前言

之前的内容为无模型控制（Model-free control）做了一个铺垫。想象我们把一个智能体放入一个未知环境中，智能体该如何在该环境中不断改进自己的策略，以获得最大的未来回报，这正是Model Free Control要讨论的问题。

Introduction

Model Free Control除了可以解决MDP模型未知的问题，也同样适用于那些MDP模型已知，但是模型复杂度过高的问题。所以在这些问题中，通过采样的方式来获取经验，以不断优化智能体自身的策略。

由于采样也要靠执行策略，所以Model Free Control主要有以下两种方式：

- On-policy learning：采样和优化涉及的是同一个策略
- Off-policy learning：采样策略和优化策略不相同

这里可以结合后面的具体算法来理解。

On-Policy Monte-Carlo Control

在Model Based Control中就已经提出了一种通用的策略迭代（Generalised Policy Iteration）框架，他也适用于Model Free的情况。但是有几个关键点需要注意：

- 有MDP模型时，我们在评估一个策略时，具体是采用state-value function还是action-value function都不会有任何影响，因为我们有Bellman Equation可以用来转化；但是没有MDP模型，我们在Policy improvement采用Greedy Strategy时，为了保证其可行，在评估策略阶段只能采用action-value function
- ϵ-Greedy Policy：由于我们对于环境的未知，即没有MDP模型，为了保证探索和利用的平衡，我们需要保证所有的action都有一个非空的被选择的概率，最简单的方式就是以$1-\epsilon$概率选择满足贪心的动作，以ϵ概率随机选择一个动作。
- 为什么需要ϵ：其实一开始我一直在思考一个问题，为什么Model Based条件下，不需要ϵ，而是直接Greedy Policy Improvement就可以了，其实想想问题本质就很简单了：Model Free条件下，你是采样方式，如果在一开始就放弃一些在上一个策略影响下而看起来不那么优的action，那么你相当于在下一轮迭代中彻底放弃探索这个动作之后的路径；但是如果是Model Based条件，由于我们有MDP模型，我们在评估新的策略时，我们的更新还是会考虑到所有路径，即那些更优的动作还是会在新一轮中被发现。所以这也是为什么要把ϵ量的概率均分给每一个动作：在未知环境的条件下，我们不想错过任何一个在未来策略提升后可能比之前动作要更优的动作，这种可能性很复杂，但是事实上是存在的。
- Monte-Carlo Control：Monte-Carlo policy evaluation+ϵ-Greedy policy improvement
- Convergence：在谈收敛性之前，相信大家都有一个疑问：每一次采样需要到底多少episode才能保证收敛？或者说，我们真的需要无限条episodes来准确评估当前策略嘛？GLIE(Greedy in the limit with Infinite Exploration)理论给出了答案。只要最终可以保证Infinite Exploration就行，所以实际的算法每一轮采样一条episode即可。这也保证了我们在policy evaluation的时候，只需要保证$Q \approx q_\pi$

- GLIE Monte-Carlo Control Algorithm

On-Policy Temporal-Difference Learning

仔细回忆TD的一些优势，支持连续环境（continuing environments），方差更小等等，同理，我们可以得到On-Policy Temporal-Difference Learning Algorithm。

- Policy evaluation： Sarsa，$Q \approx q_\pi$
- Policy improvement： ϵ-Greedy policy improvement
- Convergence：GlIE + Robinns-Monro sequence of step size α_t
- Others：n-step Sarsa，Forward View Sarsa(λ)，Backward View Sarsa(λ)

Off -Policy Learning

Off-Policy Learning这件事情本身很好理解，值得关注的是他出现的理由：

- 模仿观察人类或其他智能体的策略来学习自身策略
- 重复利用很早以前策略的经验
- 学习最优策略但是却按探索性策略行动
- 学习多个策略时，遵照一个策略来行动

关于Importance Sampling，他适用于Monte-Carlo Control 和 TD Control。但是我们重点关注一下Q-learning.Q-learning的思想是你的下一个动作根据探索性的行为策略决定，但是你通过bootstrapping而利用的下一个状态的action-value中的动作则是由目标策略决定的：这样我们就可以实现探索和利用的完美平衡！而事实上他也是那样做的：

- 目标策略π是关于$Q(s,a)$的贪心策略
- 行为策略μ是关于$Q(s,a)$的ϵ-贪心策略

Q-learning和Sarsa是一组很好的对比对象来理解Q-learning的特性。