Universal Value Function Approximators

高辰潇

Nanjing University

December 8, 2020

目录

目录

概述

- Universal Value Function Approximators
- Tom Schaul, Dan Horgan, Karol Gregor, David Silver, ICML2015
- 核心思想
  - 通过低秩矩阵分解，将值函数 $V(s, g)$ 表示成两个分别和 s 和 g 相关的 embedding 向量 ϕ, ψ 的内积，借此解耦当前状态和目标状态（goal states）对值函数/奖赏的影响。
  - 通过解耦状态和目标，使得值函数在目标空间 $\mathcal{G}$ 上也能拥有一定的近似和泛化能力，在训练集数据受限/遇到未曾见过的目标状态时，也能取得一定的性能。

目录

术语

- MDP: $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{G}, \mathcal{R})$。
- 转移概率模型：$\mathcal{T}(s, a, s') := \mathcal{P}(s_{t+1} = s' | s_t = s, a_t = a)$。
- 目标状态集合：给定/人为设定的“目标”，其中包含着关于任务的知识。在本论文中，所考虑的目标均为状态空间的子集，即 $\mathcal{G} \subseteq \mathcal{S}$。
- 伪奖赏/目标相关奖赏：$R_g(s, a, s')$，用于替代原 MDP 问题中的奖赏，表示在目标为 $g \in \mathcal{G}$ 的子任务中，Agent 从 s 采取行动 a 转移到状态 s' 的奖赏。

术语 (cont'd)

- 广义值函数：对于任务 g，累积伪奖赏的期望

$$V_{g,\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} R_g(s_{t+1}, a_t, s_t) \prod_{k=0}^{t} \gamma_g(s_k) \middle| s_0 = s\right]$$

- 广义动作值函数

$$Q_{g,\pi}(s, a) := \mathbb{E}_{s'}\left[\sum_{t=0}^{\infty} R_g(s_{t+1,a_t,s_t}) + \gamma_g(s') V_{g,\pi}(s')\right]$$

其中 γ_g 一方面作为折扣因子，另一方面也作为 g 所代表的子目标的终止标志。

目录

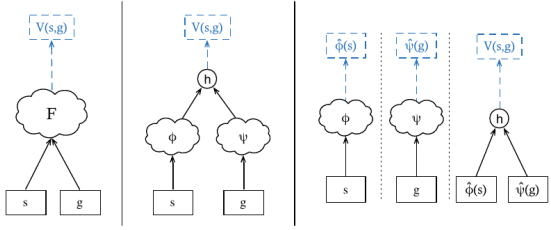
Architecture：Supervised Learning for UVFA

- 首先在监督学习的环境下探讨 UVFA 的学习过程：令 $V(s,g) = V_g^*(s)$
- *训练目标*：得到 value function approximator：$V(s,g;\theta) : \mathcal{S} \times \mathcal{G} \to \mathbb{R}$
- *训练数据*：部分状态 s、部分目标 g 下的广义状态值 $V(s,g)$

Architecture：Supervised Learning for UVFA

- Architecture 1：将 s 和 g 拼接成一个向量，使用 MLP 端到端地进行训练
- Architecture 2：将状态 s 和目标 g 分别输入 mapping functions 得到各自的 embeddings（$\phi(s)$ 和 $\psi(g)$），然后使用 output function h 组合得到输出 $h(\phi(s), \psi(g))$
  - 在本文中，使用的 output function 为 embeddings 的内积 $h(\phi, \psi) = \phi^\top \psi$
  - *partially symmetric*：由于 $\mathcal{G} \in \mathcal{S}$，使用结构 2 时可共享 ϕ 和 ψ 的某些网络层
  - *symmetric*：UVFA 函数本身可能是对称的，可通过使用相同的网络 $\phi = \psi$ 和使用对称的输出函数 h 来满足这一性质

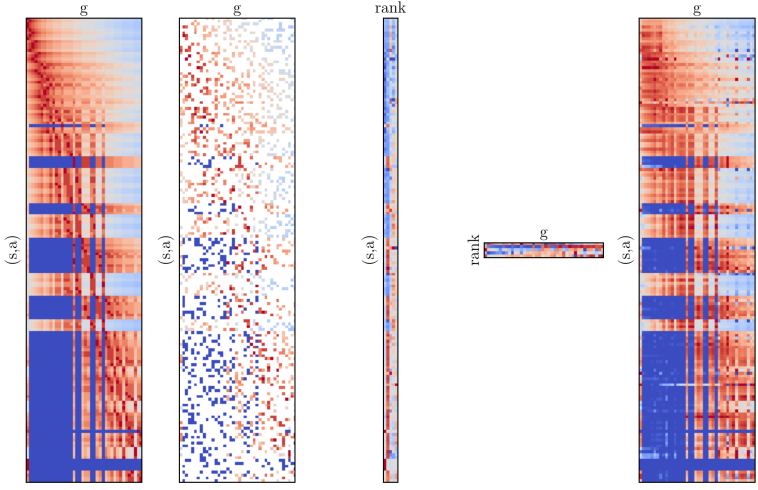
$$V^*_g(s) = V^*_s(g) \qquad \forall s, g$$



Supervised Learning for UVFA

- 对于 Architecture 2
- Step 1：将所有已知的 $V_g^*(s)$ 组织成矩阵的形式，每一行代表一个观测到的状态 s，每一列代表一个目标 g，得到矩阵 M
- Step 2：对 M 进行矩阵分解，找到一个低秩矩阵相乘的近似表示 $M = \hat{\phi}_s^\top \hat{\psi}_g$，其中 $\hat{\phi}_s$ 和 $\hat{\psi}_g$ 为状态和目标在 n 维空间中的嵌入
- Step 3：通过多变量回归等方式，调整网络 ϕ 和 ψ 的参数，使得 $\phi(s)$ 逼近 $\hat{\phi}_s$、$\psi(g)$ 逼近 $\hat{\psi}_g$。

Supervised Learning for UVFA



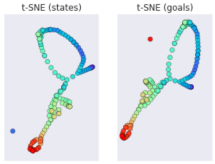
Experiments: Supervised learning for UVFA

- 实验环境
  1. 带有 4 个房间的栅格世界
  2. 岩浆世界
- $\mathcal{G} \subset \mathcal{S}$，伪奖赏函数被定义为

$$R_g(s, a, s') = \begin{cases} 1, & s' = g \text{ and } \gamma_{ext} \neq 0 \\ 0, & \text{otherwise.} \end{cases}$$

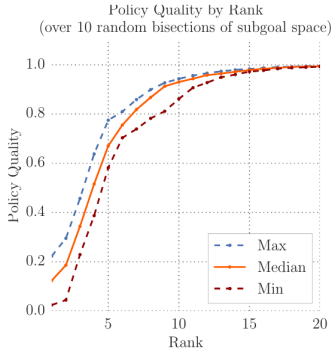
$$\gamma_g(s) = \begin{cases} 0, & s = g \\ 1, & \text{otherwise.} \end{cases}$$

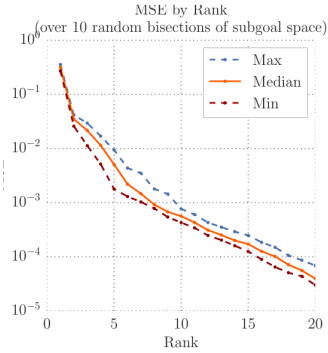
- 输入状态为指示每个 Grid 中填充物的二元向量的拼接



Experiments: Supervised Learning for UVFA

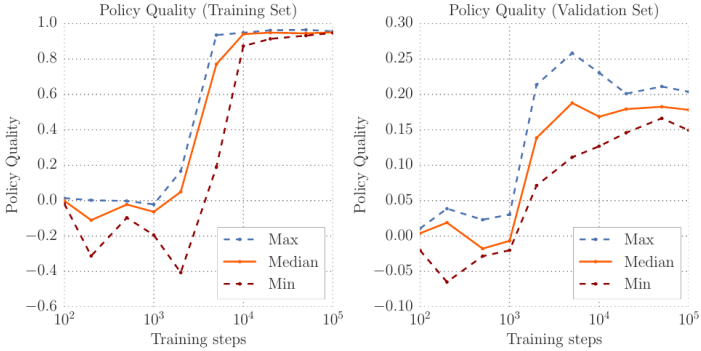
- 重建原始数据矩阵 M 的能力
  - 策略质量（最优策略的累积汇报为 1）
  - 估计误差





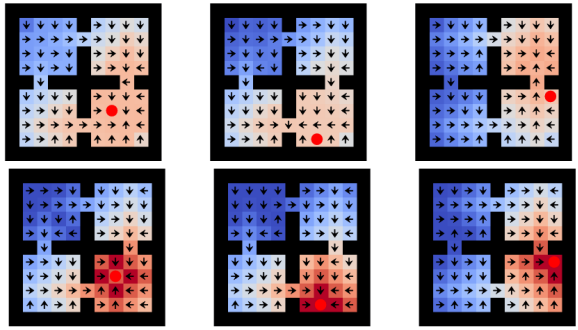
Experiments: Supervised Learning for UVFA

- **插值**
  - 训练集中包含所有状态和一半的目标集合
  - $n = 7$



Experiments: Supervised Learning for UVFA

- **外推**
  - 泛化到完全不同的空间中的目标上
  - 假设：位置空间中的所有状态 s 都包含在训练集的数据中
  - 与插值实验的不同之处：缺失数据的分布不同
- **实验结果**：$24\% \pm 8\%$



Combining UVFA with Reinforcement Learning

- 在具体的强化学习问题中，需要从给定的 transitions 中建立伪值函数/动作值函数
- **Horde 框架**
  - Off-Policy
  - Two-stream 结构

Algorithm 1 UVFA learning from Horde targets

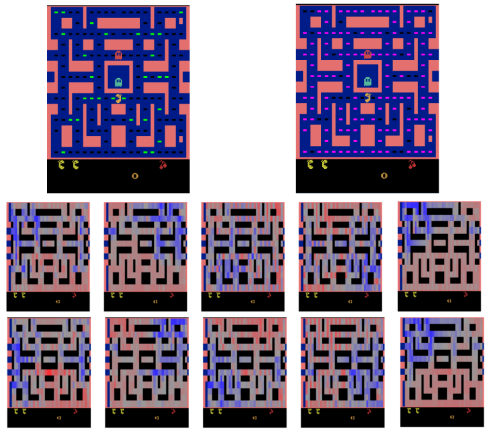
Input: rank n, training goals $\mathcal{G}_T$, budgets b_1, b_2, b_3
Initialise transition history $\mathcal{H}$
for $t = 1$ **to** b_1 **do**
  $\mathcal{H} \leftarrow \mathcal{H} \cup (s_t, a_t, \gamma_{ext}, s_{t+1})$
end for
for $i = 1$ **to** b_2 **do**
  Pick a random transition t from $\mathcal{H}$
  Pick a random goal g from $\mathcal{G}_T$
  Update Q_g given a transition t
end for
Initialise data matrix $\mathbf{M}$
for $(s_t, a_t, \gamma_{ext}, s_{t+1})$ **in** $\mathcal{H}$ **do**
  for g **in** $\mathcal{G}_T$ **do**
    $\mathbf{M}_{t,g} \leftarrow Q_g(s_t, a_t)$
  end for
end for
Compute rank-n factorisation $\mathbf{M} \approx \hat{\boldsymbol{\phi}}^\top \hat{\boldsymbol{\psi}}$
Initialise embedding networks ϕ and ψ
for $i = 1$ **to** b_3 **do**
  Pick a random transition t from $\mathcal{H}$
  Do regression update of $\phi(s_t, a_t)$ toward $\hat{\boldsymbol{\phi}}_t$
  Pick a random goal g from $\mathcal{G}_T$
  Do regression update of $\psi(g)$ toward $\hat{\boldsymbol{\psi}}_g$
end for
return $Q(s, a, g) := h(\phi(s, a), \psi(g))$

南京大学 NANJING UNIVERSITY

LAMDA Learning And Mining from DatA

Experiment on Ms Pacman

- Pacman 环境，每个 goal 对应地图中的“豆子”
- 使用环境中的 29 个 goal 构造数据矩阵 M，然后随机从剩下的目标中选取五个，可视化伪值函数
- 作为对比，中间一行是直接在所有 goal 上进行训练的 Horde 所得到的结果



目录

讨论

- Universal Value Function Approximators
- 将值函数分解为关于状态 s 和关于目标 g 的组合，以取得值函数在目标集合 $\mathcal{G}$ 上的近似能力和泛化能力
- **局限**
  - 目标集合 $\mathcal{G}$ 和伪奖赏函数 R_g 的值需要手动设定，其中融合了一定有关环境的知识
  - 仅考虑了目标集合 $\mathcal{G} \subset \mathcal{S}$ 的情况，并且目标 $\mathcal{G}$ 需要有一定的结构
  - 部分实验的假设过强
- **值得借鉴的思路**
  - 解耦当前状态和目标对值函数的影响，考虑使用更复杂的 $\mathcal{G}$，通过 UVFA 得到 embedding，从而在同一 MDP 下具有多个任务目标的强化学习问题中取得更好的表现。

references

- Horde 框架: Sutton, Richard S., et al. "Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction." The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2. 2011.