增强学习

Outline

1. 简介
2. 学习任务
3. Q 学习
4. 非确定性回报和动作
5. Temporal Difference Learning

Topic

1. 简介
2. 学习任务
3. Q 学习
4. 非确定性回报和动作
5. Temporal Difference Learning

增强学习（Reinforcement Learning ）

- 增强学习要解决的问题：一个能够感知环境的自治 agent，怎样学习选择能达到其目标的最优动作。
  - 学习控制移动机器人、在工厂中学习进行最优操作工序、以及学习棋类对弈等。
  - 当 agent 在其环境中作出每个动作时，施教者会提供奖赏或惩罚信息，以表示结果状态的正确与否。
    - 例如，在训练 agent 进行棋类对弈时，施教者可在游戏胜利时给出正回报，而在游戏失败时给出负回报，其他时候为零回报。
    - Agent 的任务就是从这个非直接的、有延迟的回报中学习，以便后续的动作产生最大的累积回报。
- Q 学习的算法：可从有延迟的回报中获取最优控制策略，即使 agent 没有有关其动作会对环境产生怎样的效果的先验知识。
- 增强学习与动态规划（dynamic programming）算法有关，后者常被用于解决最优化问题。

学习控制策略

- 学习控制策略以选择动作的问题在某种程度上类似于其他章讨论过的函数逼近问题。
- 这里待学习的目标函数为控制策略 $\pi : S \to A$ 。它在给定当前状态 S 集合中的 s 时，从集合 A 中输出一个合适的动作 a 。

延迟回报（delayed reward）

然而，增强学习问题与其他的函数逼近问题有几个重要不同：

- 延迟回报（delayed reward）
  - Agent 的任务是学习一个目标函数 π 。它把当前状态 s 映射到最优动作 $a = \pi(s)$ 。
  - 在前面章节中，我们总是假定在学习 π 这样的目标函数时，每个训练样例是序偶的形式 $\langle s, \pi(s) \rangle$ 。
  - 然而在增强学习中，训练信息不能以这种形式得到。
  - 相反，施教者只在 agent 执行其序列动作时提供一个序列立即回报值，因此 agent 面临一个时间信用分配（temporal credit assignment ）的问题：确定最终回报的生成应归功于其序列中哪一个动作。

探索（exploration ）

- 探索（exploration ）
  - 在增强学习中，agent 通过其选择的动作序列影响训练样例的分布。
  - 这产生了一个问题：哪种实验策略可产生最有效的学习。学习器面临的是一个折中的问题：
    - 是选择探索未知的状态和动作（以收集新信息），
    - 还是选择它已经学习过、会产生高回报的状态和动作（以使累积回报最大化）。

部分可观察状态（partially observable states）

- 部分可观察状态（partially observable states）
  - 虽然为了方便起见，可以假定 agent 传感器在每一步可感知到环境的全部状态，但在实际的情况下传感器只能提供部分信息。
  - 例如：带有前向镜头的机器人不能看到它后面的情况。在此情况下可能需要结合考虑其以前的观察以及当前的传感器数据以选择动作，而最佳的策略有可能是选择特定的动作以改进环境可观察性。

长期学习（life-long learning）

- 长期学习（life-long learning）
  - 不象分离的函数逼近任务，机器人学习问题经常要求此机器人在相同的环境下使用相同的传感器学习多个相关任务。
  - 怎样在窄小的走廊中行走，以及怎样从激光打印机中取得打印纸等。
  - 这使得有可能使用先前获得的经验或知识在学习新任务时减小样本复杂度。

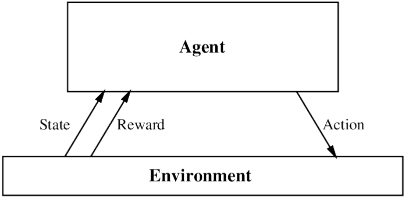
示例

- 建造一个可学习机器人。
  - 机器人（或 agent）有一些传感器可以观察其环境的状态（state）并能做出一组动作（action）改变这些状态。
  - 移动机器人具有镜头和声纳等传感器，并可以做出“直走”和“转弯”等动作。
- 学习的任务是获得一个控制策略（policy），以选择能达到目的的行为。
  - 此机器人的任务是在其电池电量转低时找到充电器进行充电。

One Example: TD-Gammon

- Tesauro, 1995
- Learn to play Backgammon
- Immediate reward
  - +100 if win
  - -100 if lose
  - 0 for all other states
- Trained by playing 1.5 million games against itself Now approximately equal to best human player

Reinforcement Learning Problem



$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \xrightarrow[r_2]{a_2} \ldots$$

Goal: Learn to choose actions that maximize

$r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots$, where $0 \leqslant \gamma < 1$

Topic

1. 简介
2. 学习任务
3. Q 学习
4. 非确定性回报和动作
5. Temporal Difference Learning

Markov Decision Processes

假设

- 有限状态集合 S
- 行动集合 A
- 在离散时间，agent 观测状态 $s_t \in S$ 选择行动 $a_t \in A$
- 接受即时回报 r_t
- 状态转换为 to s_{t+1}
- Markov 假设: $s_{t+1} = \delta(s_t, a_t)$, $r_t = r(s_t, a_t)$
  - r_t 与 s_{t+1} 只依赖当前状态与动作
  - 函数 δ 与 r 可以是非确定的
  - 对 agent 来说，函数 δ 与 r 可以是未知的

Agent's Learning Task

在环境中执行动作, 观察结果,

- 学习动作策略 $\pi : S \rightarrow A$，从 S 中的任意初始状态最大化

$$E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots]$$

- $0 \leq \gamma < 1$ 是未来回报的折算因子
- 目标函数是 $\pi : S \rightarrow A$
- 没有 $\langle s, a \rangle$ 形式的训练样例
- 训练样例形式为 $\langle \langle s, a \rangle, r \rangle$

Value Function

考虑确定世界
对每个策略 π ，定义评估函数

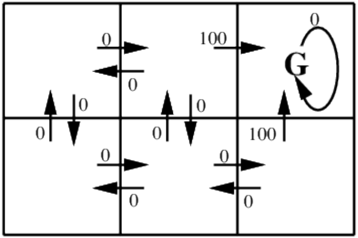
$$V^\pi(s) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$$
$$\equiv \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

其中 $r_t, r_{t+1}, \ldots$ 按策略 π 从状态 s 开始生成。任务是学习最优策略 π^*

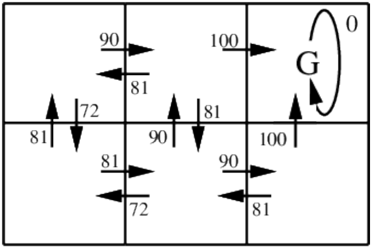
$$\pi^* \equiv \arg\max_\pi V^\pi(s), (\forall s)$$

A simple deterministic world to illustrate the basic concepts of Q -learning.

$r(s, a)$ (immediate reward) values

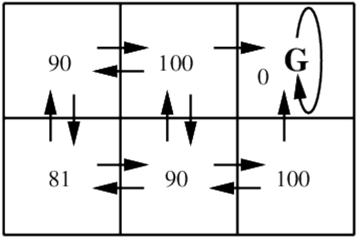


$Q(s, a)$ values

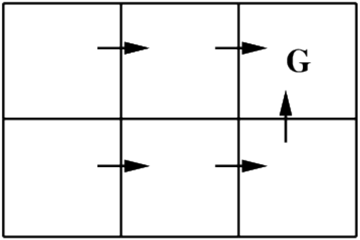


A simple deterministic world to illustrate the basic concepts of Q-learning.

$V^*(s)$ values



One optimal policy



What to Learn

学习评估函数 V^{π^*} (记作 V^*)
从任意状态 s 前瞻性搜索选择最优行动

$$\pi^*(s) = \arg\max_a [r(s,a) + \gamma V^*(\delta(s,a))]$$

问题：

-
- This works well if agent knows $\delta : S \times A \rightarrow S$, and $r : S \times A \rightarrow \Re$
-
- But when it doesn't, it can't choose actions this way

Topic

1. 简介
2. 学习任务
3. Q 学习
4. 非确定性回报和动作
5. Temporal Difference Learning

Q Function

与 V^* 类似定义新函数

$$Q(s, a) \equiv r(s, a) + \gamma V^*(\delta(s, a))$$

若 agent 学习 Q, 可以在不知道 δ 的情况下选取最优行动!

$$\pi^*(s) = \arg\max_a [r(s, a) + \gamma V^*(\delta(s, a))]$$

$$\pi^*(s) = \arg\max_a Q(s, a)$$

Q 是 agent 将要学习的评估函数

Training Rule to Learn Q

Q 与 V^* 有关:

$$V^*(s) = \max_{a'} Q(s, a')$$

Q 可以递归表示:

$$\begin{aligned} Q(s_t, a_t) &= r(s_t, a_t) + \gamma V^*(\delta(s_t, a_t)) \\ &= r(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') \end{aligned}$$

设 $\hat{Q}$ 表示当前对 Q 的逼近. 考虑训练规则

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

其中 s' 是在状态 s 应用行动 a 后得到的新状态

Q Learning for Deterministic Worlds

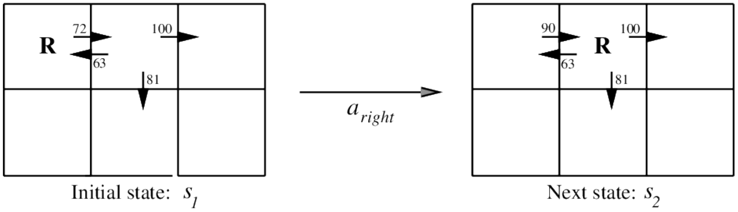
对每个 s, a 初始化 initialize table entry $\hat{Q}(s, a) \leftarrow 0$

- 对每个 s, a ，初始化表项 $\hat{Q}(s, a) \leftarrow 0$
- 观察当前状态 s
- 一直重复：
  - 选择一个动作 a 并执行它
  - 接收到立即回报 r
  - 观察新状态 s'
  - 对 $\hat{Q}(s, a)$ 按照下式更新表项：

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

  - $s \leftarrow s'$

Updating $\hat{Q}$



Updating $\hat{Q}$

$$
\begin{aligned}
\hat{Q}(s_1, a_{right}) &\leftarrow r + \gamma \max_{a'} \hat{Q}(s_2, a') \\
&\leftarrow 0 + 0.9 \ \max\{63, 81, 100\} \\
&\leftarrow 90
\end{aligned}
$$

若回报非负，则

$$(\forall s, a, n) \ \ \hat{Q}_{n+1}(s, a) \geq \hat{Q}_n(s, a)$$

$$(\forall s, a, n) \ \ 0 \leq \hat{Q}_n(s, a) \leq Q(s, a)$$

$\hat{Q}$ 收敛到 Q. 考虑确定世界，每个 $\langle s, a \rangle$ 无限频繁访问。

证明

在一个完全区间（full interval）（一个区间，其间每个 $\langle s, a \rangle$ 都被访问．），$\hat{Q}$ 表中的最大误差按因子 γ 减小。

证明 (续)

令 $\hat{Q}_n$ 为 n 次更新后的表，Δ_n 是 $\hat{Q}_n$ 中的最大误差，即：

$$\Delta_n = \max_{s,a} |\hat{Q}_n(s,a) - Q(s,a)|$$

对在第 $n+1$ 次更新的任意表项 $\hat{Q}_n(s,a)$, 在修正后的估计 $\hat{Q}_{n+1}(s,a)$ 中的误差为：

$$\begin{aligned}
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &= |(r + \gamma \max_{a'} \hat{Q}_n(s',a')) - (r + \gamma \max_{a'} Q(s',a'))| \\
&= \gamma |\max_{a'} \hat{Q}_n(s',a') - \max_{a'} Q(s',a')| \\
&\leq \gamma \max_{a'} |\hat{Q}_n(s',a') - Q(s',a')| \\
&\leq \gamma \max_{s'',a'} |\hat{Q}_n(s'',a') - Q(s'',a')| \\
|\hat{Q}_{n+1}(s,a) - Q(s,a)| &\leq \gamma \Delta_n
\end{aligned}$$

注：对任意两个函数，下式成立：

$$|\max_a f_1(a) - \max_a f_2(a)| \leq \max_a |f_1(a) - f_2(a)|$$

Topic

1 简介

2 学习任务

3 Q 学习

4 非确定性回报和动作

5 Temporal Difference Learning

Nondeterministic Case

回报与下一个状态是非确定性的
通过求期望重定义 V, Q by

$$V^{\pi}(s) \equiv E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots] \equiv E[\sum_{i=0}^{\infty} \gamma^i r_{t+i}]$$

$$Q(s,a) \equiv E[r(s,a) + \gamma V^*(\delta(s,a))]$$

Q learning generalizes to nondeterministic worlds

将 Q learning 推广到非确定性世界
将训练规则改为

$$\hat{Q}_n(s,a) \leftarrow (1-\alpha_n)\hat{Q}_{n-1}(s,a) + \alpha_n[r + \max_{a'} \hat{Q}_{n-1}(s',a')]$$

其中

$$\alpha_n = \frac{1}{1 + visits_n(s,a)}$$

仍可证明 $\hat{Q}$ 收敛至 Q [Watkins and Dayan, 1992]

Topic

1. 简介
2. 学习任务
3. Q 学习
4. 非确定性回报和动作
5. Temporal Difference Learning

Temporal Difference Learning

Q learning: 减小相继 Q 估计之间的不一致

单步时间差分:

$$Q^{(1)}(s_t, a_t) \equiv r_t + \gamma \max_a \hat{Q}(s_{t+1}, a)$$

两步

$$Q^{(2)}(s_t, a_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 \max_a \hat{Q}(s_{t+2}, a)$$

n 步

$$Q^{(n)}(s_t, a_t) \equiv r_t + \gamma r_{t+1} + \cdots + \gamma^{(n-1)} r_{t+n-1} + \gamma^n \max_a \hat{Q}(s_{t+n}, a)$$

混合多步:

$$Q^{\lambda}(s_t, a_t) \equiv (1-\lambda) \left[Q^{(1)}(s_t, a_t) + \lambda Q^{(2)}(s_t, a_t) + \lambda^2 Q^{(3)}(s_t, a_t) + \cdots \right]$$

Temporal Difference Learning

$$Q^{\lambda}(s_t, a_t) \equiv (1-\lambda)\left[Q^{(1)}(s_t, a_t) + \lambda Q^{(2)}(s_t, a_t) + \lambda^2 Q^{(3)}(s_t, a_t) + \cdots\right]$$

等效表达式:

$$\begin{aligned} Q^{\lambda}(s_t, a_t) &= r_t + \gamma[\quad (1-\lambda)\max_a \hat{Q}(s_t, a_t) \\ &\qquad\qquad + \lambda\ Q^{\lambda}(s_{t+1}, a_{t+1})] \end{aligned}$$

TD(λ) 算法使用上述训练规则

- 有时收敛比 Q learning 快
- 学习 V^* 对任意 $0 \le \lambda \le 1$ 收敛 (Dayan, 1992)
- Tesauro's TD-Gammon uses this algorithm

Subtleties and Ongoing Research

- Replace $\hat{Q}$ table with neural net or other generalizer
- Handle case where state only partially observable
- Design optimal exploration strategies
- Extend to continuous action, state
- Learn and use $\hat{\delta} : S \times A \rightarrow S$
- Relationship to dynamic programming