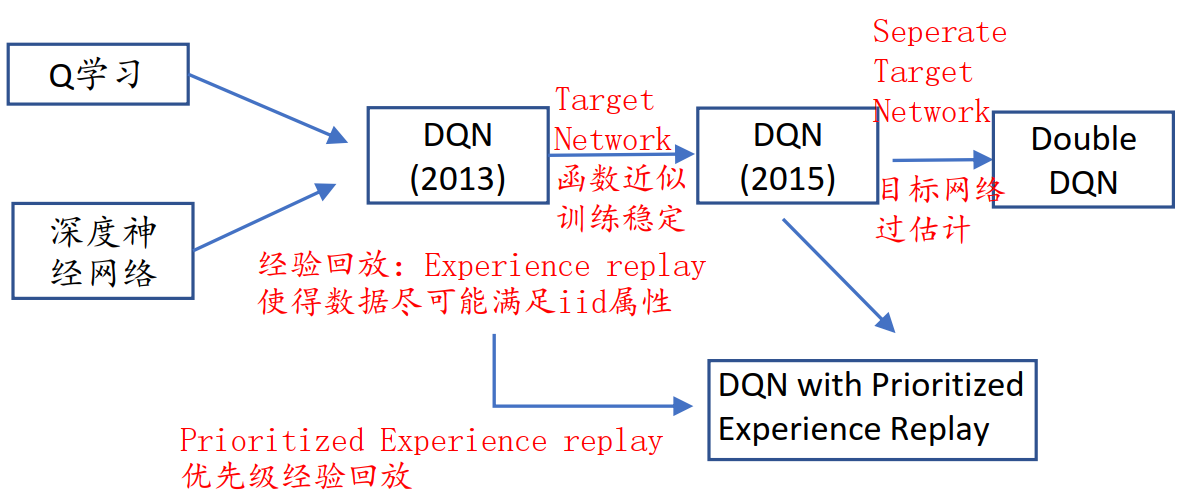
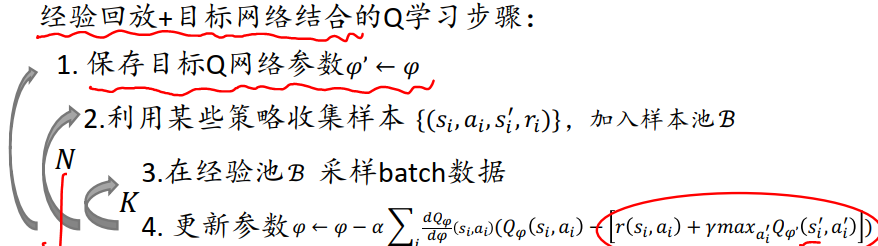
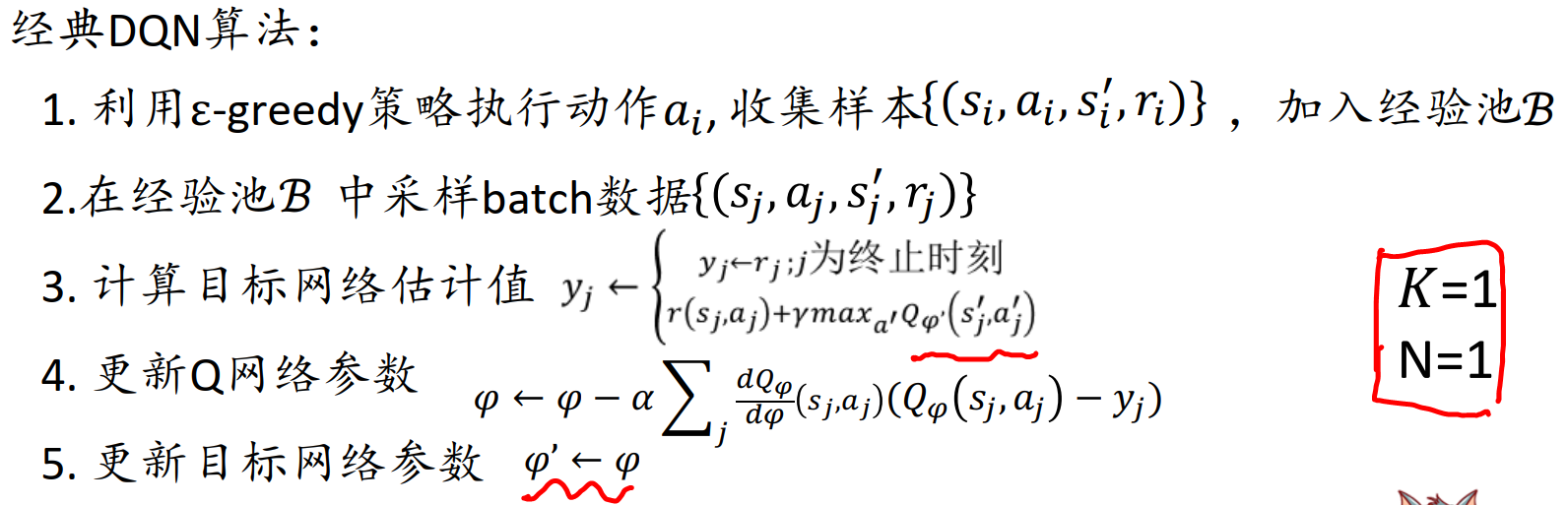
**深度强化学习:**使用深度神经网络作为强化学习的函数近似器。

传统在线Q学习问题：1，样本不满足iid条件，时间相关性很强；2，Q学习理论上无法证明收敛，因为更新并非梯度下降。

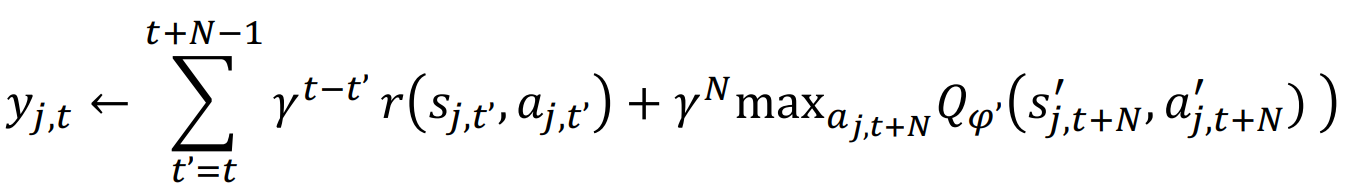




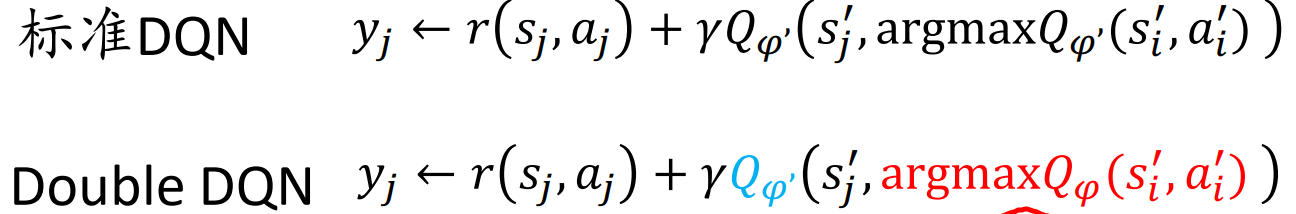
DQN:通过经验回放缓解了数据iid的问题, Batch训练降低方差,目标网络结合保证了训练的稳定性



DQN的多步回报版本，得到更低的方差：



DQN中Q值估计并不准确，TD目标值中的max操作，将引入一个正向的偏差，导致下一时刻的目标值存在过估计，解决思路：**Double DQN**使用不同的网络来分别计算目标Q网络值和选择动作；利用当前Q网络来选择动作；利用目标Q网络值来估计目标值。



DQN中经验池均匀采样的效率不高，不能反映样本的价值，Prioritized Experience Replay 就是维护了一个带优先级的经验回放，利用TD 误差去衡量优先级，TD误差大样本价值高，从而剔除价值低的样本。

**优先级经验回放带来的问题**：TD 误差对噪声敏感；TD 误差小的样本长时间得不到不更新；过分关注TD 误差大的样本，丧失了样本多样性。

**优先级确定与根据优先级采样**:优先级确定方法（TD误差），比例优先化：𝑝𝑖 =| 𝛿𝑖 | + 𝜖，通过加入一个小的噪音项，增加多样性，确保非零，排序优先化，;优先级采样概率, 随机优先化采样概率, 𝛼系数的选择决定了优先级的使用程度，𝛼=0为标准DQN均匀采样。

优先级高的样本具有更高利用率带来的“有偏”，引入重要性采样权重来平衡“有偏”问题。前期注重优先级高的样本的利用率后期注重无偏性，更新的无偏性是训练最后接近收敛最重要的部分。

**Dueling-DQN：**将Q函数分解成V函数和优势(A)函数，不依赖动作的值函数𝑉（s）依赖动作的优势函数𝐴（𝑠, 𝑎）

分解的原因：对于很多状态并不需要估计每个动作的值，增加了V函数的学习机会；V函数的泛化性能好，当有新动作加入时，并不需要重新学习；减少了Q 函数由于状态和动作维度差导致的噪声和突变

DQN训练技巧：算法确定后，首先在简单可信的任务上训练(比如Atrai)，确保你的配置是正确的；一般情况下，经验池越大，越有助于提升稳定性；收敛过程曲折震荡，需要保持耐心(有时性能比随机动作还糟)；初始epsilon值可以设置大一些，逐渐减小

**模仿学习**

**适用情况**

1.提供一系列演示轨迹：状态-动作序列  
2.当专家可以很容易地演示，同时演示轨迹很容易收集的时候，逆强化学习将非常适用。**模仿学习的目标**：智能体需要找到期望策略，使得该策略下状态-动作轨迹分布尽可能匹配专家演示样本的轨迹分布

**输入**：状态空间、动作空间；状态转移概率模型 𝑃（𝑠′|𝑠, 𝑎）；没有奖赏函数𝑅；专家演示样本集 𝑠0, 𝑎0, 𝑠1, 𝑎1, …

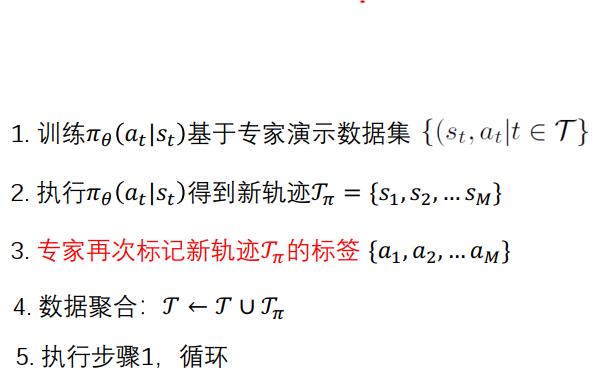
利用监督学习获得期望策略 (状态到专家动作的映射)输入为演示样本集中的状态，标签为演示样本集中的动作。

**行为克隆：**专家演示样本集,在专家演示样本集训练得到期望策略最小化误差

监督学习成功的基本假设：训练集和测试集的分布一致, 而行为克隆存在的问题：复合误差，最终导致分布漂移

**样本增广：**累计误差的解决办法

**数据聚合：**确保 𝑝𝑑𝑎𝑡𝑎(𝑠𝑡) ≈ 𝑝𝜋𝜃（𝑠𝑡）?由于无法直接改变𝜋𝜃（专家策略），通过重新标记当前策略下的新样本标签，添加至 𝑝𝑑𝑎𝑡𝑎确保专家轨迹分布与智能体轨迹分布尽可能接近。



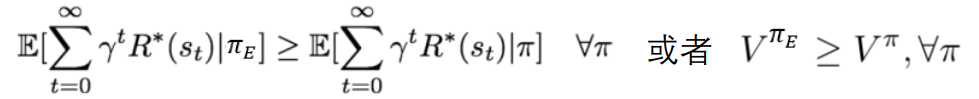
数据聚合的问题：1. 一直在执行不安全/部分训练的策略，2. 需要大量专家资源重新标记样本，同时与任务相关

模仿学习经常面临：数据分配不匹配；专家非马尔可夫行为；

专家多模态行为一些提升的办法：1. 增加左右侧输入图像数据2. 从一个稳定的轨迹分布中采样（用更稳定的控制策略采集样本）3. 增加on-policy的数据（数据聚合）,4改进模型以提升精度（RNN/LSTM等）5,需要足够多的数据

**逆强化学习**

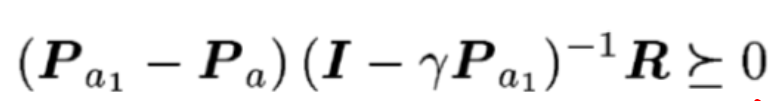
假设专家策略𝜋𝐸是最优的，从专家演示样本中恢复出奖赏函数𝑅∗



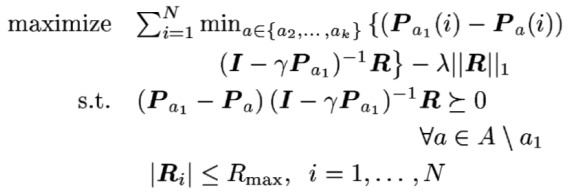
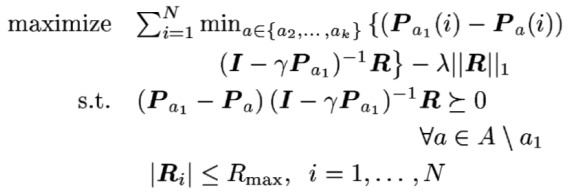
**最大边际形式化**

**最初IRL：**状态转移概率模型𝑃（𝑠′|𝑠,𝑎）已知，状态-动作空间为离散，假设专家策略𝜋𝐸是最优的，从专家演示样本中恢复出奖赏函数𝑅∗

a1取自被认为是最优的专家策略𝜋，需满足：



有限维离散状态空间IRL对应的优化问题: (让专家策略相比次好策略的优势尽可能大)

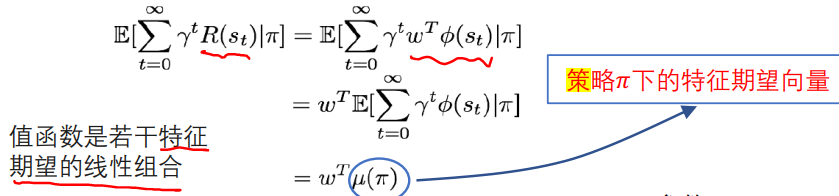
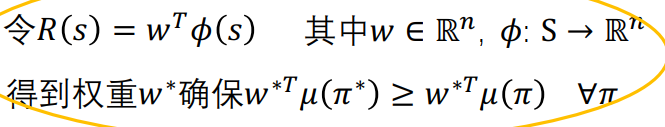
 对于有限维离散状态-动作空间情况，可以利用线性规划求解𝑅,对于连续状态空间的情况，难以直接求解，需要逼近求解𝑅

**学徒学习**

奖赏函数𝑅通过状态特征的线性函数逼近,

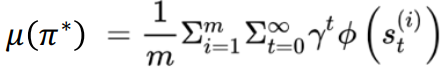
𝑅(s) = 𝑤𝑇𝜙(s) 𝑤 ∈ ℝ𝑛, 𝜙: S → ℝ𝑛  
𝜙(s)为定义的基函数

目标：基于专家演示样本训练权重w, 从专家演示样本中恢复出奖赏函数𝑅∗，使得专家策略𝜋𝐸为最优策略𝜋∗,和IRL上述一致

找到奖赏函数的参数使得专家策略的值函数优于其他策略，注意：𝑅（𝑠𝑡）是指状态𝑠𝑡下对应的𝑎𝑡的奖赏，也可以理解为上𝑅（𝑠𝑡, 𝑎𝑡）

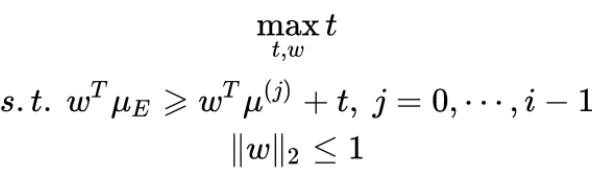
当给定m条专家轨迹后，可以估计专家策略的特征期望为：

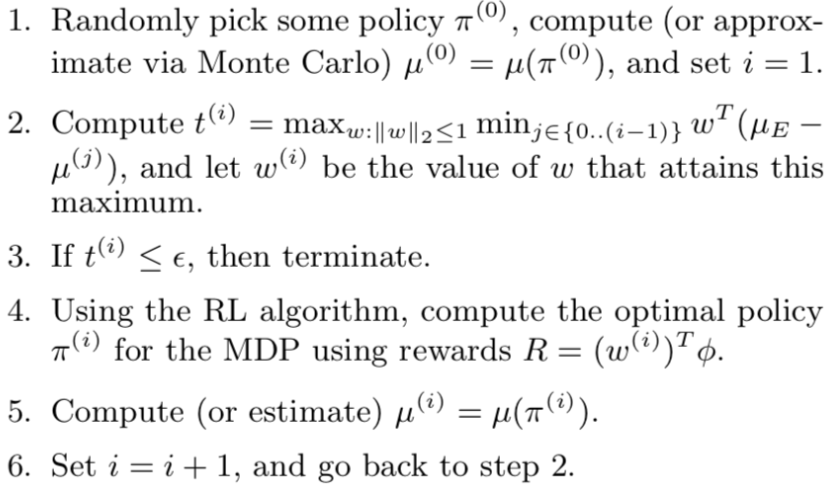


找到一个策略，使其表现与专家策略相近，其实就是找到一个策略的特征期望与专家策略的特征期望相近：



优化目标：





**学徒学习总结**;1.在已经迭代得到的最优策略中，利用最大边际方法求出当前的回报函数的参数值；2.将求出的回报函数作为当前系统的回报函数，并利用强化学习方法求出此时的最优策略。3.可以基于不同的专家演示数据学习到不同的风格

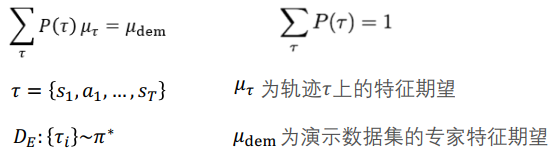
**最大边际方法中的“歧义”问题**

1.一个最优策略可能对应无穷个奖赏函数2.有无穷多个随机策略可以匹配特征数3.那么应该选择哪一个呢？存在随机的偏好

**基于概率模型的形式化**

从概率模型的角度分析：在概率论中，熵是不确定性的度量。不确定性越大，熵越大。在学习概率模型时，在所有满足约束的概率模型（分布）中，熵最大的模型是最好的模型最大熵方法避免了歧义问题，熵最大意味着概率分布越近似均匀分布。

假设存在一个概率分布𝑃(𝜏)，在该概率分布下，产生了专家轨迹，

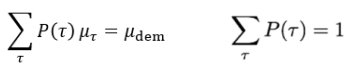


在满足上述约束条件的所有概率分布中， 熵最大的概率分布是除此约束外对其他未知情况不做任何主观假设的分布

**最大熵优化问题**

优化目标：

示例数据特征约束条件

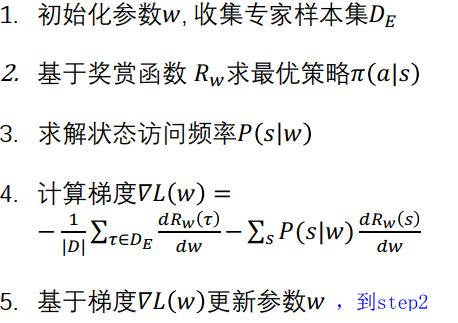


针对线性奖赏函数的情况，该优化问题等价于指定权重 𝑤 使得策略在匹配专家特征期望的同时，确保熵值最大化，消除歧义问题

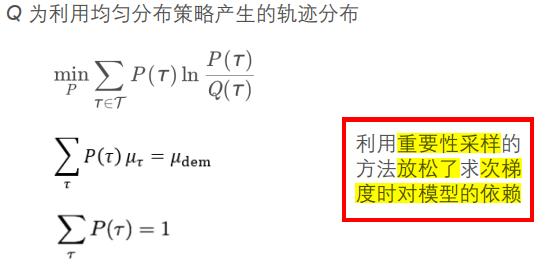
最大熵原理：在满足示例数据特征约束条件下，示例数据的分布熵最大化，意味着最大化示例数据关于回报函数参数的对数似然值。

策略轨迹的回报越大， 表示该轨迹被从专家数据集中采样的机率越大，轨迹选择实际上取决于未知的奖赏信号。

**最大熵逆强化学习：**



对于状态转移概率模型未知的情况，利用相对熵。



最大熵逆强化学习提供了在许多可能的奖赏函数中选择最优的途径，消除了最大边际IRL中的歧义问题，影响重大。

最大熵逆强化学习的不足：1. 假设奖赏函数可以表示为特征的线性组合，2. 要求转移概率模型已知

**价值函数逼近**

**大规模强化学习使用逼近器好处**:计算复杂度:使用更少的参数逼近价值函数 (而不是对每个状态, 动作存储价值)；泛化能力: 更少样本就能学到较精确的价值函数(而不是遍历整个状态, 动作空间)多样性:多种特征表示和逼近器结构(选择针对问题属性的逼近器)

价值函数逼近的类型：

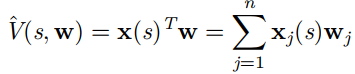
1.V 函数逼近:输入 s, 输出 V(s)  
2.Q 函数逼近:输入 (s, a), 输出 Q(s, a)  
3.Q 函数逼近:输入s,输出所有的 Q(s, a), ∀a

**线性函数逼近:**

用特征向量 (Feature Vector)代表某一状态:

x(s)=[x1(s), x2(s), x3(s)…xn(s)]T

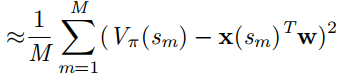
用一组特征的线性组合逼近价值函数:



均方误差(MSE) 描述逼近函数和真实价值函数之间的误差(期望损失)

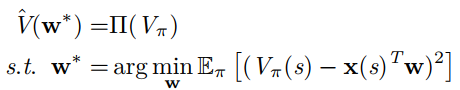
d 代表了智能体在策略 π 下的状态分布

对线性逼近器, 用样本近似期望损失:

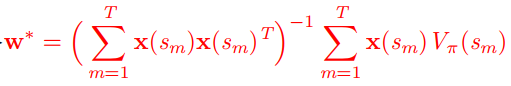


我们统一用 Π 代表从f(x)函数空间到 g(x|w) 函数空间的最佳匹配 (best fit)

线性逼近器的最佳匹配满足：



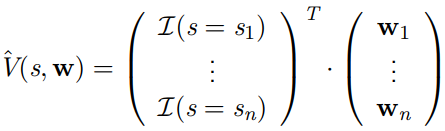
求解闭解：



最小二乘法优点: 计算过程简单, 一次性求解最佳权重;但是需要对 n × n 的矩阵求逆, 计算复杂度是 O(n3);当特征空间比较大时, 计算量大, 矩阵可能不满秩, 易有误差

**常见的特征表示方法**

**查表法**：可以看作是线性价值函数逼近器的一种特例，每个状态对应一个权重 wj 代表该状态的价值



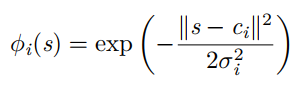
**离散化法**：连续的空间 S 划分成 非重叠的, 相邻的子空间 {Si}，状态在同一个子空间上具有相同的价值。(相似于上式中，区别是)

**粗糙编码**：同样是考虑连续状态空间，一个特征对应空间中的一个圆形区域，但是特征可以是可以重叠的.

*I(s is in circle i)*

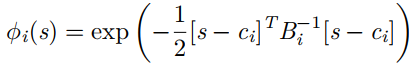
改变圆圈的数量,改变圆圈的范围,甚至改变圆圈的形状, 都能获得不同的特征表示

**径向基函数(RBF):** 是粗糙编码向连续型特征表示的泛化特征值可以是 [0,1] 之间的任意值, 代表属于该特征的程度,最典型的 RBF是高斯函数



ci 和 σi 分别对应高斯的中心点和宽度

多维状态的高斯函数:



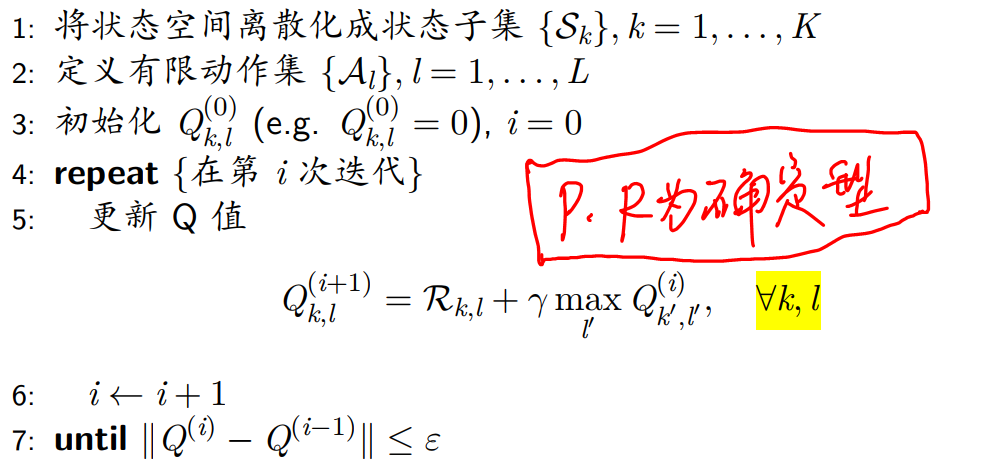
ci 是中心向量, Bi 是协方差矩阵

定义一组 RBF {(ci, Bi)}, 组成特征向量

可进行归一化，比上配分函数

**价值迭代 + 离散化方法**

求解连续空间 MDPs 问题，需要将连续的状态空间离散化成有限状态子集和有限动作集，对连续空间 MDPs, 奖励函数和转移函数通常是确定型的，因此离散化的价值迭代变成

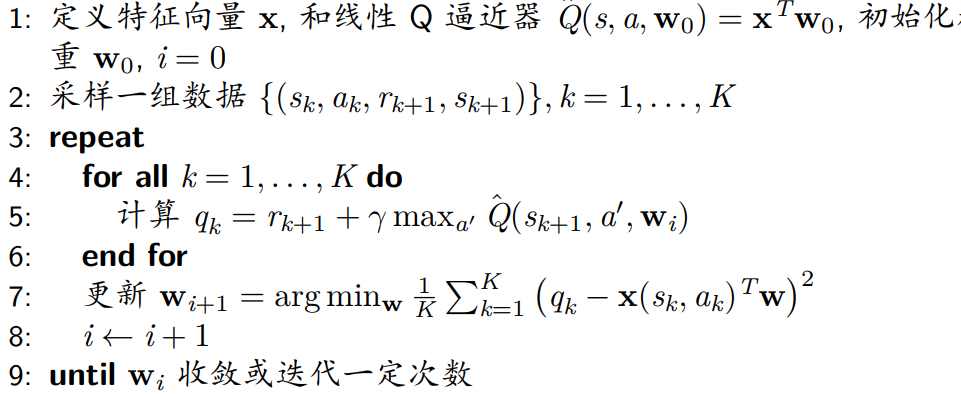


**离散化方法的优缺点**：离散化的价值迭代依然具有收敛性；离散化程度越高, 结果越精确；但是存储空间和计算量随离散集数量增加而增加；算法复杂度随维数呈指数增长, 维数灾的困境

**Fitted Q Iteration（Q\_learning）**

在代入到价值迭代算子后, 将结果重新映射回逼近器空间:

ΠT在无穷范数上是 γ-收缩算子，使用这种逼近器FQI能够收敛，但收敛的结果不一定等于最优Q函数（由于逼近误差的存在）

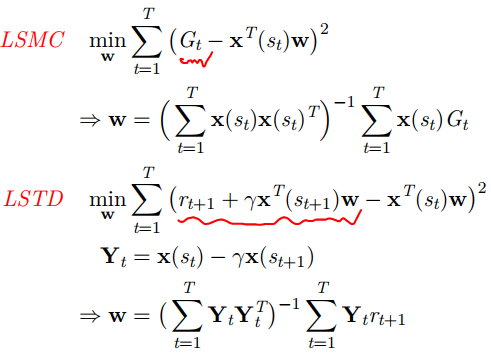
Least-Squared FQI用样本 (rk+1,sk+1)近似奖励和转移函数R,P.用数据集 {(sk, ak)} 近似状态和动作空间，如果事先能够采样到这样一组样本, 就可以无模型的强化学习

**策略迭代 + 最小二乘（闭解）**

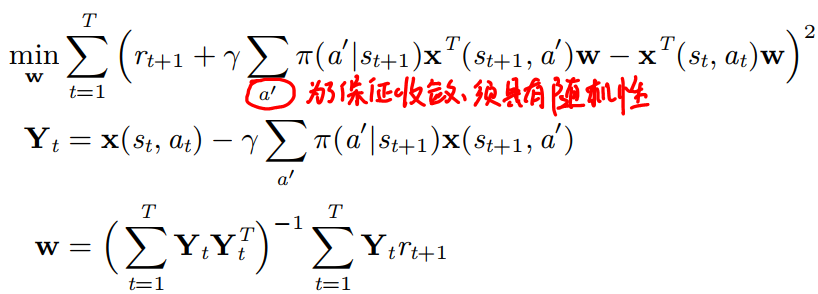
把linear approximation+least-squared 用在策略评估上,定义策略 π 的线性价值逼近器:

为避免对模型的依赖,对策略π评估Q函数:

最小二乘确定最佳匹配权重:

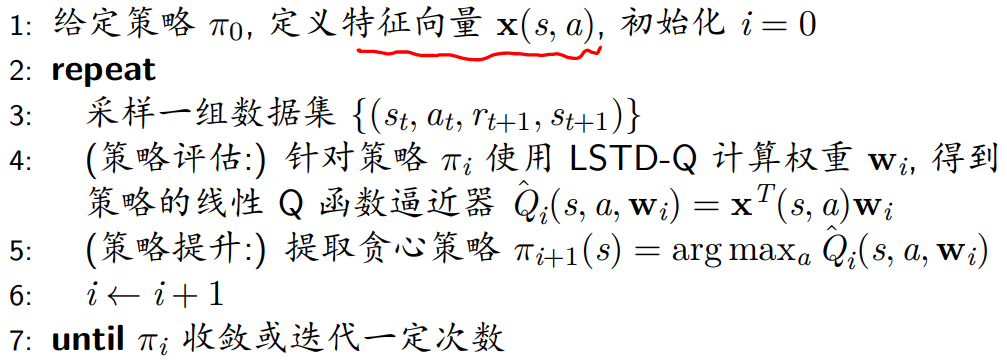


**对LSTD – Q：**



如果π是确定策略，需要具有探索性的动作,提高逼近器的泛化能力，如果π是随机策略,并且 π(a|s) > 0，可以at ∼ π(st)

**最小二乘策略迭代 LSPI：**

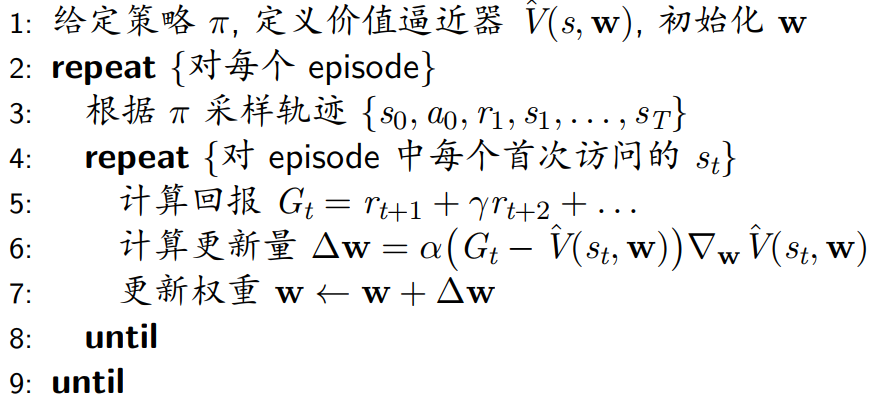


**approximate VI/PI 优缺点**

共同的优点: 和收缩逼近器结合后迭代依然是收敛的;共同的缺点:每次迭代的计算量巨大,线性最小二乘法求样本特征矩阵的逆;通常 PI 的迭代次数要比 VI 迭代次数少, 因此 approximate PI 整体计算量比VI小.

**预测学习 + 随机梯度下降法**

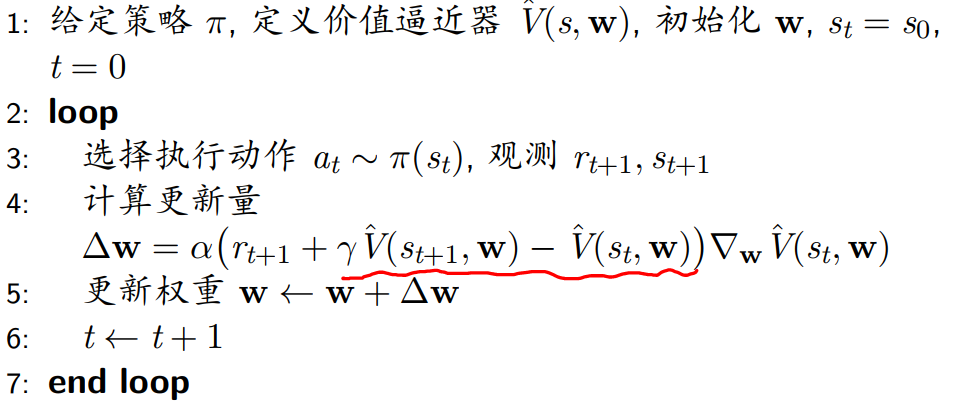
根据策略生成的轨迹使用梯度下降更新逼近器的权重, 线性和非线性逼近器都适用.

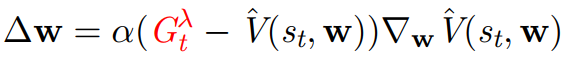
**梯度下降 MC 预测算法**

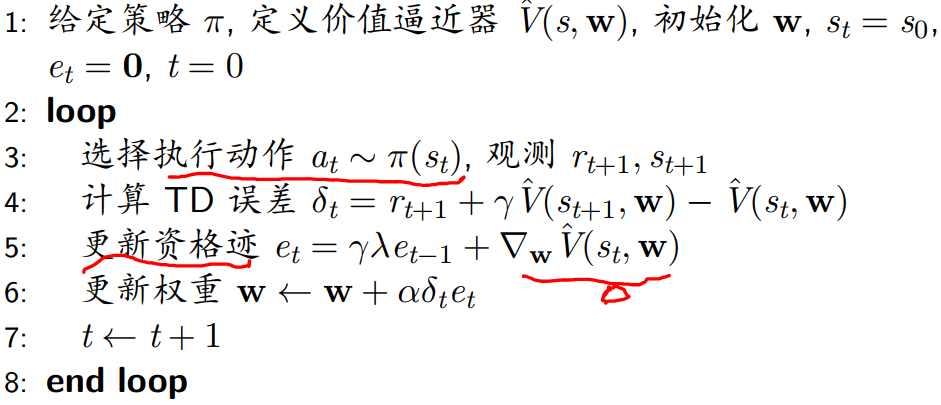
对TD目标变成是真实Vπ 有偏且近似的估计。

n-步TD学习和逼近器结合，前向算法使用λ-回报，后向算法利用资格迹，λ-回报同样是真实 Vπ 的有偏估计，可以根据策略 π 的轨迹对逼近器进行前向 TD(λ) 更新，基于逼近器的后向 TD(λ) 更新，资格迹向量et ,存储的是累加逼近器梯度。

梯度下降 TD 预测学习，

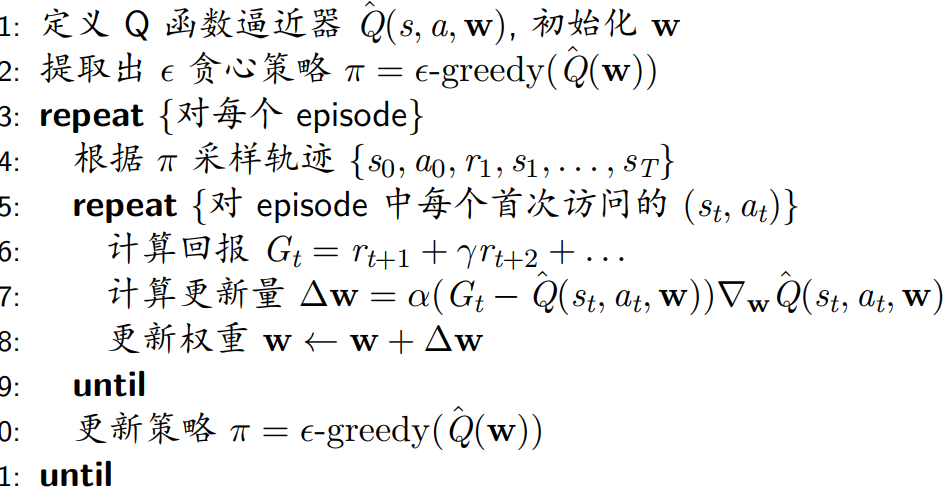


对于前向 TD(λ)：上式为梯度下降 TD(λ)（后向）算法：

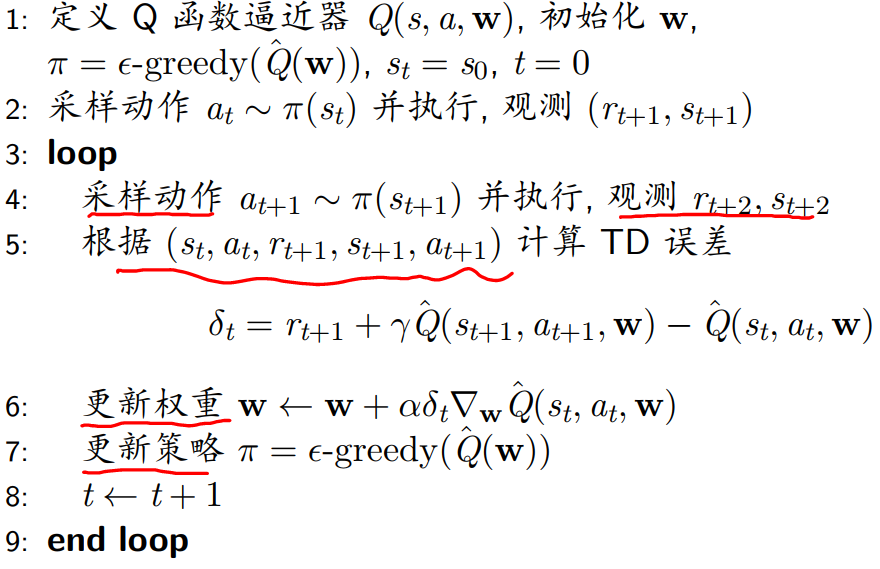
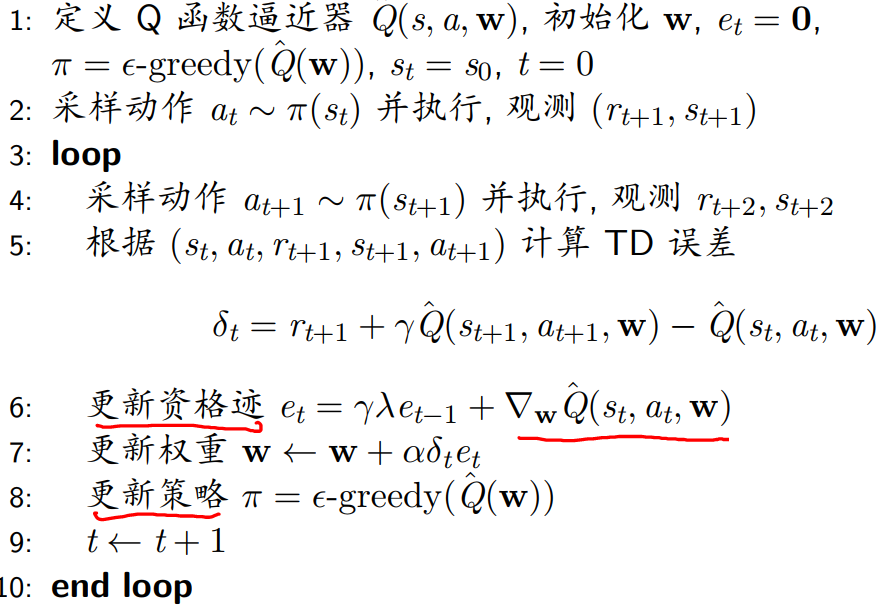


**控制学习 + 随机梯度下降法**

**梯度下降 MC 控制算法**



**梯度下降 Sarsa 算法**

**梯度下降 Sarsa(λ) 算法****梯度下降的 Q 学习算法**

