, \bar{R}^{t+1}) \\ & + \cdots + \mathbb{E}_{\boldsymbol{s}^{N_v-1}|(\boldsymbol{s}^{N_v-2},\bar{R}^{N_v-2})}\Big\{ \min_{\bar{R}^{N_v-1}\in\mathbb{D}^{N_v-1}} \bar{P}^{N_v-1}(\alpha^{N_v-1}, \bar{R}^{N_v-1}) \Big\}\Big\}. \end{aligned} \tag{23}$$

把式 (23) 代入式 (22), 即可得到如下瞬时约束下的在线单步决策主动优化问题:

$$\begin{aligned} \min_{\bar{R}^t}\ & \bar{P}^t(\alpha^t, \bar{R}^t) + Q(\boldsymbol{s}^t, \bar{R}^t), \\ \text{s.t.}\ & B^t + \bar{R}^t \geqslant \frac{1}{\Delta T}\sum_{l=2}^{t+1} B_0^l, \ \text{式 (15b)}, \end{aligned} \tag{24}$$

其中在第 t 个时间步进行决策时所有需要预测的值都反映在值函数 $Q(\boldsymbol{s}^t, \bar{R}^t)$ 中.

3.3 在线单步决策主动优化的无监督深度学习

3.3.1 满足策略约束的无监督深度学习

由于条件分布 $\mathbb{P}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{\cdot\}$ 未知, 一般情况下 $Q(\boldsymbol{s}^t, \bar{R}^t)$ 没有显式表达式, 不能通过数值算法直接求解式 (24) 中的问题获得监督学习所需要的训练标签, 因此采用无监督深度学习来学习从 $\boldsymbol{s}^t$ 到 $\bar{R}^t$ 的映射. 为此, 首先把式 (24) 中有约束的优化问题转化为如下的主对偶问题:

$$\max_{\xi_g^t,\xi_c^t}\min_{\bar{R}^t}\ L^t(\bar{R}^t, \xi_g^t, \xi_c^t) + Q(\boldsymbol{s}^t, \bar{R}^t), \tag{25}$$

其中 $L^t(\bar{R}^t, \xi_g^t, \xi_c^t) = \bar{P}^t(\alpha^t, \bar{R}^t) + \xi_g^t(\bar{R}^t - R_{\max}(\alpha^t)) + \xi_c^t(\frac{1}{\Delta T}\sum_{l=2}^{t+1} B_0^l - B^t - \bar{R}^t)$ 是拉格朗日 (Lagrange) 函数, $\xi_g^t, \xi_c^t \geqslant 0$ 是拉格朗日乘子.

根据文献 [15] 中的证明, 式 (25) 中的参数优化问题可以等价地转化为如下的泛函优化问题:

$$\max_{\xi_g(\boldsymbol{s}^t),\xi_c(\boldsymbol{s}^t)}\min_{\bar{R}(\boldsymbol{s}^t)}\ \mathbb{E}_{\boldsymbol{s}^t}\{L^t(\bar{R}(\boldsymbol{s}^t), \xi_g(\boldsymbol{s}^t), \xi_c(\boldsymbol{s}^t))\} + Q(\boldsymbol{s}^t, \bar{R}(\boldsymbol{s}^t))\}, \tag{26}$$

其中 $\bar{R}(\boldsymbol{s}^t)$ 是策略, $\xi_g(\boldsymbol{s}^t)$ 和 $\xi_c(\boldsymbol{s}^t)$ 是乘子函数. 可以引入神经网络 $\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$, $\mathcal{P}_{\xi_g}(\boldsymbol{s}^t;\theta_{\xi_g})$ 和 $\mathcal{P}_{\xi_c}(\boldsymbol{s}^t;\theta_{\xi_c})$ 来分别近似策略和两个乘子函数, 每个神经网络的输入都是 $\boldsymbol{s}^t$, 其相应的输出分别为 $\hat{R}(\boldsymbol{s}^t;\theta_r)$, $\hat{\xi}_g(\boldsymbol{s}^t;\theta_{\xi_g})$ 和 $\hat{\xi}_c(\boldsymbol{s}^t;\theta_{\xi_c})$. 可采用随机梯度下降算法来寻找最优的神经网络参数, 迭代公式如下:

$$\theta_r \leftarrow \theta_r - \frac{\delta_r}{|\mathbb{B}|}\sum_{\boldsymbol{s}^t\in\mathbb{B}}\frac{\partial\hat{R}(\boldsymbol{s}^t;\theta_r)}{\partial\theta_r}\left(\frac{\partial\bar{P}^t(\alpha^t,\bar{R}^t)}{\partial\bar{R}^t}+\frac{\partial Q(\boldsymbol{s}^t,\bar{R}^t)}{\partial\bar{R}^t}+\hat{\xi}_g(\boldsymbol{s}^t;\theta_{\xi_g})-\hat{\xi}_c(\boldsymbol{s}^t;\theta_{\xi_c})\mid_{\bar{R}^t=\hat{R}(\boldsymbol{s}^t;\theta_r)}\right), \tag{27}$$

$$\theta_{\xi_g} \leftarrow \theta_{\xi_g} - \frac{\delta_{\xi_g}}{|\mathbb{B}|}\sum_{\boldsymbol{s}^t\in\mathbb{B}}\frac{\partial\hat{\xi}_g(\boldsymbol{s}^t;\theta_{\xi_g})}{\partial\theta_{\xi_g}}(R_{\max}(\alpha^t)-\hat{R}(\boldsymbol{s}^t;\theta_r)), \tag{28}$$

$$\theta_{\xi_c} \leftarrow \theta_{\xi_c} - \frac{\delta_{\xi_c}}{|\mathbb{B}|}\sum_{\boldsymbol{s}^t\in\mathbb{B}}\frac{\partial\hat{\xi}_c(\boldsymbol{s}^t;\theta_{\xi_c})}{\partial\theta_{\xi_c}}\left(B^t+\hat{R}(\boldsymbol{s}^t;\theta_r)-\frac{1}{\Delta T}\sum_{l=2}^{t+1}B_0^l\right), \tag{29}$$

其中 $\delta_r,\delta_{\xi_g},\delta_{\xi_c}$ 表示学习率, $\mathbb{B}$ 表示对 $\boldsymbol{s}^t$ 进行随机采样后, 一个批次的样本所组成的集合, 批量大小为 $|\mathbb{B}|$, $|\cdot|$ 表示集合中元素的个数.

3.3.2 学习值函数 $Q(\boldsymbol{s}^t,\bar{\boldsymbol{R}}^t)$

由于值函数没有表达式, 在迭代更新 θ_r 时不能直接得到 $Q(\boldsymbol{s}^t,\bar{R}^t)$ 关于 $\bar{R}^t$ 的梯度. 为了解决这一问题, 可以引入一个神经网络 $\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 来近似 $Q(\boldsymbol{s}^t,\bar{R}^t)$, 其输入为 $(\boldsymbol{s}^t,\bar{R}^t)$, 输出为 $\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$. 根据式 (23) 中值函数的定义, 不难推出

$$Q(\boldsymbol{s}^t,\bar{R}^t)=\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\left\{\min_{\bar{R}^{t+1}\in\mathbb{D}^{t+1}}\bar{P}^{t+1}\left(\alpha^{t+1},\bar{R}^{t+1}\right)+Q(\boldsymbol{s}^{t+1},\bar{R}^{t+1})\right\}. \tag{30}$$

式 (30) 与贝尔曼 (Bellman) 最优方程 [22] 的形式完全相同.

由于 $\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 是对 $Q(\boldsymbol{s}^t,\bar{R}^t)$ 的近似, 将其代入式 (30) 后所得到的等式两边的项 (即 $\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 和 $\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{\min_{\bar{R}^{t+1}\in\mathbb{D}^{t+1}}\bar{P}^{t+1}(\alpha^{t+1},\bar{R}^{t+1})+\hat{Q}(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta_q)\}$) 之间存在误差, 且 $(\boldsymbol{s}^t,\bar{R}^t)$ 取不同值时误差大小不同. 因此, 可以通过使这两项之间的统计距离最小来寻找最优的神经网络参数 θ_q:

$$\min_{\theta_q}\ d_{\boldsymbol{s}^t,\bar{R}^t}\left\{\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q),\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\left\{\min_{\bar{R}^{t+1}\in\mathbb{D}^{t+1}}\bar{P}^{t+1}(\alpha^{t+1},\bar{R}^{t+1})+\hat{Q}(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta_q)\right\}\right\}, \tag{31}$$

其中 $d_{\boldsymbol{s}^t,\bar{R}^t}(\cdot,\cdot)$ 表示两个随机变量之间的距离关于 $(\boldsymbol{s}^t,\bar{R}^t)$ 的均值.

式 (31) 中优化问题的内部嵌套了优化第 $t+1$ 个时间步决策的目标函数, 由于输入 $\boldsymbol{s}^{t+1}$ 时 $\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$ 的输出 $\hat{R}(\boldsymbol{s}^{t+1};\theta_r)$ 是第 $t+1$ 个时间步最优决策的近似, 所以, $\min_{\bar{R}^{t+1}\in\mathbb{D}^{t+1}}\ \bar{P}^{t+1}(\alpha^{t+1},\bar{R}^{t+1})+Q(\boldsymbol{s}^{t+1},\bar{R}^{t+1})\approx\bar{P}^{t+1}(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r))+\hat{Q}(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q)$, 把这一近似代入式 (31), 可以得到如下优化问题:

$$\min_{\theta_q} d_{\boldsymbol{s}^t,\bar{R}^t}\left\{\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q),\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\left\{\bar{P}^{t+1}\left(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r)\right)+\hat{Q}(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q)\right\}\right\}. \tag{32}$$

由式 (32) 可知, 求统计距离时同时需要第 t 个时间步和第 $t+1$ 个时间步神经网络 $\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 的输出, 如果采用同一个网络学习这两个时间步的函数 $Q(\boldsymbol{s}^t,\bar{R}^t)$ 和 $Q(\boldsymbol{s}^{t+1},\bar{R}^{t+1})$, 则训练过程不容易收敛. 为了提高训练过程的稳定性, 引入了另一个神经网络 $\mathcal{Q}'(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta'_q)$ 来近似 $Q(\boldsymbol{s}^{t+1},\bar{R}^{t+1})$, 这个网络的输出为 $\hat{Q}'(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta'_q)$, 结构与 $\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 相同, 且 $\mathcal{Q}'(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta'_q)$ 的参数更新与 $\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 不同步. 因此, 采用如下的方式更新 $\mathcal{Q}'(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta'_q)$ 中的模型参数:

$$\theta'_q \leftarrow (1-\omega)\theta'_q+\omega\theta_q, \tag{33}$$

其中 ω 是一个很小的超参数, 从而使得每一步迭代更新后的 θ_q' 与更新前相比变化不大.

下面以欧氏距离为例推导 θ_q 的更新方程, 若选择其他可导的距离函数, 也可以导出类似的更新方程. 之所以选择欧氏距离, 是因为它是度量距离最常用的函数. 当选取 $d_{\boldsymbol{s}^t,\bar{R}^t}(\cdot,\cdot)$ 为随机变量间欧氏距离关于 $(\boldsymbol{s}^t,\bar{R}^t)$ 的集平均时, 把 $\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q')$ 和 $\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 代入式 (32) 中问题可得

$$\min_{\theta_q}\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t)}\left\{\left\{\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)-\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\left\{\bar{P}^{t+1}\left(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r)\right)+\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q')\right\}\right\}^2\right\}.$$

这个问题的目标函数关于 θ_q 的导数为

$$\begin{aligned}
&2\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t)}\Bigg\{\{\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)-\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{\bar{P}^{t+1}(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r))\\
&\quad+\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q')\}\}\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}\Bigg\}\\
&=2\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t)}\Bigg\{\{\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)\}-\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{\bar{P}^{t+1}(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r))\\
&\quad+\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q')\}\}\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}\Bigg\} \qquad (34)\\
&=2\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t)}\Bigg\{\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{(\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)-\bar{P}^{t+1}(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r))\\
&\quad-\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q'))\}\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}\Bigg\} \qquad (35)\\
&=2\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t)}\Bigg\{\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\Bigg\{(\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)-\bar{P}^{t+1}\left(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r)\right)\\
&\quad-\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q'))\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}\Bigg\}\Bigg\} \qquad (36)\\
&=2\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t,\boldsymbol{s}^{t+1})}\left\{(\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)-\bar{P}^{t+1}(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r))-\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q'))\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}\right\}, \qquad (37)
\end{aligned}$$

其中式 (34) 和 (36) 分别是因为 $\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 和 $\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}$ 与 $\boldsymbol{s}^{t+1}$ 无关. 根据 $\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t)}\{\mathbb{E}_{\boldsymbol{s}^{t+1}|(\boldsymbol{s}^t,\bar{R}^t)}\{\cdot\}\}=\mathbb{E}_{(\boldsymbol{s}^t,\bar{R}^t,\boldsymbol{s}^{t+1})}\{\cdot\}$ 可得到式 (37).

在实际应用中, 式 (37) 中关于 $(\boldsymbol{s}^t,\bar{R}^t,\boldsymbol{s}^{t+1})$ 的联合分布可以采用经验均值进行估计, 因此, θ_q 的更新方程如下:

$$\begin{aligned}
\theta_q\leftarrow&\theta_q-\frac{\delta_q}{|\mathbb{B}|}\sum_{(\boldsymbol{s}^t,\bar{R}^t,\boldsymbol{s}^{t+1})\in\mathbb{B}}\big(\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)-P^{t+1}(\alpha^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r))\\
&-\hat{Q}'(\boldsymbol{s}^{t+1},\hat{R}(\boldsymbol{s}^{t+1};\theta_r);\theta_q')\big)\frac{\partial\hat{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)}{\partial\theta_q}. \qquad (38)
\end{aligned}$$

算法 1 训练过程

随机初始化 $\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$, $\mathcal{P}_{\xi_g}(\boldsymbol{s}^t;\theta_{\xi_g})$, $\mathcal{P}_{\xi_c}(\boldsymbol{s}^t;\theta_{\xi_c})$, $\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$ 和 $\mathcal{Q}'(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta'_q)$;
for 迭代轮次 $=1,2,\ldots$ **do**
  记录初始时刻的状态 $\boldsymbol{s}^0$;
  for 时间步 $t=1,\ldots,N_v-1$ **do**
    根据 $\hat{R}(\boldsymbol{s}^t;\theta_r)$ 进行当前时间步的决策;
    利用式 (4) 计算 $g_{\text{th}}^t(\bar{R}^t)$;
    for 时隙 $i=1,\ldots,N_s$ **do**
      根据式 (3) 在第 t 个时间步中的每个时隙给用户分配功率;
    end for
    测量并记录新的状态 $\boldsymbol{s}^{t+1}$. 把样本 $\boldsymbol{e}^t\triangleq[\boldsymbol{s}^t,\bar{R}^t,\boldsymbol{s}^{t+1}]$ 保存在经验池 $\mathcal{D}$ 中;
    根据式 (27), (28), (29) 和 (38) 来依次更新所有神经网络的模型参数;
    根据 $\theta'_q\leftarrow(1-\omega)\theta'_q+\omega\theta_q$ 来更新网络 $\mathcal{Q}'(\boldsymbol{s}^{t+1},\bar{R}^{t+1};\theta'_q)$ 的参数;
  end for
end for

3.3.3 在线学习单步决策的训练过程

在每个时间步 (如第 t 个时间步), 中心处理器收集一个样本 $\boldsymbol{e}^t\triangleq[\boldsymbol{s}^t,\bar{R}^t,\boldsymbol{s}^{t+1}]$, 并将其保存在一个经验池 $\mathcal{D}=[\boldsymbol{e}^1,\ldots,\boldsymbol{e}^t,\ldots]$ 中, 再从经验池中随机选取一个批次的样本集合 $\mathbb{B}$ 来同时更新 $\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$, $\mathcal{P}_{\xi_g}(\boldsymbol{s}^t;\theta_{\xi_g})$, $\mathcal{P}_{\xi_c}(\boldsymbol{s}^t;\theta_{\xi_c})$ 和 $\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$. 称 N_v-1 个时间步内的参数更新过程为一个迭代轮次 (episode). 所有神经网络的在线训练过程如算法 1 所示.

3.3.4 在线单步决策的复杂度分析

下面采用现有文献最常用的乘法次数来评估无监督深度学习在每个时间步在线决策的复杂度. 由于可以采用一个多项式拟合式 (4) 中的函数, 从而以很少的乘法次数计算 $g_{\text{th}}^t(\bar{R}^t)$, 并且式 (3) 的计算只包括一次乘法, 故在线决策的复杂度取决于利用神经网络 $\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$ 进行前向传播得到 $\hat{R}(\boldsymbol{s}^t;\theta_r)$ 的运算. 当采用一个 L 层, 每层 d 个神经元的多层感知机作为 $\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$ 时, 无监督深度学习每个时间步在线决策的复杂度为 Ld^2.

3.4 与 DDPG 的关系及对 DRL 的理解

由 3.2 和 3.3 小节的分析可见, 无监督深度学习也能够与 DRL 一样, 以在线的方式学习 MDP 问题. 如果优化问题没有约束, 则式 (27) 与 DDPG 中更新动作网络模型参数的公式相同; 当式 (32) 的距离采用欧氏距离时, 式 (38) 与 DDPG 中更新批评家网络模型参数的公式相同, 其中批评家网络所学习的值函数 $Q(\boldsymbol{s}^t,\bar{R}^t)$ 表示的是未来多个时间步性能预测值之和; 此时, 所提出的在线无监督深度学习与 DDPG 方法相同. 不过, 为了满足预测资源分配问题中的瞬时约束, DDPG 方法需要在奖励函数中引入带有超参数的惩罚项, 在场景变化后需要重新调参, 而无监督深度学习方法则可通过对策略网络和两个乘子网络进行联合训练自动满足约束.

根据在线无监督深度学习与 DDPG 的关系, 可见对于建模为 MDP 的预测资源分配问题, 可以通过建模主动优化问题, 并将其转化为在线单步决策问题来系统化地设计状态. 此外, 由前面的分析可见, 所提出的在线无监督深度学习与强化学习之所以具有信息预测能力, 是因为在状态中包含了观测窗内的历史数据 $\boldsymbol{\alpha}_{T_o}^t$, 即只要在状态里包含历史观测, DRL 就能以端到端的方式对信息预测和资源分配进行联合优化. 尽管前述分析以预测资源分配为例, 以上分析不难退化为非 MDP 问题. 因此, 对于预测波束成形等问题, 无需像文献 [16,17] 那样专门设计进行预测的 RNN 或 RNN 层.

表 1 超参数

Table 1 Hyper-parameters

	$\mathcal{P}_R(\boldsymbol{s}^t;\theta_r)$	$\mathcal{P}_{\xi_g}(\boldsymbol{s}^t;\theta_{\xi_g})$	$\mathcal{P}_{\xi_c}(\boldsymbol{s}^t;\theta_{\xi_c})$	$\mathcal{Q}(\boldsymbol{s}^t,\bar{R}^t;\theta_q)$
Number of neurons in hidden layers	$[300, 200]$			
Activation function in hidden layers	`ReLU`			
Activation function in output layer	`Sigmoid`	`Softplus`		`Linear`
Learning rate	$\delta_r = \delta_{\xi_g} = \delta_{\xi_c} = \frac{0.0001}{1+0.5i}$			$\delta_q = \frac{0.001}{1+0.5i}, \omega = 0.001$
Batch size	64			
Buffer size	10^6			

4 仿真结果

本节首先通过仿真验证无监督深度学习相对于 DRL 方法在状态选择方面的优势, 然后比较无监督深度学习和 DRL 在求解预测资源分配问题时的系统性能.

4.1 仿真参数和超参数

一个用户在经过多个基站的一条道路上移动, 相邻基站间的距离为 500 m. 基站最大发射功率为 46 dBm, 用户与所接入基站间的最小距离为 200 m. 路径损耗为 $35.3 + 37.6 \log 10(d)$ dB, 其中 d 表示用户与所接入基站间的距离. 带宽为 20 MHz, 噪声功率为 -95 dBm. 小尺度信道增益服从瑞利 (Rayleigh) 分布. 每个视频的播放时间为 60 s, 每个视频片段的大小为 8 Mb, 播放时间为 1 s. 每个时间步和每个时隙的持续时间分别为 $\Delta T = 1$ s 和 $\tau = 1$ ms. 当以上参数不同时, 所得到的结论相同.

每个神经网络都为多层感知机, 精调后的超参数如表 1 所示, 其中 i 表示迭代步数. 考虑到大尺度信道增益较小, 为了加快训练的收敛速度, 用 $\ln(\alpha^t) + 30$ 替代神经网络输入中的 α^t.

4.2 不同场景下状态选择的影响

本小节比较无监督深度学习和 DRL 方法在不同场景下的状态选择对两种现有 DRL 方法 —— DDPG 和双延迟深度确定性策略梯度算法 (twin delayed deep deterministic policy gradient, TD3) —— 的影响.

• DDPG [8]. 分别引入动作网络和批评家网络来学习策略 $\bar{R}(\boldsymbol{s}^t)$ 和值函数 $Q(\boldsymbol{s}^t, \bar{R}^t)$, 两个神经网络都为多层感知机. 精调后的超参数为: 动作网络包含神经元数分别为 300 和 200 的两个隐藏层, 输出层激活函数为 `Sigmoid` 函数. 批评家网络包含神经元数分别为 300, 200 和 200 的 3 个隐藏层. 所有网络的隐藏层激活函数为 `ReLU` 函数, 且每一层均采用批量归一化. 动作和批评家网络的学习率分别为 $\frac{0.00001}{1+i}$ 和 $\frac{0.0001}{1+i}$, 批量大小为 64. 为了满足 QoS 约束, 在奖励函数上引入带可调超参数的惩罚项.

• TD3 [23]. 与 DDPG 类似, 但与 DDPG 的区别是引入了如下技术以保证训练的稳定性: 一是采用两个批评家网络, 计算值函数时取两者中较小的值; 二是引入延迟更新, 即批评家网络更新多次后再更新动作网络. 其超参数与 DDPG 方法相同.

由 4.1 小节中的系统参数设置可知不同视频片段的大小相同, 可以视为常数, 所以设计状态 $\boldsymbol{s}^t$ 时可以不考虑 $B_0^2, \ldots, B_0^{N_v-1}$.

表 2 匀速运动场景下选取不同状态时 DDPG 所能达到的系统性能和收敛所需要的训练时间

Table 2 System performance and training time required for convergence of DDPG when selecting different states, where each user moves at constant velocity

	$\boldsymbol{s}^t=[B^t]$	$\boldsymbol{s}^t=[\boldsymbol{\alpha}_{T_o}^t]$			$\boldsymbol{s}^t=[\boldsymbol{\alpha}_{T_o}^t,B^t]$		
Length of observation window	–	$T_o=1$	$T_o=2$	$T_o=3$	$T_o=1$	$T_o=2$	$T_o=3$
System energy consumption (J)	14.61	14.15	14.12	14.01	11.84	11.85	11.83
Training time (h)	1.23	1.35	1.33	1.29	1.02	0.98	0.92
Total training time (h)	8.12 (the time for fine-tuning the hyper-parameters is not included)						

4.2.1 匀速运动场景

首先考虑用户在道路上以 15 m/s 的速度匀速运动, 此时很容易预测大尺度信道.

根据式 (16), 单步决策在线无监督深度学习的状态为 $\boldsymbol{s}^t=[\boldsymbol{\alpha}_{T_o}^t,B^t]$. 由于用户匀速运动, 根据任意两个时间步的大尺度信道增益即可推算出用户的运动速度, 进而准确预测用户未来的大尺度信道增益. 由 3.3.3 小节可知, 一个训练样本中包含两个时间步的状态 (即 $\boldsymbol{s}^t$ 和 $\boldsymbol{s}^{+1}$), 选取观测窗长 $T_o=1$ (即 $\boldsymbol{\alpha}_{T_o}^t=[\alpha^t]$) 就能使一个训练样本中包含两个时间步的大尺度信道增益, 因此在这个场景下无监督深度学习的状态为 $\boldsymbol{s}^t=[\alpha^t,B^t]$.

对于 DRL 方法, 需要通过试错的方法来设计状态, 具体过程如下. 在第 t 个时间步进行决策时, 可能影响决策的环境变量包括当前时间步的观测 $\boldsymbol{\alpha}_{T_o}^t$ 以及到当前时间步为止基站所传输的总数据量 B^t, 这些变量所构成的所有状态组合如下: $[B^t]$, $[\boldsymbol{\alpha}_{T_o}^t]$, $[\boldsymbol{\alpha}_{T_o}^t,B^t]$. 此外, 观测窗长 T_o 也可能影响决策.

由于选取不同状态时 DDPG 和 TD3 方法的系统性能和训练时间相近, 表 2 仅给出了在选取不同状态时 DDPG 所能达到的系统性能和收敛所需要的训练时间, 其中系统性能指的是基站传输整个视频所消耗的能量, 单位为焦耳 (J), 收敛指的是超过一定的迭代轮次后总能耗的下降不超过 1%, 考虑了 $T_o=1,2,3$. 训练时间在一个配备 Intel®-Core™-i7-8700K CPU 和 Nvidia-Geforce-RTX™-1080Ti GPU 的电脑上测得, 单位为小时 (h), 总训练时间指的是给定表 1 超参数的前提下尝试了所有状态组合后所需要的训练时间之和. 从表中可见, 当状态 $\boldsymbol{s}^t$ 选取为 $[\boldsymbol{\alpha}_{T_o}^t,B^t]$ 时, 在不同观测窗长下 DDPG 可以达到的性能相近且较好, 这是因为在匀速运动场景下观测窗长选取为 1 即能准确预测未来的大尺度信道增益. 因此, 可以选取维度最小的状态 $\boldsymbol{s}^t=[\alpha^t,B^t]$, 但为了得到合适的状态, 共需要 8 个多小时的训练时间. 以上不包括调节超参数的时间, 否则所需要的总训练时间远比 8 个小时长得多.

图 4 给出了无监督深度学习和两种 DRL 方法在状态选取为 $[\alpha^t,B^t]$ 时的学习曲线, 并与在假设未来大尺度信道增益已知的前提下通过文献 [2] 中提出的数值算法求解式 (2) 中问题得到的全局最优策略 (图标 “理想策略”) 进行比较. 此外, 为了评估小尺度信道在不同时隙之间独立同分布这个假设的影响, 图 4 还给出了无监督学习和理想策略在基于 Jakes 模型生成小尺度信道 (小尺度信道在不同时隙之间不独立) 时的性能. 从图 4 中可以看出, 当小尺度信道增益服从瑞利分布时, 无监督深度学习和两种 DRL 方法都可以收敛到与 “理想策略” 非常接近的性能, 且无监督深度学习的收敛速度略快. 当小尺度信道在不同时隙之间非独立同分布时, 理想策略的性能有所提高, 而无监督深度学习方法仍然可以收敛到近最优的性能.

4.2.2 变速运动场景

下面考虑大尺度信道较难预测的场景. 用户的初始移动速度为 15 m/s, 加速度服从均值为 0 方差为 0.3 m/s^2 的高斯分布, 用户速度的取值范围为 $10\sim20$ m/s.

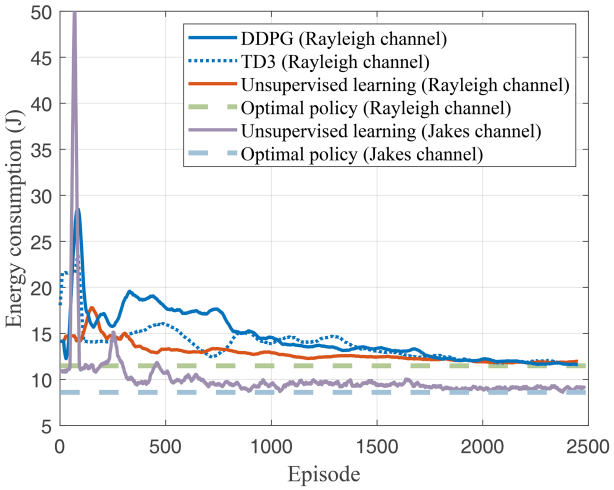

图 4 (网络版彩图) 无监督深度学习与 DRL 在 i.i.d (图标 "瑞利信道") 和非 i.i.d (图标为 "Jakes 信道") 信道下的学习曲线. Jakes 信道模型中生成信道的正弦波数为 8, 载频为 900 MHz, 最大多普勒频移为 91 Hz (采用其他参数时得到的结果类似)

Figure 4 (Color online) Learning curves of unsupervised deep learning and DRL under i.i.d channel distribution (with legend "Rayleigh channel") and non-i.i.d channel distribution (with legend "Jakes channel"). The number of sine waves used for generating Jakes channel is eight, the carrier frequency is 900 MHz, and the maximum Doppler frequency is 91 Hz (similar results can be obtained when using other parameters)

表 3 变速运动场景下 DDPG 选取不同状态时所能达到的系统性能和收敛所需要的训练时间

Table 3 System performance and training time required for convergence of DDPG when selecting different states, where each user moves at variable velocity

	$\boldsymbol{s}^t=[B^t]$	$\boldsymbol{s}^t=[\boldsymbol{\alpha}^t_{T_o}]$			$\boldsymbol{s}^t=[\boldsymbol{\alpha}^t_{T_o},B^t]$		
Length of observation window	–	$T_o=1$	$T_o=2$	$T_o=3$	$T_o=1$	$T_o=2$	$T_o=3$
System energy consumption (J)	48.84	20.12	17.03	19.34	18.97	13.34	11.73
Training time (h)	4.11	3.44	3.27	2.84	2.74	1.04	1.01
Total training time (h)	18.45 (the time for fine-tuning the hyper-parameters is not included)						

对于无监督学习, 可以由式 (16) 得到状态 $\boldsymbol{s}^t=[\boldsymbol{\alpha}^t_{T_o},B^t]$, 而 DDPG 和 TD3 的状态则需要采用试错的方式进行设计. 由于 DDPG 与 TD3 的训练时间和性能相近, 表 3 仅给出了选取不同状态时 DDPG 所达到的系统性能和所需的训练时间, 依然考虑了 $T_o=1,2,3$. 可以看出当状态选取为 $[\boldsymbol{\alpha}^t_{T_o},B^t]$ 且观测窗长 $T_o=3$ 时[1)], DDPG 所需的能耗最低. 然而, 为了得到合适的状态, 共需要 18 个多小时的训练时间. 当状态选取为 $[\boldsymbol{\alpha}^t_{T_o},B^t]$, $T_o=3$ 时, 无监督学习的系统性能与 DDPG 方法几乎相同, 但与匀速运动场景不同, 二者收敛后所需的能耗高于理想策略. 不同方法的收敛曲线与图 4 类似, 故不再给出.

4.3 系统性能评估

下面考虑 4.2.2 小节中的变速运动场景, 比较收敛后的无监督深度学习、DRL (以 DDPG [8] 为例)、"理想策略" [2] 及以下两种方法的系统性能.

- 非预测方法: 这是现有无线系统中所采用的不利用任何预测信息的资源分配方法. 为了满足 QoS

1) 对于所考虑的变速场景, 仿真表明在状态 $\boldsymbol{s}^t=[\boldsymbol{\alpha}^t_{T_o},B^t]$ 的情况下, 当观测窗长大于 3 时, 无监督学习和 DRL 方法的总能耗下降不超过 1%, 因此没有给出 $T_o>3$ 的结果. 若用户移动的随机性更强, 则需要更大的 T_o.

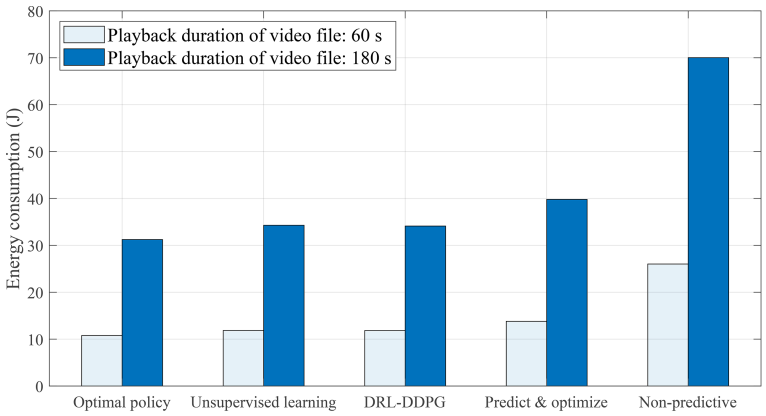

图 5 (网络版彩图) 不同方法的能耗

Figure 5 (Color online) Energy consumption of different methods

约束, 基站在每个时间步的平均数据率为 $\bar{R}^t = B_0^{t+1}$. 由于没有利用预测信息, 基站仅能通过 $\bar{R}^t = B_0^{t+1}$ 和式 (3) 在第 t 个时间步中的每个时隙给用户分配功率来最小化平均能耗.

• 先预测再优化 [2,5,6]: 首先用多层感知机预测未来时间步的信道增益, 此时观测窗长选为 20 才能达到良好的性能, 然后通过文献 [2] 中的数值算法把预测值当作真值求解式 (2) 中的优化问题.

无监督深度学习和 DDPG 的状态均为 $[\boldsymbol{\alpha}_{T_o}^t, B^t]$, 其中 $T_o = 3$. DDPG 方法通过在奖励函数上加惩罚项满足 QoS 约束, 当对所考虑的场景精调惩罚项上的超参数后, 收敛后不违背 QoS 约束的概率很低 (约为 0.003%). 无监督深度学习收敛后违背 QoS 约束的概率约为 3%, 但通过进行保守设计 (即在训练过程中令每个时间步的视频片段大小 $B_0^{t'} > B_0^t$) 就能以略微提升总能耗的代价使 QoS 约束以很高的概率得到满足. 例如, 当 $B_0^{t'} = 10$ Mb ($> B_0^t = 8$ Mb) 时, 违背 QoS 约束的概率约为 0.002%, 而总能耗的提升小于 2%.

图 5 给出了当视频播放时间分别为 60 s 和 180 s 时不同方法在以高概率满足 QoS 约束时传输视频所需的系统能耗. 可以看出无监督深度学习和 DRL 方法的能耗都小于先预测再优化方法, 而非预测方法性能最差.

5 总结

本文针对服务视频点播业务的预测资源分配问题提出了主动优化方法, 可以直接根据过去的大尺度信道增益以端到端的方式优化未来的决策变量. 通过提出在线求解单步优化预测资源分配的无监督深度学习方法, 可以系统化地设计 MDP 的状态并满足 QoS 约束, 分析了所提出的无监督学习方法与强化学习的联系. 仿真结果验证了理论分析, 表明无监督学习所达到的系统性能与 DRL 相近. 通过把影响所有用户决策的环境变量表示为状态, 所提出的无监督深度学习方法可以扩展到多用户场景. 通过把多智能体强化学习中每个智能体的优化问题转化为在线单步决策优化问题, 所提出的无监督学习方法也可以扩展到多智能体强化学习场景.

参考文献

1 Lu Z, de Veciana G. Optimizing stored video delivery for wireless networks: the value of knowing the future. IEEE Trans Multimedia, 2019, 21: 197–210

2 She C, Yang C Y. Energy efficient resource allocation for hybrid services with future channel gains. IEEE Trans Green Commun Netw, 2020, 4: 165–179
3 Atawia R, Hassanein H S, Ali N A, et al. Utilization of stochastic modeling for green predictive video delivery under network uncertainties. IEEE Trans Green Commun Netw, 2018, 2: 556–569
4 Xu W, Yang Z H, Ng D W K, et al. Edge learning for B5G networks with distributed signal processing: semantic communication, edge computing, and wireless sensing. IEEE J Sel Top Signal Process, 2023, 17: 9–39
5 Guo J, Yang C Y. Impact of prediction errors on high throughput predictive resource allocation. IEEE Trans Veh Technol, 2020, 69: 9984–9999
6 Bui N, Widmer J. Data-driven evaluation of anticipatory networking in LTE networks. IEEE Trans Mobile Comput, 2018, 17: 2252–2265
7 Zhang C Z, Guo J, Yang C Y. When the gain of predictive resource allocation for content delivery is large? Sci China Inf Sci, 2023, 66: 222302
8 Liu D, Zhao J Y, Yang C Y, et al. Accelerating deep reinforcement learning with the aid of partial model: energy-efficient predictive video streaming. IEEE Trans Wireless Commun, 2021, 20: 3734–3748
9 Meng L H, Zhang F Y, Bo L, et al. Fastconv: fast learning based adaptive bitRate algorithm for video streaming. In: Proceedings of IEEE GLOBECOM, 2019
10 Zhao B, Zhao X H. Deep reinforcement learning resource allocation in wireless sensor networks with energy harvesting and relay. IEEE Internet Things J, 2022, 9: 2330–2345
11 Shi Z Y, Xie X Z, Lu H B, et al. Deep reinforcement learning-based multidimensional resource management for energy harvesting cognitive NOMA communications. IEEE Trans Commun, 2022, 70: 3110–3125
12 Jang J, Yang H J. Deep learning-aided user association and power control with renewable energy sources. IEEE Trans Commun, 2022, 70: 2387–2403
13 Altman E. Constrained Markov Decision Processes. Boca Raton: CRC Press, 1999
14 Liu D, Sun C J, Yang C Y, et al. Optimizing wireless systems using unsupervised and reinforced-unsupervised deep learning. IEEE Network, 2020, 34: 270–277
15 Sun C J, Yang C Y. Learning to optimize with unsupervised learning: training deep neural networks for URLLC. In: Proceedings of IEEE PIMRC, 2019
16 Zhang J, Zheng G, Zhang Y, et al. Deep learning based predictive beamforming design. IEEE Trans Veh Technol, 2023, 72: 8122–8127
17 Chu M, Liu A, Lau V K N, et al. Deep reinforcement learning based end-to-end multiuser channel prediction and beamforming. IEEE Trans Wireless Commun, 2022, 21: 10271–10285
18 Attiah K M, Sohrabi F, Yu W. Deep learning for channel sensing and hybrid precoding in TDD massive MIMO OFDM systems. IEEE Trans Wireless Commun, 2022, 21: 10839–10853
19 Desset C, Debaillie B, Giannini V, et al. Flexible power modeling of LTE base stations. In: Proceedings of IEEE WCNC, 2012
20 She C Y, Yang C Y. Context aware energy efficient optimization for video on-demand service over wireless networks. In: Proceedings of IEEE ICCC/CIC, 2015
21 Hannah L A. Stochastic optimization. In: International Encyclopedia of the Social Behavioral Sciences. 2nd ed. Oxford: Elsevier, 2015. 473–481
22 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2019
23 Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: Proceedings of ICML, 2018

Predictive resource allocation: unsupervised learning of Markov decision processes

Jiajun WU, Jianyu ZHAO, Chengjian SUN & Chenyang YANG*

School of Electronics and Information Engineering, Beihang University, Beijing 100191, China
* Corresponding author. E-mail: cyyang@buaa.edu.cn

Abstract When future information of a mobile user such as trajectory is known, predictive resource allocation for video on-demand service can reduce energy consumption of base station or increase network throughput with ensured user experience. Traditional methods for predictive resource allocation first predict user information (say trajectory) and then optimize resource (say power) allocation. However, the prediction accuracy degrades as the prediction horizon increases. To deal with this issue, several recent works employed deep reinforcement learning for online decision-making by formulating the predictive resource allocation problem as Markov decision process (MDP). However, for this kind of MDP problems that is appropriately solved by reinforcement learning, existing works design the state in a trial-and-error manner. For constrained optimization problems, most existing reinforcement learning methods for wireless problems add penalty terms to the reward function with manually adjustable hyper-parameters to satisfy the constraints. This paper proposes an unsupervised deep learning method for online predictive resource allocation in an end-to-end manner, which can jointly predict information and optimize resource allocation. The proposed method is able to improve the performance of predictive resource allocation by online end-to-end unsupervised deep learning, and can systematically design the state of MDP and satisfy complex constraints such that the tedious trial-and-error methods for designing state and satisfying constraints are no longer necessary. We analyze the relationship between the unsupervised deep learning and deep reinforcement learning. Simulation results show that the proposed method needs almost the same energy consumption as deep reinforcement learning with a simplified state design process, which verifies the theoretical analysis.

Keywords predictive resource allocation, Markov decision process, unsupervised deep learning, deep reinforcement learning, state design, complex constraint