Reinforcement Learning

Arne Huckemann

CONTENTS

Contents **3**

1 Stochastic Bandits **5**

1 Two stage stochastic experiments . . . 5
2 Introduction of stochastic Bandits . . . 6
3 Algorithms: the exploration-exploitation trade-off . . . 14
3.1 Basic committ-then-exploit algorithm . . . 14
3.2 Purely Greedy . . . 18
3.3 ϵ-greedy bandit algorithms . . . 19
3.4 UCB algorithm . . . 21
3.5 Boltzmann exploration . . . 26
3.6 Policy Gradient for Stochastic Bandits . . . 29

2 Markov Decision Model **35**

1 Markov Decision Problem . . . 35
1.1 Markov-Chains . . . 35
1.2 Markov-Reward-Chains . . . 35
1.3 Markov-Decision Process . . . 37
1.4 Stochastic Control theory . . . 45
2 Basic tabular Value Iteration . . . 57
3 Basic Policy Iteration (Actor Critic) algorithm . . . 61
3.1 Policy Evaluation . . . 61
3.2 Policy Improvement . . . 65
3.3 Policy Iteration algorithms (tabular actor critic) . . . 69
3.4 Quick comparison of Value iteration and Policy Iteration . . . 70

3 Simulation based dynamic Programming methods **75**

1 A first example . . . 76
2 Monte Carlo Policy evaluation and control . . . 76
2.1 First Visit Monte Carlo Policy Evaluation . . . 77
2.2 Generalised policy iteration with first visit Monte Carlo estima- tion . . . 80

4 Finite Time MDPs **81**

1 Stochastic Fixed Point Iterations . . . 82
2 Proof of the general stochastic fixed point iteration . . . 90
2.1 Proof of boundedness . . . 90
2.2 Proof of convergence . . . 100
3 Sample based dynamic Programming . . . 103
3.1 Sample based policy evaluation algorithms . . . 105
3.2 Q-learning and the SARSA trick . . . 108

3.3 Others: Deep Q and alternate interpretation . . . 116
5 Gradient Descent Methods 119
0.1 Gradient descent for L-smooth functions . . . 120
0.2 Gradient descent for L-smooth convex functions . . . 122
0.3 Gradient descent for L-smooth functions with PL inequality . . . 126
0.4 Gradient descent with diminishing step-sizes . . . 129
0.5 Stochastic gradient descent . . . 132
1 Policy gradient theorems . . . 137
1.1 Finite-time MDPs . . . 137
1.2 Infinite-Time MDPs with discounted rewards . . . 143
1.3 Convergence of REINFORCE . . . 150
1.4 Variance Reduction methods . . . 150
6 Important Exercises (29) 153
7 Deep Reinforcement Learning 161
1 Neural Networks . . . 161
2 Distributional Reinforcement Learning . . . 161
3 Deep Q-Networks . . . 162
4 Proximal Policy Gradient . . . 166
5 Advantage Estimation . . . 166
6 Variants to PPO . . . 166
7 Stable Baselines3 . . . 166
A 5 Minuten Über Fraktale der Value function 167
1 Paar Ideen . . . 169

CHAPTER 1

STOCHASTIC BANDITS

1 Two stage stochastic experiments

Reinforcement learning can be viewed as a two stage experiment.

- Choose an action according to some distribution (first experiment)
- Given that action, observe a reward according to some distribution (second experiment)

In most applications the actions are discrete, but rewards are often not discrete.
Let (X, Y) be some pair of random variables on some probability space $(\Omega.\mathcal{F}, \mathbb{P})$. Then rewriting the joint distribution we have

$$\mathbb{P}(X = x, Y = y) = \underbrace{\mathbb{P}(X = x \mid Y = y)}_{\text{second step, given the first step}} \underbrace{\mathbb{P}(Y = y)}_{\text{First step}}$$

All of the quantities of (X, Y) can be computed knowing these two probabilities:

$$\begin{aligned}
\mathbb{E}[h(X,Y)] &= \sum_{x,y} h(x,y)\mathbb{P}(X = x, Y = y) = \sum_{x,y} h(x,y)\mathbb{P}(X = x \mid Y = y)\mathbb{P}(Y = y) \\
\mathbb{E}[h(X,Y) \mid Y = y] &= \sum_{x} h(x,y)\mathbb{P}(X = x \mid Y = y) \\
\mathbb{E}[h(X,Y) \mid Y] &= \sum_{x,y} h(x,y)\mathbb{P}(X = x \mid Y = y)\mathbf{1}_{Y=y} \\
\mathbb{P}(X \in \cdot \mid Y) &= \sum_{y} \mathbb{P}(X \in \cdot \mid Y = y)\mathbf{1}_{Y=y}
\end{aligned}$$

If Y is not discrete, then we cannot compute the conditional probability $\mathbb{P}(X = x \mid Y = y)$, as we would divide by zero.
In order to avoid this we define

Definition 1.1: Markov Kernel

$\kappa(\cdot, \cdot)$ is a Markov kernel, if it is a measure in the second component and measurable in the first.

A Markov kernel satisfies

$$\mathbb{E}[h(X,Y) \mid Y = y] = \int h(x,y)\kappa(y, dx) \quad \text{and} \quad \mathbb{E}[h(X,Y) \mid Y] = \int h(x,y)\kappa(Y, dx)$$

We also call κ the regularized conditional expectation of X given Y.
In the discrete case we would write $\kappa(y, A) = \mathbb{P}(X \in A \mid Y = y)$.
The measure $\kappa(y, \cdot)$ has the density

$$\kappa(y, x) = \frac{f_{X,Y}(x, y)}{f_X(x)} f_Y(y)$$

We will avoid the notation of the markov kernels and write

$$\mathbb{P}(X \in A \mid Y = y) := \kappa(y, A)$$

and

$$\mathbb{P}(X \in A \mid Y) := \kappa(Y, A),$$

i.e. we would have

$$\mathbb{E}[h(X, Y) \mid Y] = \int h(x, Y)\mathbb{P}(X \in dx \mid Y)$$

Later, if X and Y are independent then we have that $\kappa(y, A) = \mathbb{P}(X \in A)$ such that

$$\mathbb{E}[h(X, Y) \mid Y] = \int h(x, Y)\mathbb{P}(X \in dx)$$

If X, Y are independent then

$$\kappa(Y, dx) = \mathbb{P}_Y(\Omega)\mathbb{P}_X(dx) = \mathbb{P}_X(dx)$$

2 Introduction of stochastic Bandits

The distribution of the outcome of K-arms is given by $P_{a_1}, ..., P_{a_K}$. The choice that the actor chooses an arm is defined by the random variable A on $\mathcal{A} := \{a_1, ..., a_K\}$. It is very important to note that we assume that the distribution of the arms do not change over time, i.e. they are stationary!

Definition 2.1: Policy

A policy is a distribution π on $\mathcal{A} := \{a_1, ..., a_K\}$. We call the family $\Pi_\Theta := \{\pi_\theta \mid \theta \in \Theta\}$ where $\Theta := \{p \in \mathbb{R}^K \mid \sum_{i=1}^K p_i = 1, p_i \geq 0\}$ the family of the policy on a $K-$armed bandit.

Definition 2.2: Bandit Model

For an index set $\mathcal{A}$ and a family of distributions $\nu := (P_a)_{a\in\mathcal{A}}$ with finite expectation, which we call the reward distribution, we call

- ν a stochastic bandit model with $K < \infty$
- The action value as the expectation

$$Q_a := \int_{\mathbb{R}} x dP_a(x)$$

and the best arm as

$$a^* \in argmax_{a\in\mathcal{A}} Q_a$$

We will write for Q_{a^*} in the future Q_*.

- ($n = \infty$ allowed) A learning strategy for n rounds consists of
  - An initial distribution π_1 on $\mathcal{A}$
  - A sequence $(\pi_t)_{t=2,...,n}$ of kernels on $(\Omega_{t-1} \times \mathcal{A})$, where

$$\Omega_t := \{(a_1, x_1, ..., a_t, x_t) \in (\mathcal{A} \times \mathbb{R})^t\}$$

the set of all trajectories. We will write for the probability of playing arm a at time t given $(a_1, x_1, ..., a_t, x_t)$ as

$$\pi_t(a; a_1, x_1, ..., a_t, x_t).$$

The policy can be viewed as our statistical model and the learning strategy are a sequence of elements from this statistical model.
Further, we assume that the Q_a are unknown, but we can sample from it (generative model).
We will later see that learning strategies depend on the maximal time horizon (quite obvious).

Remark 2.1

We have that $A_1^\pi \sim \pi_1$, for all $t \in \mathbb{N}$: $X_t^\pi \sim \kappa(A_t^\pi(\omega), \cdot) =: P_{A_t^\pi(\omega)}(\cdot) = \sum_{a \in \mathcal{A}} P_a(\cdot)\mathbf{1}_{A_t^\pi(\omega)=a}$ and for all $t > 1$:

$$A_t^\pi \sim \kappa((A_1^\pi, X_1^\pi, ..., A_{t-1}^\pi, X_{t-1}^\pi)(\omega), \cdot) =: \pi_t(\cdot; A_1^\pi, X_1^\pi, ..., A_{t-1}^\pi, X_{t-1}^\pi).$$

Then

$$\begin{aligned}
\mathbb{E}[X_t^\pi] &= \mathbb{E}[\mathbb{E}[X_t^\pi \mid A_1^\pi, X_1^\pi, ..., A_t^\pi]] \\
&= \mathbb{E}[\int_{\mathcal{R}} x \ \kappa((A_1^\pi, X_1^\pi, ..., A_t^\pi), X_t^\pi \in dx)] \\
&= \mathbb{E}[\int_{\mathcal{R}} x \ \sum_{a \in \mathcal{A}} \mathbf{1}_{A_t^\pi = a} P_a(dx)] \\
&= \mathbb{E}[\sum_{a \in \mathcal{A}} \mathbf{1}_{A_t^\pi = a} Q_a] \\
&= \sum_{a \in \mathcal{A}} \mathbb{P}(A_t^\pi = a) Q_a
\end{aligned}$$

Definition 2.3: Types of Learning strategies

A learning strategy is called an index strategy, if all measures are dirac measures, i.e. one arm is always played with probability one.
A learning strategy is called soft, if all measures have probability strictly greater than zero for all arms.

Next, we define a stochastic bandit process (analogously to Markov chains, where we first introduced transitions matrices and then then defined a Markov Chain and proved its existence.

Definition 2.4: Stochastic Bandit Process

Consider a stochastic Bandit model ν and a learning strategy $(\pi_t)_{t=1,...,n}$ for n rounds. Then a stochastic bandit process $(A_t^\pi, X_t^\pi)_{t=1,...,n}$ on some $(\Omega, \mathcal{F}, \mathbb{P})$ is called a stochastic bandit process with learning strategy π, if

- $A_1^\pi \sim \pi_1$
- $\mathbb{P}(A_t^\pi = a \mid A_1^\pi, X_1^\pi, ..., A_{t-1}^\pi, X_{t-1}^\pi) = \pi(a; A_1^\pi, X_1^\pi, ..., A_{t-1}^\pi, X_{t-1}^\pi),\ a \in \mathcal{A}$
- $\mathbb{P}(X_t^\pi \in B \mid A_1^\pi, X_1^\pi, ..., A_t^\pi) = \sum_{a\in\mathcal{A}} \underbrace{P_a(B)}_{=\mathbb{P}(X_t^\pi \in B \mid A_t^\pi = a)} \mathbf{1}_{A_t^\pi = a} =: P_{A_t^\pi}(B),\ B \in \mathcal{B}(\mathcal{R})$

The interpretation of the second and third property, is that the actions depend on the past, but the rewards do not depend on the past! Only on the last action taken.
In words: A stochastic bandit process is a stochastic process that, first, given a learning strategy π in every round this learning strategy suggests probabilities for actions that are based on the past. Then, the sampled arm A_t^π is used to play the corresponding arm and observe the outcome X_t^π.
It is not clear, that such a probability space exists, that satisfies these conditions for the sotchastic bandit Process:

Theorem 2.5: Existence of stochastic Bandit Process

For every bandit model and every learning strategy $(\pi_t)_{t=1,...,n}$ there exists a corresponding stochastic bandit process (A^π, X^π).

Proof. Existence:
Let $(X_t^{(a)})_{a\in\mathcal{A}, t\in\mathbb{N}}$ be a table of independent random variables such that

$$\forall t \in \mathbb{N}, a \in \mathcal{A} : X_t^{(a)} \sim P_a$$

and let $U_1, U_2, ... \overset{iid}{\sim} \mathcal{U}([0,1])$. Then with the Kolmogorov extension theorem, there exists some joint probability space $(\Omega, \mathcal{F}, \mathbb{P})$.
We will now use $(U_t)_{t\in\mathbb{N}}$ to construct the $(A_t^\pi)_{t\in\mathbb{N}}$. For this let $(\pi_t)_{t=1,...,n}$ be a learning strategy. Construction via induction:
Remember that we can use uniformly distributed random variables to sample from discrete random variables.
$t = 1$:

$$I_k := [\sum_{i=1}^{k-1} \pi_1(a_i), \sum_{i=1}^{k} \pi_1(a_i)]$$

an interval of length $\pi_1(a_k)$ and $\bigcup_{k=1}^K I_k$ is a partition of $[0,1]$. Then

$$\bar{U}(U_1) := \sum_{k=1}^{K} a_k \mathbf{1}_{U_1 \in I_k} \sim \pi_1.$$

Now define $A_1^\pi := \bar{U}(U_1)$ and clearly

$$X_1^{(A_1^\pi)} \sim \sum_{a\in\mathcal{A}} P_a \mathbf{1}_{A_1^\pi = a}$$

then set $X_1^{(A_1^\pi)} =: X_1^\pi$.
$t \to t+1$: Now we define for an entire sequence of $(A_1^\pi, X_1^\pi, ..., X_{t-1}^\pi)$ the distribution $\pi_t(\cdot \mid A_1^\pi, ..., X_{t-1}^\pi)$ by defining first

$$I_t(A_1^\pi, ..., X_{t-1}^\pi) := \bigcup_{k=1}^K I_t^{(a_k)}$$

where

$$\forall k = 1, ..., K : I_t^{(a_k)} := [\sum_{i=1}^{k-1} \pi_t(a_i; A_1^\pi, ..., X_{t-1}^\pi), \sum_{i=1}^{k} \pi_t(a_i; A_1^\pi, ..., X_{t-1}^\pi)],$$

i.e. $I_t^{(a_k)}$ is an interval of length $\pi_t(a_k; A_1^\pi, ..., X_{t-1}^\pi)$ and $I_t(A_1^\pi, ..., X_{t-1}^\pi)$ is a partition of $[0,1]$.

Finally, we have that

$$\bar{U}(U_t) := \sum_{k=1}^K a_k \mathbf{1}_{U_t \in I_t^{(a_k)}} \sim \pi_t(\cdot; A_1^\pi, ..., X_{t-1}^\pi), \quad t = 2, ..., n.$$

Now we can use $\bar{U}(U_{t+1})$ to sample from $\pi_{t+1}(\cdot; A_1^\pi, ..., X_t^\pi)$ and set $A_{t+1}^\pi := \bar{U}(U_{t+1})$. Then, again

$$X_{t+1}^{(A_{t+1}^\pi)} \sim \sum_{a \in \mathcal{A}} P_a \mathbf{1}_{A_{t+1}^\pi = a}.$$

and set $X_{t+1}^{(A_{t+1}^\pi)} =: X_{t+1}^\pi$.
Properties:
Wrapping up, we now need to show the two properties of the stochastic bandit model for all $t = 1, ..., n$. Before we do this, note that, because $(U_t)_{t \in \mathbb{N}}$ is an iid sequence and we used this for the construction of $(A_t^\pi)_{t \in \mathbb{N}}$ and because $(X_t^{(a)})_{a \in \mathbb{N}, a \in \mathbb{N}}$ are by definition independend in both components, it holds that for all $t = 1, ..., n$ and all $a \in \mathcal{A}$ that the conditional distribution

$$\mathbb{P}(U_t \in \cdot \mid A_1^\pi = a_1, ..., X_{t-1}^\pi = x_{t-1}) = \mathbb{P}(U_t \in \cdot) = \lambda(\cdot),$$

where λ is the Lebesgue measure. Further, note that because we defined the $\pi_t(\cdot \mid A_1^\pi, .., X_{t-1}^\pi)$ using absolute continuous random variables, it is a Markov kernel, i.e.

$$\kappa(a_1, ..., x_{t-1}, \cdot) = \mathbb{P}(U_t \in \cdot \mid A_1 = a_1, ..., X_{t-1} = x_{t-1}).$$

Now we can show the two properties for all $t = 1, ..., n$

1)

$$\begin{aligned}
\mathbb{P}(A_t = a \mid A_1^\pi, ..., X_{t-1}^\pi) &= \mathbb{P}(\bar{U}(U_t) = a \mid A_1^\pi, ..., X_{t-1}^\pi) \\
&= \mathbb{P}(U_t \in I_t^{(a)} \mid A_1^\pi, ..., X_{t-1}^\pi) \\
&= \mathbb{E}[\mathbf{1}_{U_t \in I_t^{(a)}} \mid A_1^\pi, ..., X_{t-1}^\pi] \\
&= \int \mathbf{1}_{u \in I_t^{(a)}} \mathbb{P}(U_t \in du \mid A_1^\pi, ..., X_{t-1}^\pi) \\
&= \int \mathbf{1}_{u \in I_t^{(a)}} \mathbb{P}(U_t \in du) \\
&= \int \mathbf{1}_{u \in I_t^{(a)}} du \\
&= \lambda(I_t^{(a)}) \\
&\overset{Def}{=} \pi_t(a; A_1^\pi, ..., X_{t-1}^\pi)
\end{aligned}$$

Note that we were allowed to use the indepence of the kernel due to the fact that du does not depened on $A_1^\pi, X_1^\pi, ...,$ but $I_t^{(a)}$ does.

2) For $B \in \mathcal{B}(\mathcal{R})$

$$\begin{aligned}
\mathbb{P}(X_t^\pi \in B \mid A_1^\pi, X_1^\pi, ..., A_t^\pi) &= \sum_{a\in\mathcal{A}} \mathbb{P}(X_t^{(A_t^\pi)} \in B, A_t^\pi = a \mid A_1^\pi, X_1^\pi, ..., A_t^\pi) \\
&= \sum_{a\in\mathcal{A}} \mathbb{P}(X_t^{(a)} \in B, A_t^\pi = a \mid A_1^\pi, X_1^\pi, ..., A_t^\pi) \\
&= \sum_{a\in\mathcal{A}} \mathbb{E}[\mathbf{1}_{X_t^{(a)}\in B}\mathbf{1}_{A_t^\pi=a} \mid A_1^\pi, X_1^\pi, ..., A_t^\pi] \\
&\overset{measb.}{=} \sum_{a\in\mathcal{A}} \mathbf{1}_{A_t^\pi=a}\mathbb{E}[\mathbf{1}_{X_t^{(a)}\in B} \mid A_1^\pi, X_1^\pi, ..., A_t^\pi] \\
&\overset{ind.}{=} \sum_{a\in\mathcal{A}} \mathbf{1}_{A_t^\pi=a}\mathbb{E}[\mathbf{1}_{X_t^{(a)}\in B}] \\
&\overset{ind.}{=} \sum_{a\in\mathcal{A}} \mathbf{1}_{A_t^\pi=a}\mathbb{P}(X_t^{(a)} \in B) \overset{Def}{=} P_{A_t^\pi}(B).
\end{aligned}$$

□

We can have two optimization goals

(A) For fixed $n \in \mathbb{N}$ find a learning strategy $(\pi_t)_{t=1,...,n}$ such that it maximizes

$$\mathbb{E}[\sum_{i=1}^n X_t^\pi]$$

(B) For fixed $n \in \mathbb{N}$ find a learning strategy, that minimizes the probability of choosing the wrong arm, i.e.

$$\pi^* \in argmin_\pi \mathbb{P}(A_t^\pi \neq a^*).$$

If this holds for all t or only the last or some, is not predefined.

We will first do (A). Note first that there might be several best arms. In that case, it does not matter which of the best arms we choose. Second,

$$\max_\pi \mathbb{E}[\sum_{t=1}^n X_t^\pi] = \min_\pi (nQ_* - \mathbb{E}[\sum_{t=1}^n X_t^\pi])$$

Definition 2.6: Regret

Suppose ν is a bandit model and $(\pi_t)_{t=1,...,n}$ is a learning strategy, then the culmmulateted regret is defined as

$$R_n(\pi) := nQ_* - \mathbb{E}[\sum_{t=1}^n X_t^\pi].$$

It describes how much is lost, when not playing optimally. Now a natural question is: What is a good and a bad learning strategy?

We get linear regret when our learning strategy is in all times steps the uniform distribution, i.e. $\pi_i \overset{(d)}{=} \pi_1 \overset{(d)}{=} \mathcal{U}([0,1])$. Then we have

$$R_n(\pi) = nQ_* - n\mathbb{E}[X_1^\pi] = n \underbrace{(Q_* - \frac{1}{K}\sum_{k=1}^{K} Q_{a_k})}_{=const \geq 0}$$

Therefore, we want the regret to grow slower than linear.

Definition 2.7: Reward Gab

We define the reward gab of arm a as the difference between the Q value of the optimal and arm a, i.e.

$$\Delta_a := Q_* - Q_a.$$

We do not want to have that bounds of the regret for algorithm depend on the Q vaules, as they are in theory unknown. It is a little more reasonable that the regret bounds depend on the reward gabs. Often the bounds depend on the time horizon n. There are model free bounds, which do not depend on the bandit model ν and there are model based bounds, that depend on the bandit model ν, i.e. the rewards gabs, Q-Values or variance.
We will see that finding alorithms, that have model based upper bounds that are logarithmic in n are not that hard to find. But they depend on Δ_a. If now the reward Gab of the best and second best arm is arbitrary small, then we will see that this upper bound explodes, which is not so ideal, especially if n is finite. For the UCB algorithm we will get two regret bounds that look something like

$$R_n(\pi) \leq const.\sqrt{n\log(n)} + const.\sum_{a\in\mathcal{A}} \Delta_a \quad \text{and} \quad R_n(\pi) \leq const.\log(n)\sum_{a\neq a^*}\frac{1}{\Delta_a} + const.\sum_{a\in\mathcal{A}}\Delta_a.$$

the second bound is better in $n \to \infty$, but for finite $n \in \mathbb{N}$ the constants can dominate large n, because $\frac{1}{\Delta_a}$ can be huge! Here the first bound comes into play, where the constants are smaller, but its performance for $n \to \infty$ is worse. As it might not be very realistic in practical situations to have infinite samples, for finite $n \in \mathbb{N}$ the first bound is sharper for small n and the second is sharper for large n.
It is also very important to note that, if we know that the arms are of a particular distribution then we can bound the reward Gab by above. For instance if we have Bernulli arms $P_a \overset{(d)}{=} Ber(p_a)$, then

$$\Delta_a = Q_* - Q_a \leq 1.$$

This can be very beneficial!

Remark 2.2

What is the practical benefit of calculating these upper regret bounds? Well first, they give us haranteed performance bounds and as we will later see (explore then commit algorithm) these bounds will depend on different parameters of the appearing algorithms and thus can be used, to tune parameters to improve performance!

A very important tool in calculating the upper regret bounds will be the following.

Lemma 2.8: Regret Decomposition Lemma

For the number of times arm a was played $T_a(n) := \sum_{t=1}^{n} \mathbf{1}_{A_t=a}$ we can write the regret of policy π after n rounds as

$$R_n(\pi) = \sum_{a \in \mathcal{A}} \Delta_a \mathbb{E}[T_a(n)].$$

Proof. We begin by rewriting

$$\begin{aligned} R_n(\pi) = nQ_* - \mathbb{E}[\sum_{t=1}^{n} X_t^\pi] &= \sum_{t=1}^{n} \mathbb{E}[Q_* - X_t^\pi] \\ &= \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \mathbb{E}[(Q_* - X_t^\pi)\mathbf{1}_{A_t^\pi=a}] \end{aligned}$$

Now using the tower property, i.e. for every two random variables X, Y we have that

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]],$$

we get that for every $t \leq n$ and $a \in \mathcal{A}$ we have that

$$\begin{aligned} \mathbb{E}[(Q_* - X_t^\pi)\mathbf{1}_{A_t^\pi=a}] &= \mathbb{E}_\pi[\mathbb{E}[(Q_* - X_t^\pi)\mathbf{1}_{A_t^\pi=a} \mid A_1^\pi, X_1^\pi, ..., A_t^\pi]] \\ &= \mathbb{E}[\mathbf{1}_{A_t^\pi=a}\mathbb{E}_\pi[(Q_* - X_t^\pi) \mid A_1^\pi, X_1^\pi, ..., A_t^\pi]] \\ &= \mathbb{E}[\mathbf{1}_{A_t^\pi=a}(Q_* - \mathbb{E}[X_t^\pi \mid A_1^\pi, X_1^\pi, ..., A_t^\pi])] \\ &= \mathbb{E}[\mathbf{1}_{A_t^\pi=a}(Q_* - \int x\mathbb{P}(X_t^\pi \in dx \mid A_1^\pi, X_1^\pi, ..., A_t^\pi))] \\ &\overset{(*)}{=} \mathbb{E}[\mathbf{1}_{A_t^\pi=a}(Q_* - \int x dP_{A_t^\pi}(x))] \\ &= \mathbb{E}[\mathbf{1}_{A_t^\pi=a}(Q_* - Q_{A_t^\pi})] \\ &= \mathbb{E}[\mathbf{1}_{A_t^\pi=a}(Q_* - Q_a)] \\ &= \Delta_a \mathbb{E}[\mathbf{1}_{A_t^\pi=a}] = \Delta_a \mathbb{E}_\pi[T_n(a)] \end{aligned}$$

Although, we used in (*) the third condition of the stochastic bandit process, i.e. $(X_t^\pi \mid A_1^\pi, X_1^\pi, ..., A_t^\pi) \sim P_{A_t^\pi}$. This finally leads to

$$R_n(\pi) = \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \Delta_a \mathbb{E}[\mathbf{1}_{A_t^\pi=a}] = \sum_{a \in \mathcal{A}} \Delta_a \mathbb{E}[\sum_{t=1}^{n} \mathbf{1}_{A_t^\pi=a}].$$

□

The regret is therefore the sum over all reward gabs and weighted by the expected number of times we play the respective arm.

Definition 2.9: Failure Probability

We define the probability that an suboptimal arm is chosen at time t as

$$\tau_t(\pi) := \mathbb{P}_\pi(Q_* \neq Q_{A_t})$$

which we call the failure probability.

It is clearly desirable that this probability converges to zero as fast as possible. Note that this is the optimization goal of (B). This is typically not done in stochastic bandits, but rather in reinforcement learning. But this also connects very well to the regret. Note that we can write

$$\mathbb{E}_\pi[T_n(a)] = \sum_{t=1}^n \mathbb{P}_\pi(A_t = a)$$

and then the regret decomposition lemma leads to the following result

Lemma 2.10

We can rewrite the regret as

$$R_n(\pi) = \sum_{t=1}^n \sum_{a \in \mathcal{A}} \Delta_a \mathbb{P}_\pi(A_t = a)$$

and get the bounds

$$R_n(\pi) \leq \max_{a \in \mathcal{A}} \Delta_a \sum_{t=1}^n \tau_t(\pi) \quad \text{and} \quad R_n(\pi) \geq \min_{a \neq a_*} \Delta_a \sum_{t=1}^n \tau_t(\pi)$$

This shows that studying the failure probability and the regret are ultimately connected.
Note that the upper bound is quite trivial:

$$R_n(\pi) = \sum_{t=1}^n \underbrace{\sum_{a \in \mathcal{A}} \Delta_a \mathbb{P}_\pi(A_t = a)}_{= \sum_{a \neq a^*} \Delta_a \mathbb{P}_\pi(A_t = a)} \leq \sum_{t=1}^n \underbrace{\mathbb{P}_\pi(A_t \neq a^*)}_{= \tau_t(\pi)} \max_{a \in \mathcal{A}} \Delta_a.$$

If we would have an integral instead of a sum, then with the fundamental theorem of integration theory, we have that

$$\int_0^n \tau_t(\pi) dt =: Tau_0(\pi) - Tau_n(\pi), \quad Tau_t'(\pi) = \tau_t(\pi).$$

I.e. the failure probability describes the rate at which the regret upper bound changes (in approximation).

Remark 2.3

The following two result will be important, for later

- If the failure probability does not decay to zero, then we have linear regret
- If the failure probability behaves like $\frac{1}{t}$ then we have that

$$\sum_{t=1}^n \tau_t(\pi) \leq \int_0^n \frac{1}{t} dt = \log(n).$$

3 Algorithms: the exploration-exploitation trade-off

3.1 Basic committ-then-exploit algorithm

We first recall the law of large number, where for iid random varibales

$$\frac{1}{n}\sum_{t=1}^{n} Y_t \to \mathbb{E}[Y_1], n \to \infty.$$

Definition 3.1: Estimated action Value

For a stochastic bandit process (A^π, X^π) and a learning strategy π we define

$$\hat{Q}_a(t) := \frac{1}{T_a(t)} \sum_{k=1}^{t} \mathbf{1}_{A_k=a} X_k^{(a)}$$

the estimated action value.

Using the LLN the basic explore then commit algorithm is that we play m times every arm and the remaining $n - mK$ rounds play the best arm.
If we play an arm a infinitely often, then

$$\lim_{t\to\infty} \hat{Q}_a(t) = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \mathbf{1}_{A_k=a} X_k^{(a)},$$

But in general it does not hold for a learning strategy π, that $T_a(t)$ goes to infinity for every action.
For the explore then commit algorithm, we have clearly the following index strategy

$$\pi_t(a; a_1, x_1, ..., x_{t-1}) = \begin{cases} 1 & t \le Km, a = a_{t mod K+1} \\ 1 & t > Km, a = argmax_a \hat{Q}(Km) \\ 0 & else \end{cases}$$

where $\hat{Q}(t) := (\hat{Q}_1(t), ..., \hat{Q}_K(t))$.

Algorithm 1: Explore then Commit

Data: m, n, bandit model ν ;
Set $\hat{Q}(0) \equiv 0$;
while $t \le n$ **do**
  Set $A_t = \begin{cases} a_{t mod K+1} & t \le mK \\ argmax_a \hat{Q}_a(mK) & t > mK \end{cases}$;
  Set Obtain reward X_t by playing arm A_t ;
Output: Actions $A_1, ..., A_n$ and rewards $X_1, ..., X_n$

Remark 3.1

In the first mK rounds we cycle through the K arms. After that we take the best arm, at time mK for all the other rounds.
Very important: We do not update the Q-Value in exploitation phase, i.e. we do not have a greedy algorithm. If we would update the Q-value in the exploitation phase it would be called, explore then greedy. But we only consider explore then commit!

This algorithm produces the following regret bound

Definition 3.2: σ-Subgaussian

A random variable X on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is called $\sigma-$Subgaussian for $\sigma > 0$, if for all $\lambda \in \mathbb{R}$ it holds that

$$\mathcal{M}_{X-\mathbb{E}[X]}(\lambda) := \mathbb{E}[e^{\lambda(X-\mathbb{E}[X])}] \leq e^{\lambda^2\sigma^2/2}.$$

Lemma 3.3: Regret Bound for explore then commit

For a σ-Subgaussian bandit model ν and the learning strategy π that follows the explore then commit algorithm, i.e. we play m times every arm and then $n - mK$ times the best arm, it holds

$$R_n(\pi) \leq m \sum_{a \in \mathcal{A}} \Delta_a + (n - mK) \sum_{a \in \mathcal{A}} \Delta_a e^{\frac{\Delta_a^2 m}{4\sigma^2}}$$

The summand of the form $\sum_{a\in\mathcal{A}} \Delta_a$ must appear in every reasonable regret bound for learning strategies, that explore the presented arms.
In the explore then commit algorithm it is clear that for m large enough we should have $\hat{Q}_a(m) \approx Q_a(n)$ and thus the we should choose the best arm, after the exploration. But some arms could be overestimated in the sense that their rewards after m rounds are above average. In that case the algorithm could choose an arm that is suboptimal. This overestimation can be explained using concentration inequalities:
A concentration inequality is a bound of the probability, that the random variable deviates from its mean

$$c(a) \leq \mathbb{P}(|X - \mathbb{E}[X]| > a) \leq C(a) \quad \text{or} \quad c(a) \leq \mathbb{P}(X - \mathbb{E}[X] > a) \leq C(a)$$

The faster the functions $C(a), c(a)$ decrease, the more the random variable X is concentrated. A similiar way of viewing this is saying that for a strongly increasing function g the expectation

$$\mathbb{E}[g(X)] = \begin{cases} \int_{\mathbb{R}} g(y) f_X(y) dy & , X \text{ is continuous} \\ \sum_{k=1}^{N} g(a_k) P_k & , X \text{ is discrete} \end{cases}$$

exists if and only if, X is concentrated, i.e. the density f_X (or probability weights p_k) force the product to decrease very strongly!
A concentration inequality we already know is the Markov inequality

$$\mathbb{P}(|X - \mathbb{E}[X]| > a) \leq \frac{\mathbb{V}[X]}{a^2}, \quad \mathbb{E}[X^2] < \infty.$$

But this inequality is not very sharp, i.e. the upper bound is useless. For σ-Subgaussian random variables we can find a lot more sharper bounds. This will be done now and is a necessary result in order to derive the regret bound for the explore then commit algorithm.

Proposition 3.4

Let X be a σ-Subgaussian, then it holds for all $a > 0$:

$$\mathbb{P}(X \geq a) \leq e^{-\frac{a^2}{2\sigma^2}} \quad \text{and} \quad \mathbb{P}(|X| \geq a) \leq 2e^{-\frac{a^2}{2\sigma^2}}$$

Proof. The idea is to use exponential Markov inequality. For $\lambda \in \mathbb{R}$

$$\begin{aligned}\mathbb{P}(X \geq a) &= \mathbb{P}(\lambda X \geq \lambda a) = \mathbb{P}(e^{\lambda X} \geq e^{\lambda a}) \\ &\leq \frac{\mathbb{E}[e^{\lambda X}]}{e^{\lambda a}} \\ &= \frac{e^{\frac{\lambda^2\sigma^2}{2}}}{e^{\lambda a}} = e^{\frac{\lambda^2\sigma^2}{2} - \lambda a} =: h(\lambda)\end{aligned}$$

One can show that h is minimized by $\lambda^* = \frac{a}{\sigma^2}$ using differentiation. This leads to our bound.
The second case is analogously, although we first write

$$\mathbb{P}(|X| \geq a) = \mathbb{P}(X \geq a \text{ or } X \leq -a) = \mathbb{P}(X \geq a) + \mathbb{P}(-X \geq a).$$

This yields the assertion, because we showed that for any constant c then cX is $|c|\sigma$-Subgaussian.

□

Lemma 3.5

For a σ-Subgaussian random variable we have

- $\mathbb{V}[X] \leq \sigma^2$
- For a constant c it holds cX is $|c|\sigma$-Subgaussian
- If X_1, X_2 are independent σ_1, σ_2-Subgaussian (respectively) random variables, then $X_1 + X_2$ are $\sqrt{\sigma_1^2 + \sigma_2^2}$-Subgaussians.
- ...

Now what follows is called concentration inequalities.

Theorem 3.6: Höffdings Inequality

Suppose we have $X_1, X_2, ...$ iid random variables on $(\Omega, \mathcal{F}, \mathbb{P})$ with $\mu := \mathbb{E}[X_1]$ and X_1 is σ-Subgaussian, then for all $n \in \mathbb{N}$ and all $a > 0$,

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu \geq a\right) \leq e^{-\frac{na^2}{2\sigma^2}} \quad \text{and} \quad \mathbb{P}\left(|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu| \geq a\right) \leq 2e^{-\frac{na^2}{2\sigma^2}}$$

Proof. We only need to look at how the σ looks like for the Subgaussian. We have shown, that $\sum_{i=1}^n X_i$ is $\sqrt{\sum_{i=1}^n \sigma^2} = \sqrt{n}\sigma$-Subgaussian. Then $\frac{1}{n}\sum_{i=1}^n X_i$ is $\frac{\sigma}{\sqrt{n}}$-Subgaussian and the assertion immediately follows from the previous Proposition. □

Now we have derived all tools we need in order to prove the Regret bound for ETC:

Proof. Without loss of generality we can assume that the first arm is the best arm, i.e. $Q_{a_1} = Q_*$. Next, we can rewrite for $c \in \mathbb{N}$ with $c < n - mK$

$$\begin{aligned} T_a(n) &= m + (n - mK)\mathbf{1}_{\text{arm a is the best arm at mK}} \\ &= m + (n - mK)\mathbf{1}_{\hat{Q}_a(mK) \geq \max_b \hat{Q}_b(mK)} \end{aligned}$$

Thus we can write the regret as

$$\begin{aligned} R_n(\pi) &= \sum_{a \in \mathcal{A}} \Delta_a \mathbb{E}[T_a(n)] = \sum_{a \in \mathcal{A}} \Delta_a \left(m + (n - mK)\mathbb{P}(\hat{Q}_a(mK) \geq \max_b \hat{Q}_b(mK)) \right) \\ &= \sum_{a \in \mathcal{A}} \Delta_a m + \sum_{a \in \mathcal{A}} \Delta_a (n - mK)\mathbb{P}(\hat{Q}_a(mK) \geq \max_b \hat{Q}_b(mK)) \end{aligned}$$

We can estimate the probability by above if we replace the maximum by an arbitrary arm. We choose it to be the best arm a_1. Note: $Q_{a_1} = Q_*$ but this does not mean that $a_1 \in argmax_b \hat{Q}_b$. This leads to

$$\begin{aligned} R_n(\pi) &= \sum_{a \in \mathcal{A}} \Delta_a m + \sum_{a \in \mathcal{A}} \Delta_a (n - mK)\mathbb{P}(\hat{Q}_a(mK) \geq \max_b \hat{Q}_b(mK)) \\ &\leq \sum_{a \in \mathcal{A}} \Delta_a m + \sum_{a \in \mathcal{A}} \Delta_a (n - mK)\mathbb{P}(\hat{Q}_a(mK) \geq \hat{Q}_{a_1}(mK)) \\ &\leq \sum_{a \in \mathcal{A}} \Delta_a m + \sum_{a \in \mathcal{A}} \Delta_a (n - mK)\mathbb{P}(\hat{Q}_a(mK) - \hat{Q}_{a_1}(mK)) - (Q_a - Q_{a_1}) \geq \underbrace{(Q_a - Q_{a_1})}_{=\Delta_a}) \end{aligned}$$

I.e. inside the probability we have written something in the form of

$$Z_a := \frac{1}{m} \sum_{i=1}^{m} (X_j^{(a)} - X_j^{(1)}) \quad \text{and} \quad \mathbb{P}(Z_a - \mathbb{E}[Z_a] \geq \underbrace{\mathbb{E}[Z_a]}_{=\Delta_a})$$

We now want to use Höffdings Inequality. Because $X_j^{(a)} - X_j^{(1)}$ is a sequence of iid random variables and their sum is a $\sqrt{2}\sigma$ Subgaussian r.v., we get

$$\mathbb{P}(Z_a - \mathbb{E}[Z_a] \geq \Delta_a) \leq e^{-\frac{m\Delta_a^2}{2(\sqrt{2}\sigma)^2}} = e^{-\frac{m\Delta_a^2}{4\sigma^2}}$$

□

Corollary 3.7

For a two armed bandit the upper bound for the regret of the explore then commit algorithm is minimized by

$$m = \max\{1, \lceil \frac{4\sigma^2}{\Delta} \log\left(\frac{n\Delta^2}{4\sigma^2} \right) \rceil\}.$$

Proof. When we have only two arms then $\Delta := \Delta_2$ and $\Delta_1 = 0$ (in the case that the first arm is optimal). It follows

$$R_n(\pi) \leq m\Delta + (n - 2m)\Delta e^{-\frac{m\Delta^2}{4\sigma^2}}.$$

Differentiation gives the assertion. □

Thus the best algorithm in the class of explore then commit algorithms with two arms

Corollary 3.8

For a two armed bandit with optimized exploration the upper bound for the regret of the explore then commit algorithm is given by

$$R_n(\pi) = \min\left\{n\Delta, \Delta + \frac{4\sigma^2}{\Delta}(1 + \max\{0, \log(\frac{n\Delta^2}{4\sigma^2})\})\right\}$$

and for $n \geq \frac{4}{\Delta^2}$ we have the model dependend regret bound

$$R_n(\pi) \leq C_\Delta + \frac{\log(n)}{\Delta}$$

Proof. — □

Remark 3.2

The big question is wether the optimal chosen m is reasonable? The answer is no, because it is cheating! We use information we do not have, i.e. the Reward Gabs and the σ might also be unknown. One could derive algorithms, that sample from these two parameters, but this is questions, that will not be answered here.

3.2 Purely Greedy

Before we begin, we can show that we can write the Q value in a specific way, such that we get a iterative way of writing it. For every arm $a \in \mathcal{A}$ and $n \in \mathbb{N}$ we assume that $T_a(n) = n$

$$\begin{aligned}
\hat{Q}_a(n) &= \frac{1}{n}\sum_{t=1}^{n} X_t^{(a)} \\
&= \frac{1}{n}(X_n^{(a)} + \sum_{t=1}^{n-1} X_t^{(a)}) \\
&= \frac{1}{n}(X_n^{(a)} + \frac{n-1}{n-1}\sum_{t=1}^{n-1} X_t^{(a)}) \\
&= \frac{1}{n}(X_n^{(a)} + (n-1)\hat{Q}_a(n-1)) \\
&= \hat{Q}_a(n-1) + \frac{1}{n}(X_n^{(a)} - \hat{Q}_a(n-1))
\end{aligned}$$

This is similar to stochastic gradient descent, because we can view this as "old Q value + learning rate $\frac{1}{n}$ times the gradient of a loss function (distance of current rewards vs. estimated old reward)".

Remark 3.3

We call an algorithm that uses an vector that is updated in each step in order to make decisions tabular. Later a similiar approach as the above will be Q-learning.

The purely greedy algorithm always plays the action at time t with the highest estimated Q-Value, i.e. $A_t := argmax_a \hat{Q}_a(t-1)$. The trick of this algorithm is to choose the ininilized Q-value as very large, because then we force exploration. But the question is, what is large?

Algorithm 2: Purely greedy bandit algorithm

Data : bandit model ν, vector $\hat{Q}$, n
Initialise $T_a = 0$ for all a;
while $t \leq n$ **do**
  Set $A_t = \arg\max_a \hat{Q}_a$;
  Obtain reward X_t by playing arm A_t;
  Set $T_{A_t} = T_{A_t} + 1$;
  Set $\hat{Q}_{A_t} = \hat{Q}_{A_t} + \frac{1}{T_{A_t}}(X_t - \hat{Q}_{A_t})$;
Result: actions $A_1, \ldots, A_n$ and rewards $X_1, \ldots, X_n$

Problems with this algorithm is the committal behavior

Definition 3.9: Committal Behavior

When a Bandit focuses to early on a suboptimal arm, is called comittal behavior.

A second way of forcing exploration is by centering all the arms (empirically). Bad arms will then get negative $\hat{Q}$ values and good arms will get positive ones. Again, here is the problem that we need to sample from the arms a few times, in order to calculate the empirical mean and variance. Finally, we still have positive probability of commital behavior.

3.3 ϵ-greedy bandit algorithms

Algorithm 3: ε-greedy bandit algorithm

Data : bandit model ν, exploration rate $\varepsilon \in (0,1)$, vector $\hat{Q}$, n
Initialise $T_a = 0$ for all a;
while $t \leq n$ **do**
  Sample $U \sim \mathcal{U}([0,1])$;
  if $U < \varepsilon$ **then**
    [exploration part]
    Uniformly choose an arm A_t;
  else
    [greedy part]
    Set $A_t = \arg\max_a \hat{Q}_a$;
  Obtain reward X_t by playing arm A_t;
  Set $T_{A_t} = T_{A_t} + 1$;
  Set $\hat{Q}_{A_t} = \hat{Q}_{A_t} + \frac{1}{T_{A_t}}(X_t - \hat{Q}_{A_t})$;
Result: actions $A_1, \ldots, A_n$ and rewards $X_1, \ldots, X_n$

This algorithm always explores with probability ϵ and with probability $1-\epsilon$ plays greedy. The following lemma shows that if we first play every arm once and then play according to ϵ-greedy, then we get linear regret, i.e. we learn effectively nothing.

Definition 3.10: Landau Symbols

For two function f, g we write

- $f \in \mathcal{O}(g) : \iff$
$$\limsup_{x \to x_0} \frac{|f(x)|}{|g(x)|} \to const. \in [0, \infty)$$
- $f \in o(g) : \iff$
$$\limsup_{x \to x_0} \frac{|f(x)|}{|g(x)|} \to 0$$

$f \in \mathcal{O}(g)$ is therefore, that f increases/decreases asymptotically at most slower/faster than g.
$f \in o(g)$ is therefore, that f increases/decreases asymptotically strictly slower/faster than g.

Lemma 3.11

Let π be a learning strategy, that first, explores every arm once and then plays ϵ-greedy, for $\epsilon \in (0, 1)$, then
$$\lim_{n \to \infty} \frac{R_n(\pi)}{n} = \frac{\epsilon}{K} \sum_{a \in \mathcal{A}} \Delta_a$$

Proof. — □

This means that the regret of ETC is at most linear, which is effectively: nothing is learned. Therefore, ETC with constant ϵ is bad.
In the following theorem we will show that, if we use ETC with decreasing ϵ_t over time with a specific speed, then we get
$$\limsup_{n \to \infty} \frac{R_n(\pi)}{\log(n)} = const. < \infty.$$

In order to show this we will reformulate it, so that it is easier to prove. First note

Lemma 3.12

If the failure probability $\tau_n(\pi)$ behaves asymptotically like $\frac{1}{n}$, i.e. $\tau_n(\pi) \in \mathcal{O}(\frac{1}{n})$, then
$$R_n(\pi) \in \mathcal{O}(\log(n) \sum_{a \in \mathcal{A}} \Delta_a) = \mathcal{O}(\log(n)).$$

Now we only have to look at the failure probability and can use the Lemma above.

Theorem 3.13: Explore then ϵ-greedy with decreasing ϵ

Suppose that all arms take values in $[0, 1]$ and define $d < \min_{a \neq a^*} \Delta_a$ and $C > \max\{5d^2, 2\}$. Then the

ϵ-greedy with decreasing epsilon defined as $\epsilon_t := \max\{1, \frac{CK}{d^2 t}\}$ satisfies

$$\limsup_{n\to\infty} \tau_n(\pi) n \leq \frac{(K-1)C}{d^2},$$

i.e. $\tau_n(\pi) \in \mathcal{O}(\frac{1}{n})$.

Remark 3.4

We will not prove this statement.
Note that we might have now a convergence of the regret of order $\log(n)$, which is in fact better than linear, but in application the size of constants is of importance and they can be very large, because if we have very small reward gabs (i.e. very similar arms), then $\frac{1}{d^2}$ is really large and will dominate the logarithm for a long time. For example $\log(100000) \approx 11.5$, which is still very small. Further, this approximation is again cheating, as the constant d assume prior knowledge of the rewards gaps

3.4 UCB algorithm

The idea of UCB is optimism in face of uncertainty. It goes as follows. First, explore every arm and then play greedy, although we add a exploration bonus to every Q-value in respect of how often each arm is explored. I.e. as we increase the trust in the estimation, we decrease the bonus. The variance of the estimation could play a role in the bonus, but this is not considered. We define the Q-Values plus the bonus of the UCB algorithm as

$$UCB_a(t,\delta) := \begin{cases} \infty & , T_a(t) = 0 \\ \hat{Q}_a(t) + \underbrace{\sqrt{\frac{2\log(1/\delta)}{T_a(t)}}}_{\text{exploration bonus}} & , T_a(t) > 0 \end{cases}$$

This exploration bonus is motivated by concentration inequalities. Consider that we have a Subgaussian bandit model ν and play an arm n times and receive rewards $X_1, ..., X_n$, then using Höffdings Inequality

$$\begin{aligned} \mathbb{P}(\frac{1}{n}\sum_{k=1}^{n} X_k \geq Q + \sqrt{\frac{2\log(1/\delta)}{n}}) &= \mathbb{P}(\frac{1}{n}\sum_{k=1}^{n} X_k - Q \geq \sqrt{\frac{2\log(1/\delta)}{n}}) \\ &\leq \exp\left(-\frac{n\sqrt{\frac{2\log(1/\delta)}{n}}^2}{2}\right) \\ &= \delta \end{aligned}$$

Thus, the probability that we overestimate using the mean with the expectation by the exploration bonus, is smaller than $\delta \in (0,1)$. We will later see that $\delta = \frac{1}{n^2}$ is a good choice.
This learning strategy looks as follows

$$\pi_t(a; a_1, x_1, ..., x_t) = argmax_a UCB_a(t-1, \delta).$$

The choice of $UCB(0,\delta) \equiv \infty$ forces the algorithm to explore each arm at least once.

Algorithm 4: UCB Algorithm

Data : bandit model ν, $\delta \in (0,1)$, vector $\hat{Q}$, n,vector $\hat{Q}$
Initialise $T_a = 0$ for all a;
while $t \leq n$ **do**
  $A_t = argmax_a UCB_a(t-1, \delta)$;
  Obtain reward X_t by playing A_t;
  $T_{A_t} += 1$;
  $\hat{Q}_{A_t}(t) = \hat{Q}_{A_t}(t) + \frac{1}{T_{A_t}}(X_t - \hat{Q}_{A_t}(t))$
Result: actions $A_1, \ldots, A_n$ and rewards $X_1, \ldots, X_n$

We can now calculate a regret bound for this algoithm. Note that in the following we do not initialize with $\hat{Q}(0) = \infty$, but rather with $\hat{Q} \equiv 0$.

Theorem 3.14: UCB first Regret Bound

Suppose we have a 1-Subgaussian bandit model ν with $\delta = \frac{1}{n^2}$ for the parameter of the UCB algorithm. When we initialize with $\hat{Q}(0) \equiv 0$, then we get the following bound

$$R_n(\pi) \leq 3 \sum_{a \in \mathcal{A}} \Delta_a + 16 \log(n) \sum_{a \neq a^*} \frac{1}{\Delta_a}$$

Proof. W.l.o.g. let $a_* = a_1$. We will now estimate the expected number of times a suboptimal arm will be played.
The Idea of the proof is as follows

1) We know that we can write with the regret decomposition Lemma

$$R_n(\pi) = \sum_{a \in \mathcal{A}} \Delta_a \mathbb{E}[T_a(n)]$$

thus we can focus on only the expectation first.

2. We can write this expectation as

$$\mathbb{E}[T_a(n)] = \mathbb{E}[T_a(n) \mathbf{1}_{H_m}] + \mathbb{E}[T_a(n) \mathbf{1}_{H_m^C}]$$

where $H_m := \{\text{arm } a \text{ is played at most } m \text{ times}\}$

3) We can estimate this expectation by

$$\mathbb{E}[T_a(n)] =\leq m\mathbb{P}(H_m) + n\mathbb{P}(H_m^C)$$

because in case of H_m, the $T_a(n)$ could be at most equal to m.

4) We now want to estimate this probability and then optimize over m. This is a little tricky, which is why we will introduce smaller events $G_m \subseteq H_m$ for which we can estimate the probabilites.

Recall that we need for Hoeffdings inequality an iid sum of random variables. This is exactly why we did the random stack construction of the bandit process (A^π, X^π), because for every $t \leq n$ we have that $X_t^{(a)} \sim P_a$ and thus the estimations of the Q-Values is a sum of iid random variables:

$$\bar{Q}_s^{(a)} := \frac{1}{s} \sum_{k=0}^{s} X_k^{(a)}$$

and we have the relation that

$$\hat{Q}_a(t) = \bar{Q}_s^{(a)} \quad \iff \quad T_a(t) = s.$$

We now fix an arm a and want to estimate the expectation $\mathbb{E}[T_a(n)]$. For this we define the following sets

$$G_1 := \{\omega \in \Omega \mid Q_{a_1} < \min_{t \leq n} UCB_{a_1}(t)(\omega)\}$$
$$G_{2,m} := \{\omega \in \Omega \mid \bar{Q}_m^{(a)}(\omega) + \sqrt{\frac{2\log(1/\delta)}{m}} < Q_{a_1}, a \in \mathcal{A} \setminus \{a_1\}\}$$

The first set describes that the UCB algorithm never underestimates the Q-Values of the best arm a_1. The second set is the event that the estimated Q-Value and their bonus for the UCB algorithm is after m observations of one arm strictly smaller than the true Q-Value of the best arm a_1.
We define $G_m := G_1 \cap G_{2,m}$ as the event that both of the other events hold.
Step 1: To show that $G_m \subseteq H_m$ for all $m \leq n$, i.e. $\forall \omega \in G_m$ it follows that $T_a(n)(\omega) \leq m$.
We do a contradiction. Suppose $\omega \in G_m$ but $T_a(n)(\omega) > m$. Then it follows that there exists a time $t \leq n$ such that

$$A_t(\omega) = a \text{ and there is an } k = 1, ..., t-m \text{ such that } T_a(t-k)(\omega) = m,$$

i.e. we played between time $t-k$ to time $t-1$ different arms and then at time t the arm for the $m+1$ time. Now using the definitions of the two sets G_1 and $G_{a,2}$ yields

$$\begin{aligned} UCB_a(t-1)(\omega) &\overset{Def.}{=} \hat{Q}_a(t-1)(\omega) + \sqrt{\frac{2\log(1/\delta)}{T_a(t-1)(\omega)}} \\ &= \bar{Q}_m^{(a)}(\omega) + \sqrt{\frac{2\log(1/\delta)}{m}} \\ &\overset{\text{Def. } G_{a,s}}{<} Q_{a_1} \\ &\overset{\text{Def. } G_1}{<} \min_{t \leq n} UCB_{a_1}(t)(\omega) \leq UCB_{a_1}(t-1)(\omega) \end{aligned}$$

Thus the UCB algorithm would choose at time t arm a_1 and not arm a, which is a contradiction to $A_t(\omega) = a$!
Step 2: Estimation of the Probability of G_m. We begin by the following

$$\begin{aligned} \mathbb{E}[T_a(n)] &= \mathbb{E}[T_a(n)\mathbf{1}_{G_m}] + [T_a(n)\mathbf{1}_{G_m^C}] \\ &\leq \mathbb{E}[T_a(n)\mathbf{1}_{H_m}] + [T_a(n)\mathbf{1}_{G_m^C}] \\ &\leq \mathbb{E}[m\mathbf{1}_{H_m}] + [T_a(n)\mathbf{1}_{G_m^C}] \\ &\leq m + n\mathbb{E}[\mathbf{1}_{(G_1 \cap G_{2,m})^C}] \\ &= m + n\mathbb{P}(G_1^C \cup G_{2,m}^C) = m + n(\mathbb{P}(G_1^C) + \mathbb{P}(G_{2,m}^C)) \end{aligned}$$

We will now look at the two probabilities separately. First, notice that

$$\begin{aligned}
\mathbb{P}(G_1^C) &= \mathbb{P}(\exists t \leq n : Q_{a_1} \geq UCB_{a_1}(t)) \\
&= \sum_{t \leq n} \mathbb{P}(Q_{a_1} \geq UCB_{a_1}(t)) \\
&= \sum_{t \leq n} \mathbb{P}(Q_{a_1} \geq \hat{Q}_a(t) + \sqrt{\frac{2\log(1/\delta)}{T_a(t)}}) \\
&= \sum_{t \leq n} \mathbb{P}(Q_{a_1} - \frac{1}{T_a(t)} \sum_{k=1}^{T_a(t)} X_k \geq \sqrt{\frac{2\log(1/\delta)}{T_a(t)}}) \\
&\overset{\text{Hoeffding}}{\leq} \sum_{t \leq n} \delta = n\delta
\end{aligned}$$

The second probability can be estimated as follows

$$\begin{aligned}
\mathbb{P}(G_{2,m}^C) &= \mathbb{P}(\bar{Q}_m^{(a)}(\omega) + \sqrt{\frac{2\log(1/\delta)}{m}} \geq Q_{a_1}) \\
&= \mathbb{P}(\bar{Q}_m^{(a)}(\omega) + \underbrace{Q_a - Q_a}_{=0} + \sqrt{\frac{2\log(1/\delta)}{m}} \geq Q_{a_1}) \\
&= \mathbb{P}(\bar{Q}_m^{(a)}(\omega) - Q_a \geq \underbrace{Q_{a_1} - Q_a}_{=\Delta_a} - \sqrt{\frac{2\log(1/\delta)}{m}})
\end{aligned}$$

Now assume that we choose m large enough such that

$$\Delta_a - \sqrt{\frac{2\log(1/\delta)}{m}} \geq \frac{1}{2}\Delta_a$$

(for instance $m = \lceil \frac{2\log(1/\delta)}{1/4\Delta_a^2} \rceil$). Using this we get that

$$\begin{aligned}
\mathbb{P}(\bar{Q}_m^{(a)}(\omega) - Q_a \geq \Delta_a - \sqrt{\frac{2\log(1/\delta)}{m}}) &\leq \mathbb{P}(\bar{Q}_m^{(a)}(\omega) - Q_a \geq \frac{1}{2}\Delta_a) \\
&\overset{\text{Hoeffding}}{\leq} e^{-\frac{m\Delta_a^2}{8}}
\end{aligned}$$

Now plugging it all in

$$\mathbb{E}[T_a(n)] \leq m + n(n\delta + e^{-\frac{m\Delta_a^2}{8}})$$

If we plug in $\delta = \frac{1}{n^2}$ and $m = \lceil \frac{2\log(n^2)}{1/4\Delta_a^2} \rceil$ yields

$$\mathbb{E}[T_a(n)] \leq \underbrace{\lceil \frac{2\log(n^2)}{1/4\Delta_a^2} \rceil}_{\leq 1 + \frac{16\log(n)}{\Delta_a^2}} + 1 + ne^{-\log(n^2)} \leq 3 + \frac{16\log(n)}{\Delta_a^2}.$$

□

In contrast to the ETC algorithm, where we had to know the reward gabs, to calculate the optimal exploration length, in UCB we get $\mathcal{O}(\log(n))$ performance without knowing the reward gabs. They are here only a theoretical result.
As the bound above is very good in theory, for small n we have the problem that we have to divide by the reward Gabs, which could lead to very large values. If we change the final step in the proof above, we get a bound that is a little faster, but the constants appearing are smaller.

Theorem 3.15: UCB second Regret Bound

Suppose we have a $1-$Subgaussian bandit model ν, with $\delta = \frac{1}{n^2}$ for the UCB algorithm parameter. If we again initialize with $\hat{Q} \equiv 0$, then

$$R_n(\pi) \leq 8\sqrt{Kn\log(n)} + 3\sum_{a\in\mathcal{A}} \Delta_a.$$

Proof. Here we the results from the last proof, although we approach the problem a little smarter. Small reward gabs can be bounded by a threshold and thus are linear in n. The reward gabs above the threshold are the ones that make the problems and where we will apply the results from the last proof. We will then optimize this threshold. Chose $\Delta > 0$ such that

$$\sum_{a\in\mathcal{A}:\Delta_a<\Delta} \Delta_a \leq \Delta.$$

then

$$\begin{aligned}
R_n(\pi) &= \sum_{a\in\mathcal{A}} \Delta_a \mathbb{E}[T_a(n)] \\
&= \sum_{a\in\mathcal{A}:\Delta_a<\Delta} \Delta_a \mathbb{E}[T_a(n)] + \sum_{a\in\mathcal{A}:\Delta_a\geq\Delta} \Delta_a \mathbb{E}[T_a(n)] \\
&\leq \sum_{a\in\mathcal{A}:\Delta_a<\Delta} \Delta_a n + \sum_{a\in\mathcal{A}:\Delta_a\geq\Delta} \Delta_a(3 + \frac{16\log(n)}{\Delta_a^2}) \\
&\leq \Delta n + \sum_{a\in\mathcal{A}:\Delta_a\geq\Delta} \frac{16\log(n)}{\Delta_a} + \sum_{a\in\mathcal{A}} \Delta_a 3 \\
&\leq \Delta n + \frac{16K\log(n)}{\Delta} + 3\sum_{a\in\mathcal{A}} \Delta_a
\end{aligned}$$

Because $\Delta > 0$ was arbitrary, we can optimize the bound such that is as sharp as possible. The minimum is $\Delta^* := \sqrt{16K\log(n)/n}$ and plugging this in yields

$$R_n(\pi) \leq \sqrt{16Kn\log(n)} + \sqrt{16Kn\log(n)} + 3\sum_{a\in\mathcal{A}} \Delta_a = 2\cdot 4\cdot\sqrt{Kn\log(n)} + 3\sum_{a\in\mathcal{A}} \Delta_a.$$

□

The term $\sum_{a\in\mathcal{A}} \Delta_a$ appears in both bounds and is very natural, as every reasonable algorithm trys every arm at least once. Depending on the bandit model, this sum can also be bounded by above. For example for a bernulli bandit model, the sum can be approximated by K, i.e. the number of arms.

Remark 3.5

One can also show that if have $\sigma-$Subgaussian bandit model and choose $\delta = \frac{1}{n}^{2\sigma^2}$, then

$$UCB_a(t) := \begin{cases} \infty & , T_a(t) = 0 \\ \hat{Q}_a(t) + \underbrace{\sqrt{\frac{4\sigma^2 \log(n)}{T_a(t)}}}_{\text{exploration bonus}} & , T_a(t) > 0 \end{cases}$$

and we get for the two regret bounds

$$R_n(\pi) \leq 3 \sum_{a \in \mathcal{A}} \Delta_a + 16\sigma^2 \log(n) \sum_{a \neq a^*} \frac{1}{\Delta_a}$$

and

$$R_n(\pi) \leq 8\sigma \sqrt{KN \log(n)} + 3 \sum_{a \in \mathcal{A}} \Delta_a.$$

Potential: Exam Question!
The fact that σ appears, means that this is again is cheating. But one could try to obtain σ via samples.

Note that the model based regret bounds are not very sharp, because they focus to strongly on the reward Gabs compared to the regret decomposition lemma, i.e.

$$R_n(\pi) \overset{\text{regret decomposition Lemma}}{\leq} \max_{a \in \mathcal{A}} \Delta_a \sum_{t=0}^{n} \tau_t(\pi) \leq \sum_{a \in \mathcal{A}} \Delta_a \sum_{t=0}^{n} \tau_t(\pi) \leq \text{"Our bound"}$$

Lemma 3.16: Two armed regret bound ETC

Consider a two armed bandit model ν, the ETC and $\Delta \leq 1$ then

$$\exists C > 0 : \quad R_n(\pi) \leq \Delta + C\sqrt{n}.$$

3.5 Boltzmann exploration

In Boltzmann exploration the Idea is to choose an arm with a positive probability connected to the estimated Q-Value. For lower Q-values the choice of selection is lower and for higher Q-values it is higher. I.e. Boltzman is a mapping $\mathbb{R}^K \to (0, 1)$.

Definition 3.17: Boltzman Distribution

For $x, \theta \in \mathbb{R}^d$ we call $SM(\theta, x)$ the Boltzmann distribution if its probability weights are given by

$$p_k := \frac{e^{\theta_k x_k}}{\sum_{i=1}^{d} e^{\theta_i x_i}}, \quad k = 1, ..., d.$$

θ is often called inverse temperature. θ has to be chosen positive in order for the logic to hold, that the

largest entry of the vector x gets the highest probability. Else it would be reversed. If θ is zero, then we have the uniform distribution. As we increase θ, we explore less and less and exploit instead more.

Algorithm 5: Simple Boltzmann exploration

Data : bandit model ν, vector $\hat{Q}$, parameter θ
Initialise $T_a = 0$ for all a;
while $t \leq n$ **do**
  Sample A_t from $\mathrm{SM}(\theta, \hat{Q})$;
  Obtain reward X_t by playing arm A_t;
  $T_{A_t} = T_{A_t} + 1$;
  $\hat{Q}_{A_t} = \hat{Q}_{A_t} + \frac{1}{T_{A_t}}(X_t - \hat{Q}_{A_t})$;

Result: actions $A_1, \ldots, A_n$ and rewards $X_1, \ldots, X_n$

As the performance of this algorithm is clearly highly depended on the choice of θ, the following Lemma gives us a very important way of representing the Boltzman distribution, such that the choice of θ will become quite clear. Constant θ is a bad choice. We do not go into detail.

Lemma 3.18

Consider $x, \theta \in \mathbb{R}^d$ and $g_1, ..., g_d \overset{iid}{\sim} Gumble$ with scale 1 and mode 0, then

$$SM(\theta, x) \overset{(d)}{=} argmax_i\{\theta_i x_i + g_i\}.$$

Note that the Gumble distribution looks as follows: $F(t) := e^{-e^{-t}}$.

Proof. **1. Step:**

Let E_1, E_2 be two independent exponential random variables with parameter λ_1, λ_2, then

$$\begin{aligned}
\mathbb{P}(E_1 \leq E_2) &= \mathbb{E}[\mathbf{1}_{E_1 \leq E_2}] \\
&= \int_\Omega \mathbf{1}_{E_1 \leq E_2} d\mathbb{P}^{(E_1,E_2)} \\
&= \int_{0 \leq x_1 \leq x_2} d\mathbb{P}^{(E_1,E_2)}(x_1, x_2) \\
&= \int_{0 \leq x_1 \leq x_2} f_{(E_1,E_2)}(x_1, x_2) d(x_1, x_2) \\
&= \int_{0 \leq x_1 \leq x_2} f_{E_1}(x_1) f_{E_2}(x_2) d(x_1, x_2) \\
&= \int_0^\infty \int_0^{x_2} f_{E_1}(x_1) f_{E_2}(x_2) dx_1 dx_2 \\
&= \int_0^\infty \int_0^{x_2} \lambda_1 e^{-\lambda_1 x_1} \lambda_2 e^{-\lambda_2 x_2} dx_1 dx_2 \\
&= \lambda_1 \lambda_2 \int_0^\infty e^{-\lambda_1 x_1} \int_0^{x_2} e^{-\lambda_2 x_2} dx_1 dx_2 \\
&= \lambda_1 \lambda_2 \int_0^\infty e^{-\lambda_1 x_1} \frac{1}{\lambda_1}(1 - e^{-\lambda_1 x_2}) dx_2 \\
&= \lambda_2 \int_0^\infty e^{-\lambda_1 x_1} - e^{-(\lambda_1+\lambda_2) x_2} dx_2 \\
&= 1 - \frac{\lambda_2}{\lambda_1 + \lambda_2} = \frac{\lambda_1}{\lambda_1 + \lambda_2}
\end{aligned}$$

Now we can apply this and calculate the distribution of independent $E_1, ..., E_n$ exponential random variables. Note that $\min_{i=1,...,n} E_i \sim Exp(\sum_{i=1}^n \lambda_i)$ from Stochastik 1. Thies yields

$$\mathbb{P}(argmin_{j \leq n} E_j = i) = \mathbb{P}(\forall k \neq i : E_i \leq E_k) = \mathbb{P}(E_i \leq \min_{k \neq i} E_k) = \frac{\lambda_i}{\lambda_i + \sum_{k \neq i} \lambda_k}.$$

Now if $X \sim Exp(1)$ it follows that $Z := -\log(X) \sim Gumbel(0, 1)$, because

$$\mathbb{P}(Z \leq t) = \mathbb{P}(X \geq e^{-t}) = 1 - 1 + e^{-e^{-t}}, \quad t \in \mathbb{R}.$$

For every exponentially distributed random variable $E_1 \sim Exp(1)$ we have

$$\mathbb{P}(cE_1 \leq t) = 1 - e^{-t/c}, \quad c \in \mathbb{R},$$

i.e. $cE_1 \sim Exp(1/c)$. Using this we get for every $c \in \mathbb{R}$ and $g \sim Gumbel(0, 1)$

$$c + g \sim c - \log(E_1) = -(\log(e^{-c}) + \log(E_1)) = -\log(e^{-c} E_1) \sim -\log(E_{e^c}),$$

where $E_{e^c} \sim Exp(e^c)$. Thus we cam finally show the assertion

$$\begin{aligned}
\mathbb{P}(argmax_{k \leq n}(\theta_k x_k + g_k) = i) &= \mathbb{P}(argmax_{k \leq n} - \log(E_{e^{\theta_k x_k}}) = i) \\
&= \mathbb{P}(argmin_{k \leq n} E_{e^{\theta_k x_k}} = i) \\
&= \frac{e^{\theta_i x_i}}{\sum_{k \leq n} e^{\theta_k x_k}} = \mathbb{P}(SM(\theta, x) = i).
\end{aligned}$$

□

Therefore, we can now sample from the Boltzmann distribution if we take for $g_1, ..., g_d \overset{iid}{\sim} Gumble$

$$A_t \sim argmax_a\{\theta \hat{Q}_a(t-1) + g_a\} = argmax_a\{\hat{Q}_a(t-1) + \theta^{-1} g_a\}$$

This looks exactly like the UCB. In an article –Cesa-Bianci et al– it was shown that if we choose $\theta_n^{-1} = \sqrt{\frac{C}{T_a(n)}}$, then we get UCB.

3.6 Policy Gradient for Stochastic Bandits

We now have a different approach, in finding optimal policies that are more closely related to reinforcement learning. We no longer look at learning strategies that minimize regret. The goal here is to find the best arm as quickly as possible, regardless the regret.

Definition 3.19: Policy and Value of a Policy

A distribution π on $\mathcal{A}$ is called a policy and the expected reward, when playing this policy

$$V(\pi) := Q_\pi := \sum_{a \in \mathcal{A}} \pi(a) Q_a$$

is called the value of the policy.

Note, in the chapters above we had learning strategies and not policies. A policy π^* is called optimal if

$$\pi^* \in argmax_\pi V(\pi).$$

Because

$$V(\pi) = \sum_{a \in \mathcal{A}} \pi(a) Q_a \leq Q_{a^*},$$

the optimal policy is always such that it only plays the best arm, i.e. $(0, ..., 0, 1, 0, ..., 0)$. If there are multiple arms, that are the best, then it does not matter which of them is played.
We call a learning strategy value based, if one first estimates Q or V and then from this defines the policy in the next step.

Definition 3.20: Parameterized Family of Policies

If $\Theta \subseteq \mathbb{R}^d$ then we call the set of probability distributions $\{\pi_\theta \mid \theta \in \Theta\}$ on $\mathcal{A}$ a parameterized Family of Policies.

The approach is as follows. We define the mapping over θ

$$\theta \mapsto J(\theta) := Q_{\pi_\theta} := \sum_{a \in \mathcal{A}} Q_a \pi_\theta(a)$$

and then we define gradient descent (direction of steepest descent)

$$\forall n \in \mathbb{N}: \quad \theta_{n+1} = \theta_n + \alpha \nabla J(\theta).$$

Under assumptions on J as convexity, then this method converges to a global minimum.
The optimal policy is clearly the one that chooses with probability one the best arm, i.e. $\pi^*(\cdot) := \delta_{a^*}(\cdot)$. Thus the parametrized family should be able to approximate dirac measures. In practice we have the following three issues

- How can we find a good parametrization, such that the parameterized policies contains the optimal policy, but is not too large.
- If J is unknow, how can we do gradient descent?
- Even if we know J how can we do gradient descent such that we reach an global maximum.

We will now answer the second question. We will now show that we can write ∇J as an expectation and then we can sample from it.

Remark 3.6

We can write

$$\begin{aligned}
\nabla J(\theta) &= \nabla \sum_{a\in\mathcal{A}} Q_a \pi_\theta(a) \\
&= \sum_{a\in\mathcal{A}} Q_a \nabla \pi_\theta(a) \frac{\pi_\theta(a)}{\pi_\theta(a)} \\
&= \sum_{a\in\mathcal{A}} Q_a \nabla \log(\pi_\theta(a)) \pi_\theta(a) \\
&= \mathbb{E}[Q_A \nabla \log(\pi_\theta(A))] \\
&= \mathbb{E}[\mathbb{E}[X_A \mid A] \nabla \log(\pi_\theta(A))] \\
&= \mathbb{E}[\mathbb{E}[X_A \nabla \log(\pi_\theta(A)) \mid A]] \\
&= \mathbb{E}[X_A \nabla \log(\pi_\theta(A))]
\end{aligned}$$

Definition 3.21: Score Function

We call the mapping for every $a \in \mathcal{A}$

$$(a, \theta) \mapsto \nabla \log(\pi_\theta(a))$$

the score function.

We can do gradient descent, either by taking one sample

$$\theta_{n+1} = \theta_n + \alpha X_{A_n} \nabla \log(\pi_\theta(A_n)), \quad n \in \mathbb{N}$$

or multiple

$$\theta_{n+1} = \theta_n + \alpha \frac{1}{N} \sum_{i=1}^{N} X^i_{A^i_n} \nabla \log(\pi_\theta(A^i_n)), \quad n \in \mathbb{N}.$$

Now we consider the first question.

Remark 3.7

A direct parametrization is if

$$(\pi_\theta(a_1), ..., \pi_\theta(a_d)) = (\theta_1, ..., \theta_d)$$

This is tabular softmax???
We use the softmax parametrization, i.e. we have $\theta \in \mathbb{R}^d$ and then use the $SM(\theta, 1)$. This is also due to the fact that when we rewrote the gradient of J, we have a $\log(\pi_\theta(A))$ inside of the expectation, i.e.

$$\log(\pi_\theta(A)) = \log(e^{\theta_A}) - \log(\sum_{a\in\mathcal{A}} e^{\theta_a}) = \theta_A - \log(\sum_{a\in\mathcal{A}} e^{\theta_a}).$$

The only problem with softmax is that we cannot find a θ such that $\pi_\theta(a) = (0, ..., 0, 1, 0, ...0)$ of this form, i.e. the softmax is only a bijective mapping between

$$\mathbb{R}^d \to (0, 1)^d.$$

We can only approximate dirac policies, by sending one θ_a to infinity (faster than the others is also sufficient).
In the later chapters, the parametrization will be a neural network.

In the next remark we will shortly explain why we call this method reinforcement learning.

Remark 3.8

We will now look at how the score function looks like, when we use softmax parametrization and what this does to the gradient descent. We have for every $a \in \mathcal{A}$ and the i^{th} entry of the score function vector, with $|\mathcal{A}| = K$

$$(\nabla \log(\pi_\theta(a)))_{i=1,...,K} = (\nabla(\theta_a - \log(\sum_{a\in\mathcal{A}} e^{\theta_a})))_{i=1,...,K} = (\mathbf{1}_{a=i} - \frac{e^{\theta_i}}{\sum_{a\in\mathcal{A}} e^{\theta_a}})_{i=1,...,K} = (\mathbf{1}_{a=i} - \pi_\theta(i))_{i=1,...,K}$$

If we plug this into gradient descent scheme with one sample, where we took at step n action $a = k$, we get update

$$\begin{pmatrix} \theta_{1,n+1} \\ \cdots \\ \theta_{k,n+1} \\ \cdots \\ \theta_{K,n+1} \end{pmatrix} = \theta_n + \alpha X_k \cdot \begin{pmatrix} -\pi_{\theta_n}(1) \\ \cdots \\ 1 - \pi_{\theta_n}(k) \\ \cdots \\ -\pi_{\theta_n}(K) \end{pmatrix} = \begin{pmatrix} \theta_{1,n} - \alpha X_k \pi_{\theta_n}(1) \\ \cdots \\ \theta_{k,n} + \alpha X_k (1 - \pi_{\theta_n}(k)) \\ \cdots \\ \theta_{K,n} - \alpha X_k \pi_{\theta_n}(K) \end{pmatrix}$$

If playing arm $a = k$ gives a positive reward X_k then θ_k is increased and all other θ_i are decreased in the next round. If arm $a = k$ yields a negative reward, this behavior is reversed.
The problem is though, that positive rewards do not nessecarly translate to good action. If we only have positively distributed arms, then all arms are enforced if they are played, but the best arms are more strongly enforced, i.e. they go faster to infinity. This leads to the same result.
So when calculating the gradient descent method, we found an algorithm that could have been found plausibly.

We can also add, as in greedy algorithms, a baseline, such that we can more easily find, what is a "good"

arm. I.e. we have

$$\theta_{n+1} = \theta_n + \alpha(X_a - b)\nabla \log(\pi_{\theta_n}(a)).$$

We will now show why this choice is equivalent to using no baseline.

Remark 3.9

We have for the gradient of the value function for a baseline $b \in \mathbb{R}$

$$\nabla J(\theta) = \mathbb{E}[X_A \nabla \log(\pi_\theta(A))] = \mathbb{E}[X_A \nabla \log(\pi_\theta(A))] - \underbrace{\mathbb{E}[b\nabla \log(\pi_\theta(A))]}_{=0} = \mathbb{E}[(X_A - b)\nabla \log(\pi_\theta(A))].$$

We only have to show the equality with zero. This is due to

$$\begin{aligned}
\mathbb{E}[b\nabla \log(\pi_\theta(A))] &= b\sum_{a\in\mathcal{A}} \nabla \log(\pi_\theta(a))\pi_\theta(a) \\
&= b\sum_{a\in\mathcal{A}} \nabla \pi_\theta(a)\frac{\pi_\theta(a)}{\pi_\theta(a)} \\
&= b\nabla \sum_{a\in\mathcal{A}} \pi_\theta(a) \\
&= b\nabla 1 = 0
\end{aligned}$$

This approach (can) drastically reduces the committal behavior. The question is how to choose b. There are two approaches on how to choose plausibly:

- Choose b such that is reduces the variance of the estimation of the expectation.
- Choose for the update at time $n+1$ the baseline as $b = J(\theta_n) = V(\pi_{\theta_n})$ as the value of the current policy.

The first choice is very reasonable, because we want the estimation to be more stable, but as it turns out, this is a pathwise not so good choice. This is because the second choice is better??? In the second choice we only enforce an arm, that has a better reward, than the current value of the policy. The problem here is that we do not know the value function of the current policy, as this again is an expectation, i.e. we need to estimate it. This will be later called Actor-Critic. First we calculate the value of the current policy and then improve it.

Remark 3.10

One can show that choosing the baseline as

$$b^* := \frac{\mathbb{E}[X_A \|\nabla \log(\pi_\theta(A))\|_2^2]}{\mathbb{E}[\|\nabla \log(\pi_\theta(A))\|_2^2]}$$

minimizes the variance for the unbiased estimator

$$\forall b \in \mathbb{R}: \quad (X_A - b)\nabla \log(\pi_\theta(A)), \quad A \sim \pi_\theta$$

of $J(\theta)$.

Remark 3.11

Actor-Critic methods are as follows: The Critic samples from $V(\pi_{\theta_t}) = b$. Then using this baseline, the actor does gradient descent and updates θ_{t+1}.

CHAPTER 2

MARKOV DECISION MODEL

1 Markov Decision Problem

We will first do stochastic control theory and then do reinforcement learning.

1.1 Markov-Chains

A Markov Chain is stochastic process $(S_t)_{t\in\mathbb{N}}$ taking values in $\mathcal{S}$ and on $(\Omega, \mathcal{F}, \mathbb{P})$ such that

$$\mathbb{P}(S_{t+1} = s_{t+1} \mid S_0 = s_0, ..., S_t = s_t) = \mathbb{P}(S_{t+1} = s_{t+1} \mid S_t = s_t).$$

for all $s_0, ..., s_{t+1} \in \mathcal{S}$ and all $t \in \mathbb{N}$.

Theorem 1.1: Path Probabilities

Given a Markov Chain $(S_t)_{t\in\mathbb{N}}$ on $\mathcal{S}$ and a stochastic matrix $P = (p(x, y))_{x,y\in\mathcal{S}}$ and an initial distribution $\mu : \mathcal{S} \to [0, 1]$, then

$$(S_t)_{t\in\mathbb{N}} \text{ is a } (\mu, P)\text{-Markov Chain } \iff \mathbb{P}(S_0 = s_0, ..., S_t = s_t) = \mu(s_o)p(s_0, s_1) \cdot ... \cdot p(s_{t-1}, s_t),$$

for all $t \in \mathbb{N}$ and $s_0, ..., s_t \in \mathcal{S}$.

Using this property we can easily calculate a lot of different probabilities. For instance for $t, k \in \mathbb{N}$

$$\mathbb{P}(S_{t+1} = s_{t+1}, ..., S_{t+k} = s_{t+k} \mid S_t = s_t)$$

for all $s_t, ..., s_{t+k} \in \mathcal{S}$.
Further, it is imperative to notice that the shifted Chain, is again a Markov Chain, i.e. we define $\tilde{\mathbb{P}} := \mathbb{P}(\cdot \mid S_n = s_n)$, then the chain $(\tilde{S}_t)_{t\in\mathbb{N}} = (S_{t+n})_{t\in\mathbb{N}}$ is again a Markov chain on $(\Omega, \mathcal{F}, \tilde{\mathbb{P}})$.

1.2 Markov-Reward-Chains

It is important to note, that this is not going to be the same as two dimensional Markov-Chain!

Definition 1.2: Markov Reward Chain

This is a Markov-Chain $(S_t)_{t\in\mathbb{N}}$ with another coordinate $(R_t)_{t\in\mathbb{N}}$, where

$$\mathbb{P}_\mu(S_{t+1} = x_{t+1}, R_t = r_t \mid S_0 = s_0, ..., S_t = s_t)$$

which is equivalent to

$$\mathbb{P}_\mu(R_{t+1} = r_{t+1}, S_{t+1} = x_{t+1} \mid S_0 = s_0, ..., S_t = s_t)$$

but the first formulation will better fit to our formulation with MDPs later.
Like in soccer, you do not always get a reward, only when you score the goal. Shooting at the goal is like tossing a coin on wether the goal keeper catches the ball.

Remark 1.1

There exists a transition/reward kernel p on $(\mathcal{R} \times \mathcal{S}) \times \mathcal{S}$ that define the transition from a state to the next state and reward from the previous state $p(r, s'; s)$.

Remark 1.2

If we would condition on the rewards in the Markov rewards process, then we would just have a two dimensional markov chain! This is very improtant!

Similiar to the factorization in Markov Chains, there a exists a similar (but different) factorization of the Markov reward chain, as we do not condition on the rewards.

Remark 1.3

The path probabilites of the Markov-Reward chain is given by

$$\mathbb{P}_\mu(S_0 = s_0, R_0 = r_0, ..., S_n = s_n, R_n = r_n) = \mu(s_0)p(r_0, s_1; s_0) \cdot ... \cdot p(r_{n-1}, s_n; s_{n-1})$$

Similiar as in Markov Chains, we can show that the shifted Markov-Reward-Chain is again a Markov-Reward Chain.

Remark 1.4

In the case that

$$\mathbb{P}(S_n = s') > 0$$

and we define $\tilde{\mathbb{P}} := \mathbb{P}(\cdot \mid S_n = s')$, then we have that for all $n \in \mathbb{N}$

$$(\tilde{S}_t, \tilde{R}_t)_{t\in\mathbb{N}} := (S_{t+n}, R_{t+n})_{t\in\mathbb{N}}$$

is again a Markov-Reward Chain on $(\Omega, \mathcal{F}, \tilde{\mathbb{P}})$.

1.3 Markov-Decision Process

In the situation of Markov decision Process (MDPs) it is slithgtly more complicated, as we now not only observe the game, but take part in it. The goal will be to find a distribution which we call policy, that maximizes the expected reward of the game.

Definition 1.3: Markov-Decision-Model

A MDM is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ consisiting of the following

- The measurable set $(\mathcal{S}, \bar{\mathcal{S}})$, called state space
- The measurable set $(\mathcal{A}, \bar{\mathcal{A}})$ called action space, where for every $s \in \mathcal{S}$ the space $(\mathcal{A}_s, \bar{\mathcal{A}}_s)$ is called action space of the state s and

$$\mathcal{A} = \bigcup_{s \in \mathcal{S}} \mathcal{A}_s, \quad \text{and} \quad \bar{\mathcal{A}} = \sigma(\bigcup_{s \in \mathcal{S}} \bar{\mathcal{A}}_s).$$

- The reward space is defined as a measurable set $\mathcal{R} \subseteq \mathbb{R}$ always containing $0 \in \mathcal{R}$. Its restricted Borel-sigma-Algebra is defined as

$$\bar{\mathcal{R}} := \{\mathcal{R} \cap B \mid B \in \mathcal{B}(\mathbb{R})\}.$$

- A function

$$p : (\bar{\mathcal{S}} \times \bar{\mathcal{R}}) \times (\mathcal{S} \times \mathcal{A}) \to [0,1], (B, (s,a)) \mapsto p(B; s, a)$$

  is called transition function if p is a Markov Kernel on $\bar{\mathcal{S}} \otimes \bar{\mathcal{R}} \times (\mathcal{S} \times \mathcal{A})$, i.e.

  - $(s,a) \mapsto p(B; s, a)$ is $(\bar{\mathcal{S}} \otimes \bar{\mathcal{A}}) - \mathcal{B}([0,1])$ measurable for all B.
  - $B \mapsto p(B; s, a)$ is a probability measure on $\bar{\mathcal{S}} \times \bar{\mathcal{R}}$ for all (s,a).

  A MDP is called discrete if $\mathcal{S}, \mathcal{A}, \mathcal{R}$ are discrete. Then the sigma algebras are just the respective power sets.

The transition function kernel only describes the transition of $(s,a) \mapsto (s', r)$, but does not describe how the next action a' is chosen. A MDP where there is only one action (you cannot make any decisions) then this is just a markov chain.
In order not to confuse conditional probabilities with Markov kernels, we will use the notation ";" for the transition function and "|" exclusively for conditional probabilites.

Definition 1.4: Policy

For a MDP $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ a policy is defined as

- an initial distribution π_0 on $(\bar{\mathcal{A}}) \times \mathcal{S}$
- a sequence of probability kernels on $\pi := (\pi_t)_{t \in \mathbb{N}}$ on $\bar{\mathcal{A}} \times ((\mathcal{S} \times \mathcal{A})^{t-1} \times \mathcal{S})$ such that

$$\forall (s_0, s_0, ..., s_t) \in (\mathcal{S} \times \mathcal{A})^{t-1} \times \mathcal{S} : \quad \pi_t(\mathcal{A}_{s_t}; s_0, a_0, ..., s_t) = 1$$

  is a probability measure on $\mathcal{A}_s$.

The set Π is the set of all probability measures.

The second point ensures that only actions from $\mathcal{A}_{s_t}$ can be played. **From now on all sets are discrete and the σ-algebras are powersets.**

Definition 1.5: Some Definitions

For a MDP we define

- $p(s'; s, a) := p(\{s'\} \times \mathcal{R}; s, a)$
- $p(r; s, a) := p(\mathcal{S} \times \{r\}; s, a)$
- Expected reward given state action tuple (s, a)

$$r(s, a) := \sum_{r \in \mathcal{R}} r \cdot p(r; s, a)$$

- Expected reward given state action tuple (s, a) and next state is s'

$$r(s, a, s') := \sum_{r \in \mathcal{R}} r \frac{p(s', r; s, a)}{p(s'; s, a)}$$

The above are only defined when the demonitor is positive.

We can now define a stochastic process on $\mathcal{S} \times \mathcal{A} \times \mathcal{R}$ whose dynamic is solely dependend on the transition function p and policy π (and some initial distribution).

Remark 1.5

Backround:
Let $I = \mathbb{N}$ we then define for $\emptyset \neq K, J \subseteq I$ the finite projections

$$\pi_K^J : E^J \to E^K, (X_t)_{t \in J} \mapsto \pi_K^J((X_t)_{t \in J}) := (X_t)_{t \in K}$$

and

$$\pi^J : E^I \to E^J, (X_t)_{t \in I} \mapsto \pi^J((X_t)_{t \in I}) := (X_t)_{t \in J}.$$

Further for a measure Q_J on $(E_j, \epsilon^{\otimes J})$ we call the family $\{Q_J \mid J \subseteq I \text{ finite}\}$ consistent if

$$\forall J_1 \subseteq J_2 \subseteq I : \quad Q_{J_1} = Q_{J_2} \circ (\pi_{J_1}^{J_2})^{-1}, \quad \emptyset \neq J_1, J_2.$$

Finally, the Kolmogorov extension theorem tells us that for a polisch space (E, ϵ) and a consitent family $\{Q_J \mid J \subseteq I \text{ finite}\}$ there exists a unique probability measure Q on $(E^I, \epsilon^{\otimes I})$ such that

$$Q_J = Q \circ (\pi_J)^{-1}, \quad \emptyset \neq J \subseteq I.$$

Theorem 1.6: Existence of finite MDPs

Let $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$ be a MDP, π a policy and μ a distribution on $\mathcal{S}$. Then there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P}_\mu)$ carrying a stochastic process $(S_t, A_t, R_t)_{t\in\mathbb{N}_0}$ with values in $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ such that for all $t \in \mathbb{N}$ holds that

- $\mathbb{P}^\pi_\mu(S_0 = s_0, A_0 = a_0) = \mu(s_0)\pi_0(a_0; s_0)$
- $\mathbb{P}^\pi_\mu(A_t = a_t \mid S_0 = s_0, A_0 = a_0, ..., S_t = s_t) = \pi_t(a_t; s_0, a_0, ..., s_t)$
- $\mathbb{P}^\pi_\mu(S_{t+1}, R_t = r_t \mid S_t = s_t, A_t = a_t) = p(s_{t+1}, r_t; s_t, a_t)$

We will write in future $\mathbb{P}^\pi_\mu = \mathbb{P}$.

Proof. $T < \infty$:
The probability space is the set of all trajectories up to time T

$$\Omega_T := \{\omega := (s_0, a_0, r_0, ..., s_T, a_T, r_T) \in (\mathcal{S} \times \mathcal{A} \times \mathcal{R})^T\}$$

and its σ-algebra $\mathcal{F}_T$ its respective power set. We then define

$$\mathbb{P}_T(\{(s_0, a_0, r_0, ..., s_T, a_T, r_T)\}) := \mu(s_0) \cdot \pi_0(a_0; s_0) \cdot \prod_{i=1}^{T} p(s_i, r_{i-1}; s_{i-1}, a_{i-1}) \cdot \pi_i(a_i; s_0, a_0, ..., s_i) \cdot p(\mathcal{S} \times \{r_T\}; s_T, a_T)$$

We now need to show that $\mathbb{P}_T$ defined on the singleton is indeed a probability measure. This is shown in the excersise.
Now $T = \infty$:
Then using the remark above, we need to show that $\mathbb{P}_T$ is a consistent family. First, define $\Omega := (\mathcal{S} \times \mathcal{R} \times \mathcal{A})^\infty$ and $\mathcal{F}$ the respective cylinder σ-algebra. We define the two projections

$$\pi_T^{T+1} : \Omega^{T+1} \to \Omega^T, \omega \mapsto \omega\big|_T$$

and

$$\pi_T : \Omega \to \Omega^T, \omega \mapsto \Omega^T, \omega \mapsto \omega\big|_T.$$

If we show $\mathbb{P}_T = \mathbb{P}_{T+1} \circ (\pi_T^{T+1})^{-1}$ for all $T \in \mathbb{N}$, then $(\mathbb{P}_T)_{T\in\mathbb{N}}$ is a consitent family and there exists a unique probability measure $\mathbb{P}$ on $(\Omega, \mathcal{F})$ such that

$$\mathbb{P}_T = \mathbb{P} \circ (\pi_T)^{-1}, \quad \forall T \in \mathbb{N}.$$

We then define $\mathbb{P}^\pi_\mu := \mathbb{P}$ to emphasize the depends on the policy and the initial distribution μ.

Consistency:

$$
\begin{aligned}
&\mathbb{P}_{T+1}((\pi_T^{T+1})^{-1}(s_0,a_0,r_0,...,s_T,a_T,r_T))\\
&=\mathbb{P}_{T+1}(\bigcup_{s\in\mathcal{S}}\bigcup_{a\in\mathcal{A}}\bigcup_{r\in\mathcal{R}}(s_0,a_0,r_0,...,s_T,a_T,r_T,s,a,r)\\
&=\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{r\in\mathcal{R}}\mathbb{P}_{T+1}((s_0,a_0,r_0,...,s_T,a_T,r_T,s,a,r)\\
&=\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{r\in\mathcal{R}}\mu(s_0)\cdot\pi_0(a_0;s_0)\cdot\prod_{i=1}^{T}p(s_i,r_{i-1};s_{i-1},a_{i-1})\cdot\pi_i(a_i;s_0,a_0,...,s_i)\\
&\quad\cdot p(s,r_T;s_T,a_T)\cdot\pi_{T+1}(a;s_0,a_0,...,s)\cdot p(\mathcal{S}\times\{r\};s,a)\\
&=\mu(s_0)\cdot\pi_0(a_0;s_0)\cdot\prod_{i=1}^{T}p(s_i,r_{i-1};s_{i-1},a_{i-1})\cdot\pi_i(a_i;s_0,a_0,...,s_i)\\
&\qquad\cdot\underbrace{\sum_{s\in\mathcal{S}}p(s,r_T;s_T,a_T)}_{=p(\mathcal{S}\times\{r_T\};s_T,a_T)}\cdot\underbrace{\sum_{a\in\mathcal{A}}\pi_{T+1}(a;s_0,a_0,...,s)}_{=1}\cdot\underbrace{\sum_{r\in\mathcal{R}}p(\mathcal{S}\times\{r\};s,a)}_{=1}\\
&\qquad=\mathbb{P}_T((s_0,a_0,r_0,...,s_T,a_T,r_T))
\end{aligned}
$$

Canonical Construction:
Similiar to the one dimensional case where for every distribution function F there exists a random variable X defined on $(\Omega,\mathcal{F},\mathbb{P})$ such that $X\sim F$, where we used that $X(\omega)=\omega$, we can say that there exists a stochastic process $(S_t,A_t,R_t)_{t\in\mathbb{N}}$ on $(\Omega,\mathcal{F},\mathbb{P})$ such that $(S_t,A_t,R_t)_{t\in\mathbb{N}}\sim\mathbb{P}$, where $(S_t,A_t,R_t)(\omega)=(s_t,a_t,r_t)$ is the identity mapping. Properities:
We show the second condition:

$$
\begin{aligned}
&\mathbb{P}(A_t=a_t\mid S_0=s_0,A_0,...,S_t=s_t)=\frac{\mathbb{P}(S_0=s_0,A_0,...,S_t=s_t,A_t=a_t)}{\mathbb{P}(S_0=s_0,A_0,...,S_t=s_t)}\\
&=\frac{\sum_{r_0\in\mathcal{R}}...\sum_{r_t\in\mathcal{R}}\mathbb{P}(S_0=s_0,A_0,R_0=r_0,...,S_t=s_t,A_t=a_t,R_t=r_t)}{\sum_{r_0\in\mathcal{R}}...\sum_{r_t\in\mathcal{R}}\sum_{a_t\in\mathcal{A}}\mathbb{P}(S_0=s_0,A_0,R_0=r_0,...,S_t=s_t,A_t=a_t,R_t=r_t)}\\
&=\frac{\sum_{r_0\in\mathcal{R}}...\sum_{r_t\in\mathcal{R}}\mu(s_0)\cdot\pi_0(a_0;s_0)\cdot\prod_{i=1}^{t}p(s_i,r_{i-1};s_{i-1},a_{i-1})\cdot\pi_i(a_i;s_0,a_0,...,s_i)\cdot p(\mathcal{S}\times\{r_t\};s_t,a_t)}{\sum_{r_0\in\mathcal{R}}...\sum_{r_t\in\mathcal{R}}\sum_{a_t\in\mathcal{A}}\mu(s_0)\cdot\pi_0(a_0;s_0)\cdot\prod_{i=1}^{t}p(s_i,r_{i-1};s_{i-1},a_{i-1})\cdot\pi_i(a_i;s_0,a_0,...,s_i)\cdot p(\mathcal{S}\times\{r_t\};s_t,a_t)}\\
&=\frac{\sum_{r_0\in\mathcal{R}}...\sum_{r_{t-1}\in\mathcal{R}}\mu(s_0)\cdot\pi_0(a_0;s_0)\cdot\prod_{i=1}^{t-1}p(s_i,r_{i-1};s_{i-1},a_{i-1})\cdot\pi_i(a_i;s_0,a_0,...,s_i)}{\sum_{r_0\in\mathcal{R}}...\sum_{r_{t-1}\in\mathcal{R}}\mu(s_0)\cdot\pi_0(a_0;s_0)\cdot\prod_{i=1}^{t-1}p(s_i,r_{i-1};s_{i-1},a_{i-1})\cdot\pi_i(a_i;s_0,a_0,...,s_i)\cdot p(\mathcal{S}\times\{r_t\};s_t,a_t)}\\
&\cdot\underbrace{\frac{\sum_{r_t\in\mathcal{R}}p(s_t,r_{t-1};s_{t-1},a_{t-1})\cdot\pi_t(a_t;s_0,a_0,...,s_t)\cdot p(\mathcal{S}\times\{r_t\};s_t,a_t)}{\sum_{r_t\in\mathcal{R}}p(s_t,r_{t-1};s_{t-1},a_{t-1})\cdot\underbrace{\sum_{a_t\in\mathcal{A}}\pi_t(a_t;s_0,a_0,...,s_t)}_{=1}\cdot p(\mathcal{S}\times\{r_t\};s_t,a_t)}}_{=\pi_t(a_t;s_0,a_0,...,s_t)}\\
&=\pi_t(a_t;s_0,a_0,...,s_t)
\end{aligned}
$$

□

Definition 1.7: Markov Decision Process

Given a MDM $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$, a policy π and an initial distribution μ on $\mathcal{S}$, then the stochastic process $(S_t, A_t)_{t\in\mathbb{N}}$ on $(\Omega, \mathcal{F}, \mathbb{P}_\mu)$ is called discrete time Markov decision process and $(R_t)_{t\in\mathbb{N}_0}$ the corresponding reward process.
If the initial distribution is the dirac measure δ_s then we will write $\mathbb{P}_s^\pi := \mathbb{P}_{\delta_s}^\pi$.

Remark 1.6

The derivation of the path probabilities is and will be very important in the future

$$\begin{aligned}&\mathbb{P}_\mu^\pi(S_0 = s_0, A_0 = a_0, R_0 = r_0, ..., S_T = s_t, A_t = a_t, R_t = r_t)\\ &= \mu(s_0)\pi_0(a_0; s_0)\left(\prod_{i=1}^{t} p(s_i, r_{i-1}; s_{i-1}, a_{i-1}) \cdot \pi_i(a_i; s_0, a_0, ..., s_i)\right) \cdot p(\mathcal{S} \times \{r_t\}; s_t, a_t).\end{aligned}$$

In literature one does not differentiate between the Markov decision Model and the Markov Decision Process! For us, the Markov decision process is the Markov decision model plus the policy and an initial distribution.
We are also interested also in path probabilies started at a specific time $t \in \mathbb{N}$, i.e. for $T > t$ we have

$$\begin{aligned}&\mathbb{P}(S_t = s_t, A_t = a_t, R_t = r_t, ..., S_T = s_T, A_T = a_T, R_T = r_T \mid S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1})\\ &= \frac{\mathbb{P}(S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, S_t = s_t, A_t = a_t, R_t = r_t, ..., S_T = s_T, A_T = a_T, R_T = r_T)}{\mathbb{P}(S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1})}\\ &= \frac{\sum_{s_0\in\mathcal{S}}\sum_{a_0\in\mathcal{A}_{s_0}}\sum_{r_0\in\mathcal{R}}\cdots\sum_{s_{t-2}\in\mathcal{S}}\sum_{a_{t-2}\in\mathcal{A}_{s_{t-2}}}\sum_{r_{t-2}\in\mathcal{R}}\mathbb{P}(S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, R_{t-1} = r_{t-1}, ..., S_T = s_T, A_T = a_T, R_T = r_T)}{\sum_{s_0\in\mathcal{S}}\sum_{a_0\in\mathcal{A}_{s_0}}\sum_{r_0\in\mathcal{R}}\cdots\sum_{s_{t-2}\in\mathcal{S}}\sum_{a_{t-2}\in\mathcal{A}_{s_{t-2}}}\sum_{r_{t-2}\in\mathcal{R}}\mathbb{P}(S_0 = s_0, A_0 = a_0, R_0 = r_0, ...S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1})}\\ &= \sum_{r_{t-2}\in\mathcal{R}} p(\{s_t\} \times \mathcal{R}; s_{t-1}a_{t-1}) \cdot \pi_t(a_t; s_0a_0, ..., a_{t-1}, s_t) \prod_{i=t+1}^{T} p(s_i, r_{i-1}; s_{i-1}, a_{i-1}) \cdot \pi_i(a_i; s_0a_0, ..., a_{i-1}, s_i) \cdot p(\mathcal{S} \times \{r_T\}; s_T, a_T)\end{aligned}$$

and similarly

$$\begin{aligned}&\mathbb{P}(S_t = s_t, A_t = a_t, R_t = r_t, ..., S_T = s_T, A_T = a_T, R_T = r_T \mid S_{t-1} = s_{t-1})\\ &= \sum_{a_{t-1}\in\mathcal{A}_{s_{t-1}}} \pi_{t-1}(a_{t-1}; s_0a_0, ..., a_{t-2}, s_{t-1}) \prod_{i=t}^{T} p(s_i, r_{i-1}; s_{i-1}, a_{i-1}) \cdot \pi_i(a_i; s_0a_0, ..., a_{i-1}, s_i) \cdot p(\mathcal{S} \times \{r_T\}; s_T, a_T)\end{aligned}$$

Example 1.1

The probability below looks differently if we know in which state we are, compared to the case that we do not know where we are.
Here we show that if we know in which state we are in and the transition of (s, a) to r and s' are independend (conditional independence), i.e. there exists two kernels h, q such that

$$p(s', r; s, a) := h(s'; s, a)q(r; s, a)$$

Case 1: I know where I am:

The reward r when coming from (s,a) does not depend on the state s' where it is going, i.e.

$$\begin{aligned}
&\mathbb{P}(R_t = r \mid S_t = s, A_t = a, S_{t+1} = s') \\
&= \frac{\mathbb{P}(R_t = r, S_t = s, A_t = a, S_{t+1} = s')}{\mathbb{P}(S_t = s, A_t = a, S_{t+1} = s')} \frac{\mathbb{P}(A_t = a, S_{t+1} = s')}{\mathbb{P}(A_t = a, S_{t+1} = s')} \\
&= \frac{\mathbb{P}(R_t = r, S_{t+1} = s' \mid S_t = s, A_t = a)}{\mathbb{P}(S_{t+1} = s' \mid A_t = a, S_t = s)} \\
&= \frac{p(s', r; s, a)}{p(\{s'\} \times \mathcal{R}; s, a)} \\
&= \frac{h(s'; s, a) q(r; s, a)}{h(s', s, a) \cdot 1} = q(r; s, a) = p(\mathcal{S} \times \{r\}; s, a) \\
&= \mathbb{P}(R_t = r \mid S_t = s, A_t = a)
\end{aligned}$$

Case 2: I do not know where I am:
The reward r when coming from $(\mathcal{S}, a)$ depends on the state s' where it is going, i.e.

$$\begin{aligned}
&\mathbb{P}(R_t = r \mid A_t = a, S_{t+1} = s') \\
&= \frac{\sum_{s \in \mathcal{S}} \mathbb{P}(R_t = r, S_t = s, A_t = a, S_{t+1} = s')}{\sum_{s \in \mathcal{S}} \mathbb{P}(S_t = s, A_t = a, S_{t+1} = s')} \\
&= \frac{\sum_{s \in \mathcal{S}} \mathbb{P}(R_t = r, S_{t+1} = s' \mid S_t = s, A_t = a)}{\sum_{s \in \mathcal{S}} \mathbb{P}(S_{t+1} = s' \mid A_t = a, S_t = s)} \\
&= \frac{\sum_{s \in \mathcal{S}} p(s', r; s, a)}{\sum_{s \in \mathcal{S}} p(\{s'\} \times \mathcal{R}; s, a)} \\
&= \frac{\sum_{s \in \mathcal{S}} h(s'; s, a) q(r; s, a)}{\sum_{s \in \mathcal{S}} h(s', s, a) \cdot 1}
\end{aligned}$$

Example 1.2

In many examples, the reward are deterministic functions wrt. previuos state, ation and next state, i.e. $R_t := R(S_t, A_t, S_{t+1})$, then the transition function is

$$p(s', r; s, a) := h(s'; s, a) \mathbf{1}_{R(s,a,s')=r}.$$

This will be the case in the Ice vendor example.
In the case as above, if we know where we are, then R_t and S_{t+1} are independend, i.e.

$$R(s, a, s') = R(s, a)$$

Definition 1.8: Terminating State

A state $s \in \mathcal{S}$ satisfying $p(s; s, a) = 1$ for all $a \in \mathcal{A}_s$ is called terminating sate. We define by Δ the set of all terminating states and we assume that

$$\forall s \in \Delta : \quad p(s, 0; s, a) = 1.$$

Example 1.3: Grid World

—

Windy grid World: Depending on the strength of the wind, it is not clear on how to play. Therefore, find policy that maximizes $V(\pi)$. This will be later the Bellman optimality equations.
Action Masking: Define all actions for every state, but not every action is allowed in every state.
In Grid world: $R(s,a,s') = R(s,a) = R(s) = g(s)$.

Finite time MDP. Grid world is not finite time. Different goals.

Example 1.4: Ice Vendor

—

Definition 1.9: Markov and Stationary Policies

A policy $\pi = (\pi_t)_{t\in\mathbb{N}_0} \in \Pi$ is called

i) a Markov policy if there exists a sequence of kernels $(\varphi_t)_{t\in\mathbb{N}_0}$ on $(\bar{\mathcal{A}} \times \mathcal{S})$ such that

$$\forall s_0 a_0, ..., s_t \in (\mathcal{S}\times\mathcal{A})^{t-1}\times\mathcal{S}: \quad \pi_t(\cdot; s_0, a_0, ..., s_t) = \varphi_t(\cdot; s_t).$$

The set of all Markov policies is denoted by Π_M.

- a stationary policy if there exists one kernel φ on $(\bar{\mathcal{A}} \times \mathcal{S})$ such that

$$\forall s_0 a_0, ..., s_t \in (\mathcal{S}\times\mathcal{A})^{t-1}\times\mathcal{S}: \quad \pi_t(\cdot; s_0, a_0, ..., s_t) = \varphi(\cdot; s_t).$$

The set of all stationary policies is denoted by Π_S.

- a deterministic stationary policy if there exists one kernel φ on $(\bar{\mathcal{A}} \times \mathcal{S})$ such that

$$\forall s_0 a_0, ..., s_t \in (\mathcal{S}\times\mathcal{A})^{t-1}\times\mathcal{S}: \quad \pi_t(\cdot; s_0, a_0, ..., s_t) = \varphi(\cdot; s_t).$$

and only takes values in $\{0,1\}$. The set of all stationary policies is denoted by Π_S^D.

Clearly we have the relation that

$$\Pi_S^D \subseteq \Pi_S \subseteq \Pi_M.$$

Theorem 1.10: Markov Property

If $\pi \in \Pi_M$ then $(S_t, A_t)_{t\in\mathbb{N}}$ is a Markov chain (time depended) on $(\mathcal{S}\times\mathcal{A})$ with the two-step transition

$$p^t_{(a,s),(a',s')} := p(s'; s, a) \cdot \pi_t(a'; s')$$

Proof.

$$
\begin{aligned}
&\mathbb{P}(S_{t+1}=s_{t+1}, A_{t+1}=a_{t+1} \mid S_t=s_t, A_t=a_t)\\
&= \frac{\mathbb{P}(S_t=s_t, A_t=a_t, S_{t+1}=s_{t+1}, A_{t+1}=a_{t+1})}{\mathbb{P}(S_t=s_t, A_t=a_t)}\\
&= \frac{\sum_{s_0,a_0}\cdots\sum_{s_{t-1},a_{t-1}} \mathbb{P}(S_0=s_0, A_0=a_0, ..., S_{t+1}=s_{t+1}, A_{t+1}=a_{t+1})}{\sum_{s_0,a_0}\cdots\sum_{s_{t-1},a_{t-1}} \mathbb{P}(S_0=s_0, A_0=a_0, ..., S_t=s_t, A_t=a_t)}\\
&= \frac{\sum_{s_0,a_0}\cdots\sum_{s_{t-1},a_{t-1}} \mu(s_0)\pi_0(a_0;s_0)\prod_{i=1}^{t+1} p(s_i;s_{i-1},a_{i-1})\pi_i(a_i,s_0,a_0,...,a_{i-1},s_i)}{\sum_{s_0,a_0}\cdots\sum_{s_{t-1},a_{t-1}} \mu(s_0)\pi_0(a_0;s_0)\prod_{i=1}^{t} p(s_i;s_{i-1},a_{i-1})\pi_i(a_i,s_0,a_0,...,a_{i-1},s_i)}\\
&= \frac{\sum_{s_0,a_0}\cdots\sum_{s_{t-1},a_{t-1}} \mu(s_0)\pi_0(a_0;s_0)\prod_{i=1}^{t+1} p(s_i;s_{i-1},a_{i-1})\varphi_i(a_i;s_i)}{\sum_{s_0,a_0}\cdots\sum_{s_{t-1},a_{t-1}} \mu(s_0)\pi_0(a_0;s_0)\prod_{i=1}^{t} p(s_i;s_{i-1},a_{i-1})\varphi_i(a_i;s_i)}\\
&= p(s_{t+1};s_t,a_t)\varphi_{t+1}(a_{t+1};s_{t+1})\\
&\overset{(*)}{=} \frac{\mathbb{P}(S_0=s_0, A_0=a_0, ..., S_{t+1}=s_{t+1}, A_{t+1}=a_{t+1})}{\mathbb{P}(S_0=s_0, A_0=a_0, ..., S_t=s_t, A_t=a_t)}\\
&= \mathbb{P}(S_{t+1}=s_{t+1}, A_{t+1}=a_{t+1} \mid S_0=s_0, A_0=a_0, ..., S_t=s_t, A_t=a_t)
\end{aligned}
$$

where we used in $(*)$ that

$$\mathbb{P}(S_0=s_0, A_0=a_0, ..., S_{t+1}=s_{t+1}, A_{t+1}=a_{t+1}) = \mu(s_0)\pi_0(a_0;s_0)\prod_{i=1}^{t} p(s_i;s_{i-1},a_{i-1})\pi_i(a_i,s_0,a_0,...,a_{i-1},s_i)p(s_{t+1};s_t$$

which is equivalent to $(*)$. □

Corollary 1.11

If we have a stationary distribution instead of a Markov policy in the theorem above, then $(S_t, A_t)_{t\in\mathbb{N}}$ is a time-homogenous Markov chain on $(\mathcal{S}\times\mathcal{A})$ with the two-step transition

$$p_{(a,s),(a',s')} := p(s';s,a)\cdot\pi(a';s')$$

Proof. Set $\varphi_t(a;s) = \varphi(a;s)$. □

Theorem 1.12: Markov Reward Property

If $\pi\in\Pi_S$ then $(S_t, A_t, R_t)_{t\in\mathbb{N}}$ satisfies the Markov reward process property

$$
\begin{aligned}
&= \mathbb{P}(S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_0, A_0) = (s_0, a_0), ..., (S_t, A_t) = (s_t, a_t))\\
&= \mathbb{P}(S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_t, A_t) = (s_t, a_t))
\end{aligned}
$$

With time-homogenuous state/reward transition probaility

$$p_{(s,a),(s',a',r')} = p(s',r;s,a)\pi(a';s').$$

Proof. Analogously only that we need to insert $t+1$ sums over the reward $\mathcal{R}$. □

Note if we choose again $\pi\in\Pi_M$ then it is a Markov reward process with that is not time-homogenuous

1.4 Stochastic Control theory

Remark 1.7

Here we will shortly explain the notation used

- $\mathbb{P}^\pi_{s,a}$ is the Markov reward process (S, A, R) started in (s, a), i.e.

$$\mathbb{P}^\pi_{s,a} := \mathbb{P}^\pi \otimes \delta_{S_0}(s) \otimes \delta_{A_0}(a),$$

  i.e.

$$\mathbb{P}^\pi_{s,a}(S_0 = s, A_0 = a_0) = \delta_s \delta_{a_0} = \mathbb{P}^\pi_{\delta_s \otimes \delta_a}(S_0 = s, A_0 = a_0)$$

- $\mathbb{P}^\pi_s$ the Markov reward process (S, A, R) started in s and the first action is chosen wrt. to $\pi_0(\cdot; s)$, i.e.

$$\mathbb{P}^\pi_s := \mathbb{P}^\pi \otimes \delta_{S_0}(s) \otimes \pi_0(\cdot; s)$$

  i.e.

$$\mathbb{P}^\pi_s(S_0 = s, A_0 = a_0) = \delta_s \pi_0(a_0; s) = \mathbb{P}^\pi_{\delta_s}(S_0 = s, A_0 = a_0)$$

- We then have the relation that

$$\mathbb{P}^\pi_s = \sum_{a \in \mathcal{A}} \mathbb{P}^\pi_{a,s} \pi_0(a, s).$$

A first potential optimization goal would be

$$\forall s \in \mathcal{S}: \quad argmax_\pi \mathbb{E}^\pi_s[\sum_{t=0}^{T} R_t] \quad \overset{\text{not clear}}{\iff} \quad \forall s \in \mathcal{S}, a \in \mathcal{A}: \quad argmax_\pi \mathbb{E}^\pi_{s,a}[\sum_{t=0}^{T} R_t].$$

There are different choices for T.

- $T = 0$: This is called contextual bandit. If $|\mathcal{S}| = 1$ then we have a bandit problem.
- $T = \min\{t \mid S_t = s'\}$ is stopping time for a fixed state $s' \in \mathcal{S}$. If we have $R \equiv 1$ then we count how many steps are needed to reach s', i.e. this is called stochastic shortest path.
- $T \in \mathbb{N}$ is fixed is called finite time MDP
- $T \sim Geo(1 - \gamma)$, $\gamma \in (0, 1)$, where T is independent of (S, A, R). (Because the geometric distribution is memoryless, it is very helpful in ???). For the geometric series, we had via induction that for all $n \in \mathbb{N}$:

$$\sum_{k=1}^{n} \gamma^k = \frac{1 - \gamma^{n+1}}{1 - \gamma} \to \frac{1}{1 - \gamma}, \quad n \to \infty???$$

  and for $T \sim Geo(p)$ we have that

$$\mathbb{P}(T = k) = (1 - \gamma)^{1-k}\gamma = \gamma e^{(1-k)\log(1-\gamma)}???$$

We can show that

$$\begin{aligned}
\mathbb{E}^\pi[\sum_{t=0}^{T} R_t] = \mathbb{E}^\pi[\sum_{t=0}^{\infty} \mathbf{1}_{t\le i, i=T\in\mathbb{N}} R_t] &= \mathbb{E}^\pi[\sum_{k=0}^{\infty} \mathbf{1}_{T=k} \sum_{t=0}^{\infty} \mathbf{1}_{t\le k} R_t] \\
&= \sum_{k=0}^{\infty}\sum_{t=0}^{\infty} \mathbb{E}^\pi[\mathbf{1}_{T=k}\mathbf{1}_{t\le k} R_t] \\
&\overset{ind.}{=} \sum_{k=0}^{\infty}\sum_{t=0}^{\infty} \mathbb{P}(T=k)\mathbf{1}_{t\le k}\mathbb{E}^\pi[R_t] \\
&= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \sum_{t=0}^{\infty} \mathbf{1}_{t\le k}\mathbb{E}^\pi[R_t] \\
&= (1-\gamma)\sum_{k=0}^{\infty}\sum_{t=0}^{\infty}\gamma^k \mathbf{1}_{t\le k}\mathbb{E}^\pi[R_t] \\
&= (1-\gamma)\sum_{t=0}^{\infty}\sum_{k=0}^{\infty}\gamma^k \mathbf{1}_{t\le k}\mathbb{E}^\pi[R_t] \\
&= (1-\gamma)\sum_{t=0}^{\infty}\sum_{k=t}^{\infty}\gamma^k \mathbb{E}^\pi[R_t] \\
&\overset{(*)}{=} (1-\gamma)\sum_{t=0}^{\infty}\frac{\gamma^t}{1-\gamma}\mathbb{E}^\pi[R_t] \\
&= \mathbb{E}^\pi[\sum_{t=0}^{\infty}\gamma^t R_t]
\end{aligned}$$

altough we used in $(*)$ that

$$\sum_{i=t}^{\infty}\gamma(1-\gamma)^{i-1} = (1-\gamma)^{t-1} \quad \Longleftrightarrow \quad \sum_{i=t}^{\infty}(1-\gamma)^{i-1} = \frac{(1-\gamma)^{t-1}}{\gamma}.$$

In practice it is not important how γ is chosen, but the convergence speed will later depend on this factor???

—

Definition 1.13: Value functions

For a policy $\pi \in \Pi$ and $\gamma \in (0,1)$ and the function $Q : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ defined as

$$\forall s,a: \quad Q^\pi(s,a) := \mathbb{E}^\pi_{s,a}[\sum_{t=0}^{\infty}\gamma^t R_t]$$

is called the state action value function (Q-function). The value function or state value function is defined as

$$\forall s\in\mathcal{S}: \quad V^\pi(s) := \mathbb{E}^\pi_s[\sum_{t=0}^{\infty}\gamma^t R_t] = \sum_{a\in\mathcal{A}_s}\pi_0(a;s)Q^\pi(s,a)$$

As it is very difficult to calculate an infinite sum, we will now show that both of these objects, are solutions to particular equations, that are feasable to calculate.
Note also that V^π is a vector in $\mathbb{R}^{|\mathcal{S}|}$ and Q^π is a matrix in $\mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, where the action space could be huge. Thus the state value function is simpler.
We will later though see that it is easier to deal with the Q-function.

Proposition 1.14

Bellman Equations:
Suppose we have a stationary policy, then Q^π and V^π satisfy the following equations. First, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_{s'}} p(s';s,a)\pi(a';s')Q^\pi(s',a').$$

Second, for all $s \in \mathcal{S}$:

$$V^\pi(s) = \sum_{a \in \mathcal{A}_s} \pi(a;s)(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a)V^\pi(s')).$$

Proof. The idea in this proof is to go one step forward, when calculating the Q/V value and then use the Markov Property, i.e. forget the step before.
For all (s,a) we have that

$$\begin{aligned}
&Q^\pi(s,a)\\
&= \mathbb{E}^\pi_{s,a}[\sum_{t=0}^{\infty} \gamma^t R_t]\\
&= \mathbb{E}^\pi_{s,a}[R_0] + \mathbb{E}^\pi_{s,a}[\sum_{t=1}^{\infty} \gamma^t R_t]\\
&= \mathbb{E}^\pi_{s,a}[R_0] + \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_{s'}} \mathbb{E}^\pi_{s,a}[\sum_{t=1}^{\infty} \gamma^t R_t \mathbf{1}_{s' \in \mathcal{S}, a' \in \mathcal{A}_{s'}}]\\
&= r(s,a) + \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_{s'}} \underbrace{\mathbb{E}^\pi_{s,a}[\sum_{t=1}^{\infty} \gamma^t R_t \mid S_1 = s', A_1 = a']}_{\overset{\text{Markov Property}}{=} \mathbb{E}^\pi_{s',a'}[\sum_{t=0}^{\infty} \gamma^{t+1} R_t]} \mathbb{P}(S_1 = s', A_1 = a')\\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_{s'}} \mathbb{E}^\pi_{s',a'}[\sum_{t=0}^{\infty} \gamma^t R_t] p(s';s,a)\pi(a';s')\\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_{s'}} Q^\pi(s',a') p(s';s,a)\pi(a';s')
\end{aligned}$$

and for the V function we have for all $s \in \mathcal{S}$

$$\begin{aligned} V^\pi(s) &\overset{Def}{=} \sum_{a \in \mathcal{A}_s} \pi(a;s) Q^\pi(s,a) \\ &= \sum_{a \in \mathcal{A}_s} \pi(a;s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_{s'}} Q^\pi(s',a') p(s';s,a) \pi(a';s') \right) \\ &= \sum_{a \in \mathcal{A}_s} \pi(a;s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) \sum_{a' \in \mathcal{A}_{s'}} Q^\pi(s',a') \pi(a';s') \right) \\ &= \sum_{a \in \mathcal{A}_s} \pi(a;s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V^\pi(s') \right) \end{aligned}$$

□

Note that the two equations above are just two linear equations. The first is a Matrix equation and the second is a vector equation.

Remark 1.8

Downsides:
The big problem with this is approach is that if were to be only interested in the Q/V value of one (s,a) pair, then we would still have to solve the Bellman equation for all state action pairs.

Corollary 1.15

For a stationary policy $\pi \in \Pi_s$ the following relation holds for all s, a

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V^\pi(s').$$

Now we have derived a set of linear equations that we have to solve. To make it a little easier, we will now go over the Banach fixed point theorem in order to find numerically a solution.

Theorem 1.16: Banach Fixed Point Theorem

Let $(U; \|\cdot\|)$ be a banach space (complete normed vector space), $T : U \to U$ a contraction, i.e.

$$\exists \lambda \in [0,1) \forall u_1, u_2 \in U : \quad \|Tu_1 - Tu_2\| \leq \lambda \|u_1 - u_2\|.$$

Then

- there exists an unique fixed point u^*, i.e. $Tu^* = u^*$ and
- for arbitrary $u_0 \in U$ the sequence $(u_n)_{n \in \mathbb{N}}$ defined by

$$u_{n+1} := Tu_n = T^{n+1} u_0$$

converges in U to u^*.

The idea of the BFT: Lets say we have a equation

$$f(x) = g(f(x)),$$

where we want to solve for f. In order to do so, we define the linear operator

$$T : \{f : \mathbb{R} \to \mathbb{R}\} \to \{f : \mathbb{R} \to \mathbb{R}\}, f \mapsto (Tf)(\cdot) =: g(f(\cdot))$$

If T satisfies the BFT then we have that for all $f_0 : f_n := T^n f_0 \to f^*$, where $(Tf^*)(x) = f^*(x)$ for all $x \in \mathbb{R}$. This means that

$$\forall x \in \mathbb{R} : f^*(x) = g(f^*(x)).$$

Therefore, we have a numerical method on how to solve systems, as long as the operator is a contraction. Now we need show that the Q and V Values satisfy the BFT. We will as of now generalize the Q and V functions as linear operators, i.e. for every policy

$$V^\pi(s) \in X := \{v : \mathcal{S} \to \mathbb{R}\} = \mathbb{R}^{|\mathcal{S}|} \quad \text{and} \quad Q^\pi(s,a) \in Y := \{q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\} = \mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}|}.$$

Then $(X, \|\cdot\|_\infty)$ and $(Y, \|\cdot\|_\infty)$ are Banach spaces. Therefore, we will now redefine the Bellman equations as operators.

Definition 1.17: Bellman expectation Operators

Given a Markov decision model and a stationary policy $\pi \in \Pi_s$ we define $T^\pi : X \to X$ where

$$(T^\pi v)(s) := \sum_{a \in \mathcal{A}_s} \pi(a; s) \cdot (r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s, a) v(s'))$$

and $T^\pi : Y \to Y$ where

$$(T^\pi q)(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}_s} p(s'; s, a) \pi(a'; s') q(s,a)$$

are both called Bellman expectation operators.

We denoted for both cases the operator as T. It will always be visible form the context, what we mean. In the Proposition, where we showed that the Q and V values are solutions of the Bellman equations, we thus already showed that Q^π and V^π are the fixed points of the Bellman expectation operators. What remains to be shown, is that this operator is a contraction.

Theorem 1.18: Bellman expectation operator is contraction

Both Bellman expectation operators are contractions with constant γ which is the discount factor. Their unique fixed points are

$$T^\pi V^\pi = V^\pi \quad \text{and} \quad T^\pi Q^\pi = Q^\pi.$$

Proof. For every $s \in \mathcal{S}$ it holds that

$$\begin{aligned}\|T^\pi v_1 - T^\pi v_2\|_\infty &= \max_{s\in\mathcal{S}} \left| \sum_{a\in\mathcal{A}_s} \pi(a;s)\gamma \sum_{s'\in\mathcal{S}} p(s';s,a)(v_1(s') - v_2(s')) \right| \\ &\leq \gamma\|v_1 - v_2\|_\infty \max_{s\in\mathcal{S}} \left| \sum_{a\in\mathcal{A}_s} \pi(a;s) \sum_{s'\in\mathcal{S}} p(s';s,a) \right| \\ &= \gamma\|v_1 - v_2\|_\infty\end{aligned}$$

and for every $(s,a) \in \mathcal{S}\times\mathcal{A}$ that

$$\begin{aligned}\|T^\pi q_1 - T^\pi q_2\|_\infty &= \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left| \gamma \sum_{a'\in\mathcal{A}_{s'}} \sum_{s'\in\mathcal{S}} p(s';s,a)\pi(a';s')(q_1(s',a') - q_2(s',a')) \right| \\ &\leq \gamma\|q_1 - q_2\|_\infty \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left| \sum_{a'\in\mathcal{A}_{s'}} \sum_{s'\in\mathcal{S}} p(s';s,a)\pi(a';s') \right| \\ &= \gamma\|q_1 - q_2\|_\infty.\end{aligned}$$

□

We have now derived a method in deriving numerically the value of a policy. In order to find the optimal policy for every state action pair (or just state) we will now define an optimal policy and then find their respective operators and show that they are contractions and their unique fixed points will be the optimal policy.

Definition 1.19: Optimal Policy and Optimal Value functions

For a given MDP we define

- Optimal state value function $V^* : \mathcal{S} \to \mathbb{R}$, where
$$\forall s \in \mathcal{S}: \quad V^*(s) := \sup_{\pi\in\Pi} V^\pi(s)$$
- Optimal state action function $Q^* : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ where
$$\forall s \in \mathcal{S}, a \in \mathcal{A}: \quad Q^*(s,a) := \sup_{\pi\in\Pi} Q^\pi(s,a).$$
- A policy $\pi^* \in \Pi$ is called optimal, if and only if it satisfies
$$\forall s \in \mathcal{S}: \quad V^{\pi^*}(s) = V^*(s).$$

Note that the optimal state (action) function was defined pointwise, i.e. V^* or Q^* could be choose different π for two different states $s \neq s'$. Thus it is absolutely not clear, weather optimal policies exist, as they are defined for all starting states!
For ease we will assume that we are only in finite MDPs. The next results do not hold for infinite MDPs.
Note that we could have defined the optimal policy in terms of Q^π, but in literature it is most common to define it over the state value function.

Remark 1.9

We will later see, that for finite MDPs, there is a stationary deterministic policy that solves these Bellman optimality equations.

We will now define the linear systems where the optimal Q and V functions are the solutions and the respective optimality operators.

Definition 1.20: Bellman Optimality Operators

Given a MDP we define

- The non linear system of equations

$$\forall s \in \mathcal{S}: \quad v(s) = \max_{a \in \mathcal{A}_s} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s, a) v(s') \right)$$

  is called Bellman optimality equations and the operator $T^* : X \to X???$ defined by

$$(T^* v)(s) := \max_{a \in \mathcal{A}_s} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s, a) v(s') \right)$$

- The non linear system of equations

$$\forall s \in \mathcal{S}, a \in \mathcal{A}: \quad q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s, a) \max_{a' \in \mathcal{A}_{s'}} q(s', a')$$

  is called Bellman optimality state action equation and the state action Bellman optimality operator $T^* : Y \to Y???$ is defined by

$$\forall s \in \mathcal{S}, a \in \mathcal{A}: \quad (T^* q)(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s, a) \max_{a' \in \mathcal{A}_{s'}} q(s', a')$$

WO IST PI HIN??? Again, both Bellman optimality operators will be distinguishable by the number of arguments.
Again, we will now apply the Banach fixed point theorem.

Lemma 1.21: Monotonicity

Both Bellman operators T^π and T^* are monotone, i.e.

$$u_1 \leq u_2 \Rightarrow T^* u_1 \leq T^* u_2 \text{ and } T^\pi u_1 \leq T^\pi u_2.$$

Theorem 1.22

The Bellman optimality operators are contractions and thus have unique fixed points.

Proof. — □

Theorem 1.23

The optimal value functions are the unique fixed points of the Bellman optimality operators, i.e.

$$T^*V^* = V^* \text{ and } T^*Q^* = Q^*.$$

Further, it holds that

$$V^*(s) = \max_{a \in \mathcal{A}_s} Q^*(s,a).$$

Proof. The property of contraction was already shown. We will now show that Q^* and V^* are indeed satisfy the Bellman optimality equations.
Stationary Policies:
Let π be a stationary policy. We will first show the Bellman optimality equation for Q^*. In order to differentiate easier, we define

$$\bar{Q}^*(s,a) := \sup_{\pi \in \Pi_s} Q^\pi(s,a) \quad \text{and} \quad Q^*(s,a) := \sup_{\pi \in \Pi} Q^\pi(s,a)$$

and

$$\bar{V}^*(s) := \sup_{\pi \in \Pi_s} V^\pi(s) \quad \text{and} \quad V^*(s) := \sup_{\pi \in \Pi} V^\pi(s)$$

We will begin by prooving the last property as this is very essential property in the existence of the optimal policy???
We have

$$\begin{aligned}
\bar{V}^*(s) &:= \sup_{\pi \in \Pi_S} V^\pi(s) \\
&\overset{(*)}{=} \sup_{\pi \in \Pi_S} \max_{a \in \mathcal{A}_s} Q^\pi(s,a) \\
&\overset{finite}{=} \max_{a \in \mathcal{A}_s} \sup_{\pi \in \Pi_S} Q^\pi(s,a) \\
&= \max_{a \in \mathcal{A}_s} Q^*(s,a)
\end{aligned}$$

$(*)$ has to be shown:
"$\leq$"

$$\begin{aligned}
\sup_{\pi \in \Pi_S} V^\pi(s) &= \sup_{\pi \in \Pi_S} \sum_{a' \in \mathcal{A}_s} \pi_0(a';s) Q^\pi(s,a') \\
&\leq \sup_{\pi \in \Pi_S} \max_{a \in \mathcal{A}_s} Q^\pi(s,a) \sum_{a' \in \mathcal{A}_s} \pi_0(a';s) \\
&= \sup_{\pi \in \Pi_S} \max_{a \in \mathcal{A}_s} Q^\pi(s,a)
\end{aligned}$$

"$\geq$"

$$\begin{aligned}
\sup_{\pi\in\Pi_S} V^\pi(s) &= \sup_{\pi\in\Pi_S} \sum_{a\in\mathcal{A}_s} \pi(a;s) Q^\pi(s,a) \\
&\geq \sup_{\pi\in\Pi_S^D} \sum_{a\in\mathcal{A}_s} \pi(a;s) Q^\pi(s,a) \\
&\overset{SimiliarToBelow}{=} \sum_{a\in\mathcal{A}_s} \sup_{\pi\in\Pi_S^D} \pi(a;s) Q^\pi(s,a) \\
&\geq \sum_{a\in\mathcal{A}_s} \pi(a;s) \sup_{\pi\in\Pi_S^D} Q^\pi(s,a) \\
&\geq \max_{a\in\mathcal{A}_s} \underbrace{\pi(a;s)}_{DiracMeasure} \sup_{\pi\in\Pi_S^D} Q^\pi(s,a) \\
&\geq \sup_{\pi\in\Pi} \max_{a\in\mathcal{A}_s} Q^\pi(s,a) \\
&= \max_{a\in\mathcal{A}_s} \sup_{\pi\in\Pi} Q^\pi(s,a)
\end{aligned}$$

Next, we will derive show the Bellman equations. We first use the Bellman equation inside of the supremum. For all (s,a)

$$\begin{aligned}
\bar{Q}^*(s,a) &= \sup_{\pi\in\Pi_s} Q^\pi(s,a) \\
&= \sup_{\pi\in\Pi_s} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi(a';s') Q^\pi(s',a') \right) \\
&= r(s,a) + \gamma \sup_{\pi\in\Pi_s} \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi(a';s') Q^\pi(s',a') \\
&\overset{(*)}{=} r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} \underbrace{\sup_{\pi\in\Pi_s} Q^\pi(s',a')}_{=\bar{Q}^*(s',a')}
\end{aligned}$$

Although $(*)$ is not clear at all. We will now show it.
"$\leq$":

$$\begin{aligned}
\sup_{\pi\in\Pi_s} \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi(a';s') Q^\pi(s',a') &= \sup_{\pi\in\Pi_s} \sum_{s'\in\mathcal{S}} p(s';s,a) \sum_{a'\in\mathcal{A}_{s'}} \pi(a';s') Q^\pi(s',a') \\
&\leq \sup_{\pi\in\Pi_s} \sum_{s'\in\mathcal{S}} p(s';s,a) \sum_{a'\in\mathcal{A}_{s'}} \pi(a';s') \sup_{\pi\in\Pi_s} Q^\pi(s',a') \\
&\leq \sup_{\pi\in\Pi_s} \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a\in\mathcal{A}_{s'}} \sup_{\pi\in\Pi_S} Q^\pi(s',a) \underbrace{\sum_{a'\in\mathcal{A}_{s'}} \pi(a';s')}_{=1} \\
&= \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} \sup_{\pi\in\Pi_S} Q^\pi(s',a')
\end{aligned}$$

"$\geq$":

This is a little harder. We will first find a lower bound that is even lower:

$$\sup_{\pi\in\Pi_S}\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi(a';s')Q^\pi(s,a) \geq \sup_{\pi\in\Pi_S^D}\sum_{s'\in\mathcal{S}} p(s';s,a)\max_{a'\in\mathcal{A}_{s'}} Q^\pi(s',a')$$

i.e. we only consider stationary deterministic policies. The benefit of doing this, is that not only do we have a finite sum, but also a finite supremum and thus it gets easier. We now show that in the case of deterministic stationary policies, we can interchange the sum. For ease of notation we define

$$\sup_{\pi\in\Pi_S^D}\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi(a';s')Q^\pi(s,a) =: \sup_{\pi\in\Pi_S^D}\sum_{s'\in\mathcal{S}} h(s,\pi(s))$$

And now we want to show

$$\sup_{\pi\in\Pi_S^D}\sum_{s\in\mathcal{S}} h(s,\pi(s)) = \sum_{s\in\mathcal{S}}\sup_{\pi\in\Pi_S^D} h(s,\pi(s)).$$

"$\leq$" Is trivial

"$\geq$" We will show this via contradiction. I.e. suppose

$$\sup_{\pi\in\Pi_S^D}\sum_{s\in\mathcal{S}} h(s,\pi(s)) < \sum_{s\in\mathcal{S}}\sup_{\pi\in\Pi_S^D} h(s,\pi(s)).$$

Then there exists a $\delta > 0$ such that

$$\sup_{\pi\in\Pi_S^D}\sum_{s\in\mathcal{S}} h(s,\pi(s)) < \sum_{s\in\mathcal{S}}\sup_{\pi\in\Pi_S^D} h(s,\pi(s)) - \delta.$$

Next, choose an arbitrary policy $\tilde{\pi}$ such that for all $\epsilon > 0$ it holds that

$$\forall s\in\mathcal{S}: \quad h(s,\tilde{\pi}(s)) > \sup_{\pi\in\Pi_S^D} h(s,\pi(s)) - \frac{\epsilon}{|\mathcal{S}|}.$$

Then we can show that

$$\begin{aligned}\sum_{s\in\mathcal{S}}\sup_{\pi\in\Pi_S^D} h(s,\pi(s)) &< \sum_{s\in\mathcal{S}} h(s,\tilde{\pi}(s)) + \epsilon \\ &< \sup_{\pi\in\Pi_S^D}\sum_{s\in\mathcal{S}} h(s,\pi(s)) + \epsilon \\ &< \sum_{s\in\mathcal{S}}\sup_{\pi\in\Pi_S^D} h(s,\pi(s)) - \delta + \epsilon\end{aligned}$$

If we now choose $\delta > \epsilon$ we get a contradiction.

Now we can show "$\geq$":

$$\begin{aligned}\sup_{\pi\in\Pi_S}\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi(a';s')Q^\pi(s',a') &= \sum_{s'\in\mathcal{S}} p(s';s,a)\sup_{\pi\in\Pi_S}\sum_{a'\in\mathcal{A}_{s'}}\pi(a';s')Q^\pi(s',a') \\ &\geq \sum_{s'\in\mathcal{S}} p(s';s,a)\sup_{\pi\in\Pi_S}\max_{a'\in\mathcal{A}_{s'}}\pi(a';s')Q^\pi(s',a') \\ &\geq \sum_{s'\in\mathcal{S}} p(s';s,a)\sup_{\pi\in\Pi_S^D}\max_{a'\in\mathcal{A}_{s'}}\pi(a';s')Q^\pi(s',a') \\ &= \sum_{s'\in\mathcal{S}} p(s';s,a)\sup_{\pi\in\Pi_S^D}\max_{a'\in\mathcal{A}_{s'}} Q^\pi(s',a') \\ &= \sum_{s'\in\mathcal{S}} p(s';s,a)\max_{a'\in\mathcal{A}_{s'}}\sup_{\pi\in\Pi_S^D} Q^\pi(s',a').\end{aligned}$$

Next, we will look at the Bellman optimality equation for V^*:
Here we will use the result that we showed in the beginning of the proof. For all $s \in \mathcal{S}$ we have

$$\begin{aligned}\bar{V}^*(s) &= \max_{a\in\mathcal{A}_s} \bar{Q}^*(s,a) \\ &= \max_{a\in\mathcal{A}_s} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} \bar{Q}^*(s',a') \right) \\ &= \max_{a\in\mathcal{A}_s} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \bar{V}^*(s) \right)\end{aligned}$$

Non Stationary Policies: Skipped. □

Up to now we have not said nothing about the optimal policy! We have only showed that the optimal state (value) functions are unique fixed points of the Bellman optimality operator and that is suffices to only consider stationary policies for the optimal state (value) function. Even better is that the Bellman optimality equations hold for stationary deterministic policies (showed in Dynamic Programming Programming). This makes the set Π a lot smaller and more possible to handle.

Definition 1.24: Greedy Policy

Given a function $q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ then the following stationary deterministic policy

$$\pi_q(s,a) := \begin{cases} 1 & , a \in argmax_a q(s,a) \\ 0 & , else \end{cases}$$

is called greedy policy. We can also obtain the greedy policy from the value function $v : \mathcal{S} \to \mathbb{R}$ and then using the $Q - V$ transfer

$$q_v(s,a) := r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) v(s')$$

We then write π_v.

Lemma 1.25: Bellman optimality Operator and Bellman operator for Greedy Policy are equivalent

Suppose π_q is the greedy policy obtained from q, then $T^*q = T^{\pi_q}q$.
If q is obtained from v then $T^*v = T^{\pi_v}v$.

Proof. Both follow very quickly from the definition of the greedy policy. For all (s,a) we have

$$\begin{aligned}(T^{\pi_q}q)(s,a) &= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi_q(s';a')q(s',a') \\ &= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \cdot 1 \cdot \max_{a'\in\mathcal{A}_{s'}} q(s',a') \\ &= T^*q(s,a)\end{aligned}$$

and for all $s \in \mathcal{S}$

$$\begin{aligned}(T^{\pi_v}v)(s) &= \sum_{a\in\mathcal{A}_s} \pi_v(a;s)(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a)v(s')) \\ &= \max_{a\in\mathcal{A}_s} \left(1 \cdot (r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a)v(s'))\right) \\ &= T^*v(s).\end{aligned}$$

□

Theorem 1.26: Dynamic Programming Algorithm

An optimal policy $\pi^* \in \Pi$ always exists and can be chosen as stationary deterministic. The optimal policy $\pi^* \in \Pi_S^D$ is the Greedy policy wrt. to Q^* or V^*, i.e. solving the Bellman optimality equations and then playing Greedy.

Proof. We have already proven that the optimal state (action) value function Q^* and V^* exist and is the unique solution of T^π. We now have to show that the greedy policy obtained from this optimal value function is the optimal policy. We first show $Q^* = Q^{\pi_{Q^*}}$

$$Q^* = T^*Q^* = T^{\pi_{Q^*}}Q^*$$

Because both Bellman Operators have unique solutions and $T^{\pi_{Q^*}}Q^{\pi_{Q^*}} = Q^{\pi_{Q^*}}$, we have that $Q^{\pi_{Q^*}} = Q^*$.
To show: $V^* = V^{\pi_{V^*}}$
Using the property of the value function that we have proven in a different Theorem:

$$V^*(s) = \max_{a\in\mathcal{A}_s} Q^*(s,a)$$

and the definition $V^\pi(s) := \sum_{a\in\mathcal{A}_s} \pi(a;s)Q^\pi(s,a)$. This yields

$$V^*(s) = \max_{a\in\mathcal{A}_s} Q^*(s,a) = \sum_{a\in\mathcal{A}_s} \pi_{Q^*}(a;s)Q^{\pi_{Q^*}}(s,a) = V^{\pi_{Q^*}}(s).$$

Thus the state (action) value function of the greedy policy wrt. the optimal state (action) value function is equal to the optimal state (action) value function. This holds for every initial state $s \in \mathcal{S}$ and thus we have shown the definition of the optimal policy. □

Remark 1.10

In Words:
The optimal policy is a stationary determistic Policy that can be obtained and is Greedy wrt. to the optimal Value functions (either Q or V).
In the beginning we had the question wether there exists one policy that is optimal for all starting states, i.e.

$$\sup_{\pi\in\Pi}(V^\pi(s))_{s\in\mathcal{S}} \stackrel{?}{=} (\sup_{\pi_s\in\Pi} V^{\pi_s}(s))_{s\in\mathcal{S}}.$$

Answer: Yes.

The focus from now on will be to find stationary optimal policies.

Definition 1.27: Learning Strategy

A sequence of policies $(\pi^n)_{n\in\mathbb{N}}$ for a MDP is called learning strategy, if π^{n+1} only depends on the first n rounds of learning.

We keep this definition vage. Our aim is to find learning strategies that converge as quickly as possible $\pi^n \to \pi^*$ or

$$\|V^{\pi^n} - V^*\| \to 0, n \to \infty.$$

We do not care about regret.
There are typically two approaches

- Value function based learning. Here the idea is to learn V^* and then taking the argmax, i.e. Greedy. (Remember the optimal policy is greedy, thus if we know V^* we get π^* by playing Greedy.
- Policy based learning: Tries to approximate the optimal policy π^* directly.

Remark 1.11

The following two algorithms are called Dynamic programming algorithms. Their idea is to reduce one big problem into smaller subproblems and then using these solved subproblems to solve the big problem. In infinte time MDPs this does not really make sense, because the idea of dynamic programming is to go one step and restart the problem, but in infinite time this does not reduce complexity. Later in finite time MDPs, this will have an effect and we will revisit this idea.

2 Basic tabular Value Iteration

The idea here is to use the Banach Fixed point theorem, that for any V_0 we have that

$$(T^*)^n V_0 \to T^* V^* = V^*, \quad n \to \infty.$$

Finally we only need to play greedy wrt. to V^*.

Algorithm 6: Basic Value Iteration

Data : MDP and Accuracy $\epsilon > 0$
Initialize $V \equiv 0$ and $V_{new} \equiv 0$
$\Delta := 1$
while $\Delta > \epsilon$ **do**
  set $V := V_{new}$ **for** $s \in \mathcal{S}$ **do**

$$V_{new}(s) := \underbrace{\max_{a\in\mathcal{A}_s} \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) V(s') \right)}_{(T^*V)(s)}$$

  $\Delta := \max_{s\in\mathcal{S}}(|V_{new}(s) - V(s)|)$
Result: $V := V_{new} \approx V^*$

Theorem 2.1

The Basic Value iteration algorithm terminates and the output satisfies

$$\|V - V^*\|_\infty \leq \frac{\gamma\epsilon}{1-\gamma}.$$

Proof. For $v_{n+1} := T^* v_n$ it holds that

$$\|v_n - v_{n+1}\|_\infty \leq \|v_n - v^*\|_\infty + \|v^* - v_{n+1}\|_\infty \overset{BFT}{\to} 0, \quad n \to \infty.$$

We define $V := v_n$ where $\|v_n - v_{n-1}\|_\infty < \epsilon$, i.e. the terminating condition of the algorithm. Then

$$\begin{aligned}
\|v_n - v^*\|_\infty &= \lim_{m\to\infty} \|v_n - v_{n+m}\|_\infty \\
&\overset{TelescopingSum}{=} \lim_{m\to\infty} \|\sum_{k=0}^{m-1} v_{n+k} - v_{n+k+1}\|_\infty \\
&\leq \lim_{m\to\infty} \sum_{k=0}^{m-1} \|v_{n+k} - v_{n+k+1}\|_\infty \\
&= \lim_{m\to\infty} \sum_{k=0}^{m-1} \|(T^*)^k v_n - (T^*)^k v_{n+1}\|_\infty \\
&\leq \lim_{m\to\infty} \sum_{k=0}^{m-1} \gamma^k \|v_n - v_{n+1}\|_\infty \\
&= \|v_n - v_{n+1}\|_\infty \sum_{k=0}^{\infty} \gamma^k \\
&= \|v_n - v_{n+1}\|_\infty \frac{1}{1-\gamma} \\
&\leq \|v_n - v_{n-1}\|_\infty \frac{\gamma}{1-\gamma} \leq \epsilon \frac{\gamma}{1-\gamma}.
\end{aligned}$$

□

Because the output of Algorithms do not lead to optimal Value functions, but just in approximation, we need to specify, what polices are that are derived from approximate optimal value functions

Definition 2.2: ϵ-Optimal

For $\epsilon > 0$ we call a policy $\pi \in \Pi$ $\epsilon - Optimal$ if

$$\forall s \in \mathcal{S}: \quad V^*(s) \leq V^\pi(s) + \epsilon \,(\iff \quad |V^*(s) - V^\pi(s)| \leq \epsilon)\,.$$

Remember that we can obtain the Q function trough the state Value function V.

Theorem 2.3

Suppose that V is the output from the Basic Value iteration algorithm, Q is obtained from V and π_Q is the Greedy policy of Q, then π_Q is $\frac{2\epsilon\gamma}{1-\gamma}$-optimal.

Proof. Assume that V is obtained from the algorithma and we then optain V from the $Q-V$ transfer. Then define $\pi_Q := greedy(Q)$. We get the following relation

$$\begin{aligned}
T^*V(s) &= \max_{a\in\mathcal{A}_s}\left(r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)V(s')\right)\\
&= \max_{a\in\mathcal{A}_s}Q(s,a)\\
&= \sum_{a\in\mathcal{A}_s}\pi_Q(a;s)Q(s,a)\\
&= \sum_{a\in\mathcal{A}_s}\pi_Q(a;s)\left(r(s,a)+\gamma\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}_{s'}}p(s';s,a)\pi(a;s)Q(s',a')\right)\\
&= \sum_{a\in\mathcal{A}_s}\pi_Q(a;s)\left(r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)V(s')\right)\\
&= T^{\pi_Q}V
\end{aligned}$$

Then for $V := v_n := T^*v_{n-1}$ it holds that

$$\begin{aligned}
\|V^{\pi_Q}-V\|_\infty &\le \|V^{\pi_Q}-T^*V\|_\infty + \|T^*V-V\|_\infty\\
&= \|T^{\pi_Q}V^{\pi_Q}-T^*V\|_\infty + \|T^*V-T^*v_{n-1}\|_\infty\\
&= \|T^*V^{\pi_Q}-T^*V\|_\infty + \|T^*V-T^*v_{n-1}\|_\infty\\
&\le \gamma\|V^{\pi_Q}-V\|_\infty + \gamma\|V-v_{n-1}\|_\infty
\end{aligned}$$

Which is equivalent to

$$(1-\gamma)\|V^{\pi_Q}-V\|_\infty \le \gamma\|V-v_{n-1}\|_\infty \quad\Longleftrightarrow\quad \|V^{\pi_Q}-V\|_\infty \le \frac{\gamma}{1-\gamma}\|V-v_{n-1}\|_\infty$$

Now we can use from the previous proof that $\|V-V^*\|_\infty \le \frac{\gamma}{1-\gamma}\|v_n-v_{n-1}\|_\infty \le \frac{\epsilon\gamma}{1-\gamma}$, where we used in the last step, that the algorithm terminates under the condition that the distance to the previous step is smaller than ϵ. Using this we get that

$$\begin{aligned}
\|V^{\pi_Q}-V^*\|_\infty &\le \|V^{\pi_Q}-V\|_\infty + \|V-V^*\|_\infty\\
&\le \frac{\gamma}{1-\gamma}\|V-v_{n-1}\|_\infty + \frac{\gamma}{1-\gamma}\|v_n-v_{n-1}\|_\infty\\
&= \frac{2\gamma}{1-\gamma}\|v_n-v_{n-1}\|_\infty \le \frac{2\epsilon\gamma}{1-\gamma}
\end{aligned}$$

We use here the Lemma that

$$T^\pi V = T^*V \quad\Longleftrightarrow\quad \pi = greedy(V)$$

□

In order to compare different algorithms we define the speed of convergence.

Definition 2.4: Convergence of Order α

Suppose we have a normed space $(V, \|\cdot\|)$, then for a sequence $(y_n)_{n\in\mathbb{N}} \subset V$ with limit $v^* \in V$ we call it convergence of order $\alpha > 0$, if

$$\exists K < 1 \forall n \in \mathbb{N}: \quad \|y_{n+1} - y^*\| \leq K\|y_n - y^*\|^\alpha.$$

In the case that $\alpha = 1$, we call it linear convergence.

Linear convergence should be called exponential convergence, because of

$$\|y_{n+1} - y^*\| \leq K\|y_n - y^*\| \leq K^2\|y_{n-1} - y^*\| \leq ... \leq K^n\|y_0 - y^*\| \in \mathcal{O}(K^n).$$

Because T^* is a contraction, we have at least linear convergence of the Basic Value iteration Algorithm. The next theorem shows that this cannot be further improved.

Theorem 2.5

For all initializations the convergence order of the Basic Value Iteration algorithm is linear with constant $K = \gamma$. There is one initialization, where the convergence is exactly linear, i.e. there is equality with $K = \gamma$.

Proof. For $V_{n+1} := T^*v_n$ it holds that

$$\|v_{n+1} - V^*\|_\infty = \|T^*v_n - T^*V^*\|_\infty \leq \gamma\|v_n - V^*\|_\infty.$$

We actually do not get better than linear convergence, due to the following. If we initailize by $v_0 := V^* + k\mathbf{1}$ we then get that

$$\begin{aligned}\|v_1 - V^*\|_\infty &= \|T^*v_0 - V^*\|_\infty \\ &= \|T^*(V^* + k\mathbf{1} - V^*\|_\infty \overset{(*)}{=} \|V^* + T^*k\mathbf{1} - V^*\|_\infty \\ &= \|\gamma k\mathbf{1}\|_\infty = \gamma\|k\mathbf{1}\|_\infty \\ &= \gamma\|V^* + k\mathbf{1} - V^*\|_\infty = \gamma\|v_0 - V^*\|_\infty\end{aligned}$$

Via induction it follows that

$$\|v_n - V^*\|_\infty = \gamma\|v_{n-1} - V^*\|_\infty$$

for all $n \in \mathbb{N}$ □

Remark 2.1

If the state space get bigger, then the vector V gets larger and the norm $\|V_0 - V^*\|_\infty$ increases, as the supremum over a larger set is larger.

Remark 2.2

Note that we could directly do the Banach fixed point theorem for Q instead of doing the $Q-V$ transfer twice. This will be later done in Q-learning and SARSA. Further, note that we assumed knowledge of the transition functions. This is not always the case.

3 Basic Policy Iteration (Actor Critic) algorithm

This will be our first class of real Reinforcement Learning algorithms, as this is not part of stochastic Control. The idea is as follows of Policy Iteration (Actor Critic): Initialize some Policy π_0 and then iterate between the following two steps

- Policy Evaluation: Evaluate the current Policy π, by calculating Q^π or V^π
- Policy Improvement: Improve the current Policy. We will later see that we get an strict improvement of the Policy if the new policy $\pi' = greedy(Q^\pi)$.

In contrast to Value Iteration, the Policy Iteration method is much more clever, as it uses more understanding of the Optimal Control Problem (because it knows that the optimal policy is greedy???).
Value Iteration is the class of algorithms known as Value based methods and Policy Iteration is in the class of Policy based methods, as it works closely with the policy in the Policy improvement step.

Remark 3.1

This method is called Actor Critc, as we always alternate between the actor and the Critic:

- Critic: The Critic evaluates the current Policy π
- Actor: The Actor improves the Policy.

3.1 Policy Evaluation

We will now discuss how we can calculate the Value function V^π of a specific policy π. There are three ways of doing this.

- Approximate the expectation within Q^π by Monte Carlo Methods
- Solve the Bellman expectation equation using linear Algebra methods
- Use Banch Fixed Point theorem for the Operator T^π.

We will first discuss linear Algebra methods and then the Banach fixed point theorem.

Remark 3.2

Recall that we can write the Bellman expectation operator for $s \in \mathcal{S}$ as

$$(T^\pi v)(s) = \sum_{a \in \mathcal{A}_s} \pi(a; s) \left(r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s, a) v(s') \right)$$

We can write this as Matrix/vector Product as follows

$$V^\pi = r^\pi + \gamma P^\pi V^\pi$$

where

$$P^\pi := \left(\sum_{a \in \mathcal{A}_f} \pi(a; s) p(s'; s, a) \right)_{(s,s') \in \mathcal{S}^2}$$

and

$$r^\pi := \left(\sum_{a \in \mathcal{A}_s} \pi(a;s) r(s,a) \right)_{s \in \mathcal{S}}$$

where $P^\pi \in \mathbb{R}^{|\mathcal{S}|^2}$ and $r^\pi, V^\pi \in \mathbb{R}^{|\mathcal{S}|}$.

Remark 3.3

Check that the operator is $T^\pi = r^\pi + (\gamma P^\pi)$.:

$$\begin{aligned}
V &= T^\pi V \\
&= \left(\sum_{a \in \mathcal{A}_s} \pi(a;s)(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V(s') \right)_{s \in \mathcal{S}} \\
&= \left(\sum_{a \in \mathcal{A}_s} \pi(a;s) r(s,a) \right)_{s \in \mathcal{S}} + \left(\sum_{a \in \mathcal{A}_s} \pi(a;s) \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V(s') \right)_{s \in \mathcal{S}} \\
&= r_\pi + \gamma \left(\sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_s} \pi(a;s) p(s';s,a) V(s') \right)_{s \in \mathcal{S}} \\
&= r_\pi + \gamma \left(\sum_{s' \in \mathcal{S}} P^\pi(s,s') V(s') \right)_{s \in \mathcal{S}} \\
&= r_\pi + \gamma \left(\langle P^\pi(s,\cdot), V \rangle \right)_{s \in \mathcal{S}} \\
&= r_\pi + \gamma P^\pi V
\end{aligned}$$

Theorem 3.1

The linear (affine) equation $V^\pi = r^\pi + P^\pi V^\pi$ has a unique solution given by

$$V^\pi = (I - \gamma P^\pi)^{-1} r^\pi.$$

Proof. First, we need to know weather we can invert any parts of the matricies appearing in this equation. We can do this, because with the Banach fixed point theorem, there exists a unique solution and thus they are with LA1 inveritble. Thus we can calculate

$$\begin{aligned}
V^\pi = r^\pi + \gamma P^\pi V^\pi &\iff V^\pi - \gamma P^\pi V^\pi = r^\pi \\
&\iff V^\pi = (I - \gamma P^\pi)^{-1} r^\pi.
\end{aligned}$$

□

A important point to consider, is that calculating the state value function in this form is for programming more efficient, as loops are slower than doing matrix calculations.

Remark 3.4

Looking at the perforamnce of this method yields that it is very slow, as inverting a matrix is of order $\mathcal{O}(n^3)$, i.e. very slow, where $n = |\mathcal{S}|$.
In comparison, for the Banach fixed point theorem we do matrix multiplikations, which is of order $\mathcal{O}(n^2)$, which is faster.

Algorithm 7: Iterative Policy Evaluation (Naive)

Data : Policy $\pi \in \Pi,\ \epsilon > 0$
Result: Approximation V^n

Initialize $V \equiv 0$, $V_{\text{new}} \equiv 0$
$\Delta = 1$
while $\Delta \geq \epsilon$ **do**
  $V = V_{\text{new}}$
  for $s \in \mathcal{S}$ **do**

$$V_{\text{new}}(s) \;=\; \sum_{a \in \mathcal{A}_s} \pi(a \,|\, s) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s',\, r \,|\, s,\, a) \left[r \;+\; \gamma\, V(s') \right]$$

$$\Delta \;=\; \max_{s \in \mathcal{S}} \left| V_{\text{new}}(s) \;-\; V(s) \right|$$

Result: $V_{\text{new}} \;\approx\; V^{\pi}$

Theorem 3.2

The algorithm above terminates and its output satisifies

$$\|V - V^{\pi}\| \leq \frac{\gamma \epsilon}{1 - \gamma}.$$

Proof. Is the exact same proof as with T^*, due to the fact that T^{π} is also a contraction with γ and nothing more was used. Thus we also get the exact same bound. □

The Performance of Naive Policy iteration algorithm can be improved, if we do not update the value function for every state, but just for the once we deem as "relevant".

Remark 3.5

Asynchroneous Updates are used for the Gauss-Seidel Algorithm. Here, the matrix A only updates one entry of the vector v_n , i.e. for $s \in \mathcal{S}$

$$v_{n+1}(s) := (Av_n)(s)$$

and then for another $s' \in \mathcal{S}$ we have

$$v_{n+2} = (Av_{n+1})(s').$$

But note that if we always choose different $s \in \mathcal{S}$ and do the matrix multiplikation $dim(A)$-often (lets assume Full rank), then it is not the same as doing standard matrix multiplication once. But, if we

define the pointwise operator Bellman expectation operator

$$T_s^\pi V(s') = \begin{cases} T^\pi V(s) & , s = s' \\ V(s) & , s \neq s' \end{cases}$$

then the limit is the same as in the standard Bellman expectation operator. Then it holds that

a) $\bar{T}^\pi$ is different from T^π

b) V^π is a fixed point of $\bar{T}^\pi$

c) $\bar{T}^\pi$ is a contraction.

Proof. a) T^π updates v for all $s \in \mathcal{S}$, while $T_{s_1}^\pi$ only updates $v(s_1)$ and leaves the $v(s_i)$ unchanged for $i \neq 1$. If we now apply $T_{s_2}^\pi$ then the Bellman operator changes to

$$(T_{s_2}^\pi \circ T_{s_1}^\pi)(v)(s_2) = \sum_{a \in \mathcal{A}_{s_2}} \pi(a; s_2) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S} \setminus \{s_1\}} p(s'; s_2, a) v(s') + \gamma p(s_1; s_2, a) T_{s_1}^\pi v(s_1) \right).$$

But if we update the value functio V^π it is the same

$$\begin{aligned}(T_{s_2}^\pi \circ T_{s_1}^\pi)(V^\pi)(s_2) &= \sum_{a \in \mathcal{A}_{s_2}} \pi(a; s_2) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S} \setminus \{s_1\}} p(s'; s_2, a) V^\pi(s') + \gamma p(s_1; s_2, a) \underbrace{T_{s_1}^\pi V^\pi(s_1)}_{= T^\pi V^\pi(s_1) = V^\pi(s_1)} \right) \\ &= \sum_{a \in \mathcal{A}_{s_2}} \pi(a; s_2) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'; s_2, a) V^\pi(s') \right).\end{aligned}$$

b) We show pointwise via induction that V^π is the fixed point of T^π.
IA:

$$\begin{aligned}\bar{T}^\pi(V^\pi)(s_1) &= (T_{s_k}^\pi \circ \cdots \circ T_{s_1}^\pi)(V^\pi)(s_1) \\ &= T_{s_1}^\pi(V^\pi)(s_1) = T^\pi(V^\pi)(s_1) = V^\pi(s_1)\end{aligned}$$

Due to the fact that for all $s \neq s_1$ now updates occurs, i.e. $T_{s_1}^\pi V^\pi(s) = V^\pi(s)$ it follows that

$$(\bar{T}^\pi)(V^\pi) = (T_{s_k}^\pi \circ \cdots \circ T_{s_1}^\pi)(V^\pi) = (T_{s_k}^\pi \circ \cdots \circ T_{s_2}^\pi)(V^\pi).$$

IS:

$$\begin{aligned}\bar{T}^\pi(V^\pi)(s_{i+1}) &= (T_{s_k}^\pi \circ \cdots \circ T_{s_1}^\pi)(V^\pi)(s_{i+1}) \\ &= T_{s_{i+1}}^\pi(V^\pi)(s_{i+1}) = T^\pi(V^\pi)(s_{i+1}) = V^\pi(s_{i+1}).\end{aligned}$$

c) In the proof before we looked at $(\bar{T}^\pi)(V^\pi)(s) = (T_s^\pi)(V^\pi)(s)$, but for arbitrary value functions this does not hold (as in a)). Before doing the proof one would need to show that

$$\forall i = 1, ..., K-1: \quad \|\tilde{u}^{(i)} - \tilde{v}^{(i)}\|_\infty \leq \|u - v\|_\infty$$

where $\tilde{u}^{(i)} := (T^\pi_{s_i} \circ \cdots \circ T^\pi_{s_1})(u)$. We dont show this. Then it follows that

$$\begin{aligned}
\|\bar{T}^\pi u - \bar{T}^\pi v\|_\infty &= \|(T^\pi_{s_K} \circ \cdots \circ T^\pi_{s_1})(u) - (T^\pi_{s_K} \circ \cdots \circ T^\pi_{s_1})(v)\|_\infty \\
&= \max_{i=1,\dots,K} \left|(T^\pi_{s_K} \circ \cdots \circ T^\pi_{s_1})(u)(s_i) - (T^\pi_{s_K} \circ \cdots \circ T^\pi_{s_1})(v)(s_i)\right| \\
&= \max_{i=1,\dots,K} \left|(T^\pi_{s_i} \circ \cdots \circ T^\pi_{s_1})(u)(s_i) - (T^\pi_{s_i} \circ \cdots \circ T^\pi_{s_1})(v)(s_i)\right| \\
&= \max_{i=1,\dots,K} \left|(T^\pi_{s_i})(\tilde{u}^{(i-1)})(s_i) - (T^\pi_{s_i})(\tilde{v}^{(i-1)})(s_i)\right| \\
&\leq \gamma \max_{i=1,\dots,K} \left|\tilde{u}^{(i-1)}(s_i) - \tilde{v}^{(i-1)}(s_i)\right| \\
&\leq \gamma \max_{i=1,\dots,K} \left|\|\tilde{u}^{(i-1)} - \tilde{v}^{(i-1)}\|_\infty\right| \\
&\leq \gamma \|u - v\|_\infty.
\end{aligned}$$

□

Algorithm 8: Iterative Policy Evaluation (Totally Asynchronous Updates)

Data : Policy $\pi \in \Pi,\ \epsilon > 0$
Result: Approximation $V \approx V^\pi$

Initialize $V(s) = 0 \quad \forall s \in \mathcal{S}$
$\Delta = 2\epsilon$
while $\Delta \geq \epsilon$ **do**
  $\Delta = 0$
  for $s \in \mathcal{S}$ **do**
    $v = V(s)$

$$V(s) \ = \ \sum_{a \in \mathcal{A}_s} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) \left[r + \gamma V(s') \right]$$

    $\Delta = \max\{\Delta,\ |V(s) - v|\}$
Result: V

3.2 Policy Improvement

Definition 3.3: Policy Improvement

For two policies $\pi, \pi' \in \Pi$ we say that π' is an improvement of π if and only if

$$\forall s \in \mathcal{S}: \quad V^\pi(s) \leq V^{\pi'}(s)$$

and we call it a strict improvement if additionally

$$\exists s' \in \mathcal{S}: \quad V^\pi(s') < V^{\pi'}(s')$$

Clearly, if π is not optimal, then there exists a policy π' that is a strict improvement. Further, as we can easily show (for stationary policies Π^D_S) we have for all $s \in \mathcal{S}$

$$V^\pi(s) = \sum_{a \in \mathcal{A}_s} \pi(a; s) Q^\pi(s, a) \leq \max_{a \in \mathcal{A}_s} Q^\pi(s, a) = \sum_{a \in \mathcal{A}_s} \pi_{Q^\pi}(a; s) Q^\pi(s, a)$$

where π_{Q^π} is the greedy policy wrt. Q^π. Thus, we have it looks like the Greedy policy wrt. Q^π is at least as good as π. But we have not shown this yet! This will now be done in the next theorem.

Theorem 3.4: Policy Improvement Theorem

Let $\pi, \pi' \in \Pi_S$ be two stationary policies.

- We have
$$\forall s \in \mathcal{S}: \quad V^\pi(s) \leq \sum_{a \in \mathcal{A}_S} \pi'(a;s) Q^\pi(s,a) \quad \Rightarrow \quad V^\pi(s) \leq V^{\pi'}(s),$$
i.e. a policy Improvement.
- $$\forall s \in \mathcal{S}: \quad V^\pi(s) < \sum_{a \in \mathcal{A}_S} \pi'(a;s) Q^\pi(s,a) \quad \Rightarrow \quad V^\pi(s) < V^{\pi'}(s),$$
i.e. a strict policy Improvement.
- For every stationary policy $\pi \in \Pi_S$ the greedy policy wrt. Q^π improves π.

Proof. We use in the following always the three conditions of the Bellman expectation operator: Monotonicity, Contraction and $(T^\pi(v+r)) = T^\pi v + r$.
i)
Using the Q-V transfer we get

$$\begin{aligned} V^\pi(s) &\overset{VSS}{\leq} \sum_{a \in \mathcal{A}_s} \pi'(a;s) Q^\pi(a,s) \\ &= \sum_{a \in \mathcal{A}_s} \pi'(a;s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V^\pi(s') \right) \\ &= T^{\pi'} V^\pi(s) \end{aligned}$$

Now using the monotinicty of T^π and the condition above we get

$$V^\pi(s) \overset{above}{\leq} T^{\pi'} V^\pi(s) \overset{monotonicity}{\leq} T^{\pi'} T^{\pi'} V^\pi(s) \leq ... \leq \lim_{n \to \infty} (T^{\pi'})^n V^\pi(s)$$

Now using the Banach fixed point theorem, the above limit converges $\lim_{n\to\infty} (T^{\pi'})^n V^\pi = V^{\pi'}$ (uniformily, i.e. pointwise).
ii)
Analogously, for " $<$ " in the first equation

$$\begin{aligned} V^\pi(s) &\overset{VSS}{<} \sum_{a \in \mathcal{A}_s} \pi'(a;s) Q^\pi(a,s) \\ &= \sum_{a \in \mathcal{A}_s} \pi'(a;s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V^\pi(s') \right) \\ &= T^{\pi'} V^\pi(s) \end{aligned}$$

Altough we have to note here that for a sequence $(a_n)_{n \in \mathbb{N}}$ with limit a it holds

$$\forall n: a_n > a \quad \Rightarrow \quad \lim_{n \to \infty} a_n \geq a$$

and not $\lim_{n\to\infty} a_n > a$ (Example: $a_n = 1/n$)! But this is no problem for the proof, just a thing to be careful of, when considering the limit of $(T^{\pi'})^n$. iii)
In the motivation we have already shown that

$$V^{\pi}(s) = \sum_{a\in\mathcal{A}_s} \pi(a;s)Q^{\pi}(s,a) \leq \max_{a\in\mathcal{A}_s} Q^{\pi}(s,a) = \sum_{a\in\mathcal{A}_s} \pi_{Q^{\pi}}(a;s)Q^{\pi}(s,a)$$

This is exactly the VSS for i), thus the policy improvement follows. □

Lemma 3.5

For $\pi, \pi' \in \Pi_S$ and π' the greedy policy obtained from Q' then it holds that

$$V^{\pi} = V^{\pi'} (\text{or } Q^{\pi} = Q^{\pi'}) \quad \Rightarrow \quad \pi, \pi' \text{ are optimal.}$$

Proof. Because the with a Lemma before

$$V^{\pi} = V^{\pi'} \Rightarrow Q^{\pi} = Q^{\pi'}$$

we will now work with Q. We want to show that $Q^{\pi} = Q^{*} = Q^{\pi\prime}$. Let $\pi' = greedy(Q^{\pi})$ then

$$\begin{aligned} Q^{\pi}(s,a) = Q^{\pi'}(s,a) &= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \sum_{a\in\mathcal{A}_{s'}} p(s';s,a)\pi'(a;s)Q^{\pi'}(s',a') \\ &= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} Q^{\pi'}(s',a') \\ &\overset{Equality}{=} r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} Q^{\pi}(s',a') \\ &= T^{*}Q^{\pi}(s,a) \end{aligned}$$

We showed that $Q^{\pi}(s,a)$ and $Q^{\pi'}(s,a)$ are fixed points of T^* and therefore both π and π' are optimal. □

From the above immediately follows that

Corollary 3.6

For non-optimal policy $\pi \in \Pi_S$ and the greedy policy $\pi' = greedy(Q^{\pi})$ it follows that the greedy policy is an strict improvement to π.

Proof. With one Lemma we had that the greedy Policy is an improvement wrt. to π. In the Lemma directly above we showed that if we have equality of Q^{π} and $Q^{\pi'}$ then both are optimal. Thus if would have a contradiction, if for the non optimal π we had

$$Q^{\pi}(s,a) = Q^{\pi'}(s,a), \quad \text{and} \quad Q^{\pi}(s,a) > Q^{\pi'}(s,a) \quad \text{is not possible.}$$

□

Because in all algorithms we can only approximate Q^{π} and V^{π} we define the following

Definition 3.7

A policy $\pi \in \Pi_S$ is called

- soft, if it satisfies
$$\forall s \in \mathcal{S}, a \in \mathcal{A}_s : \pi(a; s) > 0.$$
- ϵ-soft for some $\epsilon \in (0, 1]$ if
$$\forall s \in \mathcal{S}, a \in \mathcal{A}_s : \pi(a; s) > \frac{\epsilon}{|\mathcal{A}_s|}.$$
(Note: We need to divide by $|\mathcal{A}_s|$ in order to ensure the lower bound stays below 1 and $1 = \sum_{a \in \mathcal{A}_s} \pi(a; s) \geq \epsilon$).
- ϵ-greedy wrt. Q, if we play the best action in state a_s^* with probability $(1 - \epsilon)$ and all other **at least** uniformly on $\mathcal{A}_s \setminus \{a_s^*\}$, i.e.
$$\pi(a; s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}_s|} & , a = a_s^* \\ \frac{\epsilon}{|\mathcal{A}_s|} & , else \end{cases}$$

Similar to the ϵ-greedy learning strategies in stochastic bandits, that grow with linear regret, ϵ-greedy policies for MDPs cannot be optimal as they play a non optimal policy with positive probability. So they typically cannot be guaranteed to improve policies. But we can show that they can improve ϵ-soft policies

Proposition 3.8

Let $\pi \in \Pi_S$ be an ϵ-soft policy and π' the respective ϵ-greedy policy wrt. Q^π, then π' is an improvement to π.

Proof. We will now show the condition of the policy improvement theorem.

$$
\begin{aligned}
V^\pi(s) &= \sum_{a\in\mathcal{A}_s} \pi(a;s) Q^\pi(s,a) \\
&= \sum_{a\in\mathcal{A}_s} \pi(a;s) Q^\pi(s,a) + \sum_{a\in\mathcal{A}_s} \frac{\epsilon}{|\mathcal{A}_s|} Q^\pi(s,a) - \sum_{a\in\mathcal{A}_s} \frac{\epsilon}{|\mathcal{A}_s|} Q^\pi(s,a) \\
&= \sum_{a\in\mathcal{A}_s} \frac{\epsilon}{|\mathcal{A}_s|} Q^\pi(s,a) + \sum_{a\in\mathcal{A}_s} (\pi(a;s) - \frac{\epsilon}{|\mathcal{A}_s|}) Q^\pi(s,a) \\
&= \sum_{a\in\mathcal{A}_s} \frac{\epsilon}{|\mathcal{A}_s|} Q^\pi(s,a) + (1-\epsilon) \sum_{a\in\mathcal{A}_s} \underbrace{\frac{\pi(a;s) - \frac{\epsilon}{|\mathcal{A}_s|}}{1-\epsilon}}_{\geq 0, \epsilon-\text{soft}} Q^\pi(s,a) \\
&\leq \sum_{a\in\mathcal{A}_s} \frac{\epsilon}{|\mathcal{A}_s|} Q^\pi(s,a) + (1-\epsilon) \max_{a'\in\mathcal{A}_s} Q^\pi(s,a') \underbrace{\sum_{a\in\mathcal{A}_s} \frac{\pi(a;s) - \frac{\epsilon}{|\mathcal{A}_s|}}{1-\epsilon}}_{=\frac{1-\epsilon}{1-\epsilon}} \\
&= \sum_{a\in\mathcal{A}_s\setminus\{a_s^*\}} \frac{\epsilon}{|\mathcal{A}_s|} Q^\pi(s,a) + (1-\epsilon) \max_{a'\in\mathcal{A}_s} Q^\pi(s,a') + \frac{\epsilon}{|\mathcal{A}_s|} \max_{a'\in\mathcal{A}_s} Q^\pi(s,a) \\
&= \sum_{a\in\mathcal{A}_s} \pi'(a;s) Q^\pi(s,a).
\end{aligned}
$$

□

3.3 Policy Iteration algorithms (tabular actor critic)

We now alternate between the steps of Policy evaluation and improvement and thus get the Policy Iteration Algorithm in its most simple form.

Algorithm 9: Greedy Exact Policy Iteration (Actor-Critic)

Data : initial policy $\pi \in \Pi$, initial value function V
Initialize V_{new}, π_{new}
stop = False
while *stop* = *False* **do**
  `// Policy Evaluation (critic): compute` V^π `(e.g. via Bellman equation).`

$$Q^\pi(s,a) = \sum_{s'\in\mathcal{S}} \sum_{r\in\mathcal{R}} p\big(s', r; s, a\big) \Big[r + \gamma\, V\big(s'\big) \Big]$$

  `// Policy Improvement: derive greedy policy` π_{new} `from` Q^π.
  if $Q^{\pi_{\text{new}}} = Q^\pi$ **then**
    stop = True
  $\pi = \pi_{\text{new}}$
return π^*
Result: optimal policy $\pi^* \in \Pi$

Theorem 3.9: Greedy exact policy iteration

Initialized in any policy the algorithm above terminates in finite number of iterations (at most $|\mathcal{A}| \cdot |\mathcal{S}|$ steps (this is the number of deterministic possible)) in a finite MDP and its output is the solution of the Bellman optimality equation and thus π^*.

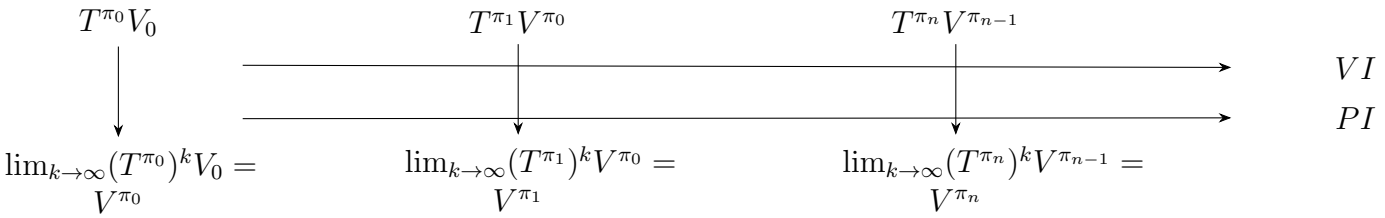
Proof. First, note that in every improvement step, we have a strict improvement as long as we have not reached the optimal policy. Because the Bellman Equation has a unique solution, we never visit a policy twice, because every Policy only has one Q-Value. Finally, due to the fact that there are only finitely many deterministic policies, the time needed to reach optimal policy is at most the number of policies that exist, i.e. $|\mathcal{A}| \cdot |\mathcal{S}|$. □

3.4 Quick comparison of Value iteration and Policy Iteration

Policy Iteration (PI) and Value iteration (VI) are very similiar: Let π_n be the policy obtained from PI (i.e. $\pi_n = greedy(V^{\pi_{n-1}})$) at step $n \in \mathbb{N}$ then

$$V^{\pi_n} \overset{(n\to\infty)}{\leftarrow} T^{\pi_n} V^{\pi_{n-1}} = T^* V^{\pi_{n-1}} \overset{(n\to\infty)}{\rightarrow} V^*$$

Thus, Value iteration is approximating V^π always using one "Banach fixed point theorem step", i.e. for an initial vector V_0 we have



VI only does one step and PI multiple.
Performance: First, it is important to note that Policy iteration and Value iteration have the same upper bound, but lower bounds are not clear (not done here).
1. Computational costs vs. Speed:
We define

$$T^n := (T^{\pi_n} \circ .... \circ T^{\pi_0})$$

where $\pi_n := greedy(V^{\pi_{n-1}})$. Then due to $T^*V = T^{\pi_V}V$ for $\pi_V := greedy(V)$ we have for every initial vector V:

$$\lim_{n\to\infty} T^n V = V^* = \lim_{n\to\infty} (T^*)^n V.$$

This is Value iteration. Now due to

$$\begin{aligned}\lim_{n\to\infty} T^n V &= \lim_{n\to\infty} (T^{\pi_n} \circ .... \circ T^{\pi_0})V = \lim_{n\to\infty} ((T^{\pi_n})^2 \circ .... \circ (T^{\pi_0})^2)V \\ &= ... = \lim_{n\to\infty} \lim_{k\to\infty} ((T^{\pi_n})^k \circ .... \circ (T^{\pi_0})^k)V \end{aligned} \tag{2.1}$$

we define

$$T_k^n := ((T^{\pi_n})^k \circ .... \circ (T^{\pi_0})^k)$$

and get

$$\lim_{n\to\infty}\lim_{k\to\infty} T_k^n V = V^*.$$

This is policy iteration. Due to the fact that $cost(T^\pi) < cost(T^*)$ choose $k \in \mathbb{N}$ such that

$$k \cdot n \cdot cost(T^\pi) \leq n \cdot cost(T^*), \quad n \in \mathbb{N}.$$

Using contractions it holds for every initial vector V :

$$\|T_k^n V - V^*\|_\infty \leq \gamma^{kn}\|V - V^*\|_\infty \quad \text{and} \quad \|T^n V - V^*\|_\infty \leq \gamma^n \|V - V^*\|_\infty.$$

Thus, for the same costs PI is better than VI by a factor of γ^k.

2. Variance of T^π and T^*:
Next, we show that the standard Monte Carlo Estimator for

$$T^\pi Q(s,a) = \mathbb{E}_{s,a}^\pi[R_0 + \gamma Q(S_1, A_1)]$$

is smaller than the one for

$$T^* Q(s,a) = \mathbb{E}_{s,a}^\pi[R_0 + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q(S_1, a')].$$

This is due to: For all $s \in \mathcal{S}$

$$\begin{aligned}\max_{a\in\mathcal{A}_s} Q(s,a) - \mathbb{E}[\max_{a\in\mathcal{A}_S} Q(S,a)] &= \mathbb{E}[\max_{a\in\mathcal{A}_s} Q(s,a) - \max_{a\in\mathcal{A}_S} Q(S,a)] \\ &\geq \mathbb{E}[Q(s,a') - Q(S,a'')] \end{aligned} \tag{2.2}$$

for arbitrary $a', a'' \in \mathcal{A}_s$. This yields

$$\begin{aligned}\mathbb{V}[\max_{a\in\mathcal{A}_s} Q(S_1,a)] &= \sum_{s\in\mathcal{S}} \mathbb{P}(S_1 = s)\left(\max_{a\in\mathcal{A}_s} Q(s,a) - \mathbb{E}[\max_{a\in\mathcal{A}_s} Q(S_2,a)]\right)^2 \\ &= \sum_{s\in\mathcal{S}} \mathbb{P}(S_1 = s)\left(\max_{a\in\mathcal{A}_s} Q(s,a) - \mathbb{E}[\max_{a\in\mathcal{A}_s} Q(S_2,a)]\right)^2 \sum_{a\in\mathcal{A}_s} \mathbb{P}(A_1 = a) \\ &\overset{above}{\geq} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}_s} \mathbb{P}(S_2 = s)\left(Q(s,a) - \mathbb{E}[Q(S_2,A_2)]\right)^2 \mathbb{P}(A_1 = a) \\ &= \mathbb{V}[Q(S_1,A_1)].\end{aligned}$$

Therefore,

$$\mathbb{V}[R_0 + \gamma \max_{a\in\mathcal{A}_{S_1}} Q(S_1,a)] \geq \mathbb{V}[R_0 + \gamma Q(S_1,A_1)].$$

If the Environment has a high variance, i.e. the conditional distribution $(S \mid S = s, A = a)$ has high a variance, then T^* handles this worse than T^π.

3. Connection to Markov Chains:

Definition 3.10: Mixing Times

For a Markov Chain $(X_n)_{n\in\mathbb{N}}$ taking values in $(E, \mathcal{B}(E))$ with unique stationary distribution ν we define the mixing time as

$$\tau_{mix}(\epsilon) := \min\{n \in \mathbb{N} \mid \max_{x\in E} \|\mathbb{P}_x(X_n = \cdot) - \nu(\cdot)\|_{TV} \leq \epsilon\}, \quad \epsilon > 0,$$

where

$$\|\mu - \nu\|_{TV} = \tfrac{1}{2}\sum_{y\in\mathcal{S}} |\mu(y) - \nu(y)|.$$

is called total variation.

The mixing time discribes the minimum time that is needed such that the MCMC Chain has at least a distance of ϵ to the stationary distribution wrt. $\|\cdot\|_{TV}$.

Definition 3.11: Spectral Gap

For a transition matrix $P \in [0,1]^{n\times n}$ with eigenvalues $0 \leq \lambda_n \leq .... \leq \lambda_2 \leq \lambda_1 = 1$ we define

$$Gap(P) := 1 - |\lambda_2|.$$

The smaller $|\lambda_2|$ is, the quicker the Markovchain "forgets" initial conditions.

Lemma 3.12

For an irreducible, lazy and reversible Markov Chain with unique stationary distribution ν it holds that

$$\tau_{mix}(\epsilon) \leq \frac{1}{Gap(P)}\left(\log(\frac{1}{\min_x \nu(x)}) + \log(\frac{1}{\epsilon})\right), \quad \epsilon > 0.$$

Proof. David A. Levin, Yuval Peres, Elizabeth L. Wilmer: Markov Chains and Mixing Times, American Mathematical Society, 2009. Theorem 12.4 (Page 189). □

Remark 3.6

Thus, if we consider the MCMC-algorithm, it converges faster (i.e. faster mixing time), if the Spectral Gab of the transition matrix P is high.
Similar to the MCMC method we can interpret in VI the connection of the Spectral Gap to the speed of the method, because the transition matrix $p(s'; s, a)$ is a probability Matrix and (S, A) is Markov Chain. Although we would need to assume irreducibility, lazyness and reversibility on the transition probabilities. And even if $p(s'; s, a)$ satisfies all these conditions, it is not clear wether the result of speed applies to T^π or even T^*. (Probably not T^* as it is non linear).

4. Application to Windy Cliff Walk:
In Windy Cliff walk if the transition probabilities are deterministic (no wind) the variance is zero. As the wind increases, the variance of transition probabilities increases and the performance of VI should get worse and worse compared to PI. Further, as long as the Spectral Gap of $p(s'; s, a)$ increases PI should improve in performance. Finally, in all cases with the same computational costs, PI will be better than VI.

Proposition 3.13

(Vergleich der Mischzeiten via Spektrallücken) Seien P, P' zwei irreduzible, lazy und reversible Transition-Matrizen auf demselben endlichen Zustandsraum $\mathcal{S}$. Bezeichne

$$\text{gap}(P) \;=\; 1-\lambda_2(P) \quad \text{und} \quad \text{gap}(P') \;=\; 1-\lambda_2(P'),$$

wobei $\lambda_2(\cdot)$ den zweitgrößten Eigenwert in Absolutbetrag bezeichnet. Beide Matrizen besitzen jeweils eine eindeutige stationäre Verteilung ν bzw. ν', und wir definieren die Mischzeiten gemäß

$$\tau_{mix}(\epsilon) \;=\; \min\Big\{n\in\mathbb{N}\;\Big|\;\max_{x\in\mathcal{S}}\big\|\mathbb{P}_x(X_n=\cdot)-\nu(\cdot)\big\|_{TV} \;\le\; \epsilon\Big\},$$

bzw. analog für P'.
Gilt nun

$$\text{gap}(P') \;>\; \text{gap}(P),$$

so folgt aus dem Spektrallücken-Formalismus, dass

$$\tau'_{mix}(\epsilon) \;<\; \tau_{mix}(\epsilon) \quad \text{für jede } \epsilon>0.$$

Mit anderen Worten, die Markov-Kette mit der größeren Spektrallücke besitzt eine (asymptotisch) kürzere Mischzeit und konvergiert schneller zu ihrer stationären Verteilung.

Proof. Dies ist eine direkte Folge der Standardabschätzungen zur Mischzeit irreduzibler, lazy und reversibler Markov-Ketten (vgl. Theorem 12.4 in Levin–Peres–Wilmer: *Markov Chains and Mixing Times*). Dort wird u.a. gezeigt, dass

$$\tau_{mix}(\epsilon) \;\le\; \frac{1}{\text{gap}(P)}\Big(\log\big(\tfrac{1}{\nu_{\min}}\big)+\log\big(\tfrac{1}{\epsilon}\big)\Big),$$

wobei $\nu_{\min}=\min_{x\in\mathcal{S}}\nu(x)$ und $\text{gap}(P)=1-\lambda_2(P)$. Eine analoge Ungleichung gilt für P'. Wegen $\text{gap}(P')>\text{gap}(P)$ ergibt sich unmittelbar $\tau'_{mix}(\epsilon)<\tau_{mix}(\epsilon)$ für kleine $\epsilon>0$. □

Proposition 3.14

(Spektrallücke und beschleunigte Policy-Evaluation) Betrachte ein MDP mit endlicher Zustandsmenge $\mathcal{S}$ und festem Diskontfaktor $0<\gamma<1$. Sei π eine stationäre, stochastische Politik und

$$P^{\pi}(s,s') \;=\; \sum_{a\in\mathcal{A}_s}\pi(a;s)\,p(s';\,s,a)$$

die zugehörige Transitionsmatrix, wobei $p(s';\,s,a)$ die Wahrscheinlichkeit bezeichnet, von Zustand s durch Ausführung von Aktion a nach s' zu gelangen. Angenommen, P^{π} ist irreduzibel, lazy und reversibel, und es existiert eine eindeutige stationäre Verteilung ν^{π}.
Sei π' eine zweite Politik mit analogen Eigenschaften. Falls

$$\text{gap}(P^{\pi'}) \;>\; \text{gap}(P^{\pi}),$$

so gilt: Die stochastischen Prozesse (X_n) nach $P^{\pi'}$ *mischen* schneller. Bei Verfahren zur Policy-Evaluation (z. B. Monte-Carlo- oder Temporal-Difference-Methoden) führt dies zu einer schnelleren Konvergenz auf die wahren Werte $Q^{\pi'}$ bzw. $V^{\pi'}$ im Vergleich zur Evaluierung unter π.

Proof. Die Policy-Evaluation basiert auf wiederholten Besuchen der Markov-Kette unter Politik π. Eine größere Spektrallücke $\text{gap}(P^{\pi'})$ impliziert eine kürzere Mischzeit $\tau_{mix}^{\pi'}(\epsilon)$. Für Monte-Carlo- oder TD-Verfahren bedeutet dies, dass die Zustandsverteilungen sich schneller an ihre stationäre Verteilung anpassen, was statistisch gesehen die Varianz von Schätzungen für Q^π und V^π reduziert und somit zu schnelleren Konvergenzraten führt. □

Proposition 3.15

(Policy Iteration und Folgen von Transitionsmatrizen) Sei $(\pi_k)_{k\in\mathbb{N}}$ eine *Folge* von Politiken, die während einer Policy Iteration (oder deren Varianten) entsteht. Jede Politik π_k besitzt eine zugehörige Transitionsmatrix P^{π_k}, welche irreduzibel, lazy und reversibel ist, mit eindeutiger stationärer Verteilung ν^{π_k}.
Ist für alle k

$$\text{gap}(P^{\pi_k}) \;\geq\; \gamma_0 > 0,$$

mit einer *konstanten* Untergrenze γ_0, so konvergieren die zugehörigen Policy-Evaluation-Schritte (z. B. mittels Monte-Carlo auf P^{π_k}) in jeder Iteration k *gleichartig schnell*. Weiterhin, falls sich die Spektrallücke sogar *vergrößert* mit k, kann man strengere (bessere) Konvergenzabschätzungen von Iteration zu Iteration formulieren.

Proof. Die wesentliche Idee ist, dass jeder Evaluationsschritt in einer Iteration k (zu Politik π_k) ein stochastisches Verfahren ist, das auf der Markov-Kette P^{π_k} basiert. Falls sämtliche P^{π_k} eine Spektrallücke oberhalb eines festen γ_0 aufweisen, sind die zugehörigen Mischzeiten $\tau_{mix}^{\pi_k}(\epsilon)$ einheitlich nach oben beschränkt. Damit erhält man eine gleichförmige Konvergenzgeschwindigkeit für alle Teilschritte der Policy Iteration. Wenn $\text{gap}(P^{\pi_{k+1}}) > \text{gap}(P^{\pi_k})$, verbessert sich diese Geschwindigkeit sogar von einer Iteration zur nächsten. □

CHAPTER 3

SIMULATION BASED DYNAMIC PROGRAMMING METHODS

In reality there are two problems: We do not know p and $\mathcal{A}$ and $\mathcal{S}$ might be too big. Thus we first have the problem that we do not know T^* or T^π. The second problem is that $\mathcal{A}$ and $\mathcal{S}$ might be too big to explore in its entirety or that inverting the Q-Value matrix, might lead to bad conditioning, i.e.

$$cond(Q) = \|Q\|\|Q^{-1}\|.$$

Definition 0.1

An algorithm to learn optimal policies is called model-based if the algorithm requires explicit knowledge on the Markov decision model. A solution algorithm is called model-free if the algorithm only requires the ability to sample all appearing random variables.

We will now consider only model-free methods.

1 A first example

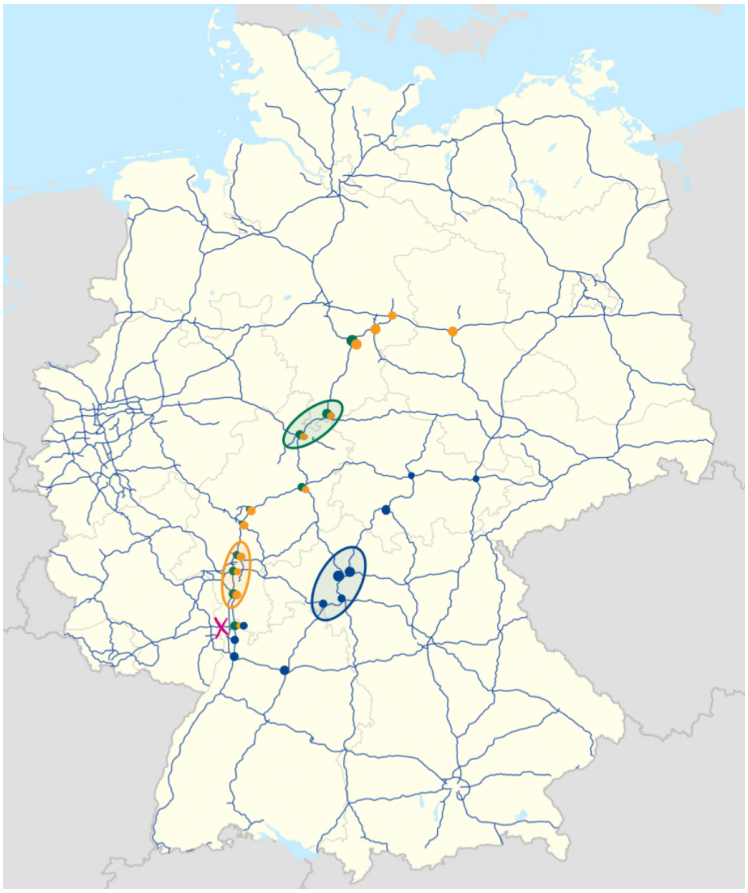


Figure 3.1

Imagine you want to travel from Berlin to Mannheim and want to reduce the travel time. The dots on the map are Autobahn intersections. In order to choose the shortest path, one needs to find an optimal policy, that takes the shortest path in every state. In order to evaluate this policy, we need an estimate for all trajectories. One first approach would be to do Monte Carlo estimation of the State Value function, as this is an expectation. One can do this by sampling different trajectories and calculating the rewards. This method is computationally extensive and not very good. There are multiple ways to improve this approach. We will first focus on Monte Carlo estimation methods.

2 Monte Carlo Policy evaluation and control

Definition 2.1: Rollout

A trajectory $(S_t, A_t, R_t)_{t=1,\dots,n}$ of a MDP is called Rollout.

- One first approach would be to improve the variance of the estimation. Imagine we have two unbiased estimators $\hat{X}$ and $\bar{X}$. Then one could show that for all $\alpha \in (0,1)$

$$\alpha\hat{X} + (1-\alpha)\bar{X}$$

is also an unbiased estimator and with a lower variance. If we choose $\alpha = \frac{n-1}{n}$ then we have the memory trick, if we assume that $\hat{X}$ is an old estimate and $\bar{X}$ is a new estimate of $\mathbb{E}[f(X)]$. We will later see that choosing $\alpha_n = \frac{1}{n}$ is very good (Robins-Monro).

- Using the estimator

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T}\gamma^t R_t^i, \quad T \to \infty$$

for $\mathbb{E}[\sum_{t=0}^{\infty}\gamma^t R_t]$ will always be biased, because we cannot simulate $T = \infty$. A workaround will be the trick, where we showed that for $T \sim Geo(1-\gamma)$

$$\mathbb{E}[\sum_{t=0}^{\infty}\gamma^t R_t] = \mathbb{E}[\sum_{t=0}^{T} R_t],$$

thus the estimator

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T_i} R_t^i, \quad T_1, ..., T_N \overset{iid}{\sim} Geo(1-\gamma)$$

is unbiased.

The with these two approaches is still, that they are sample ineffective and computationally extensive.

Remark 2.1

Sample efficiency means that algorithms are supposed to use as few random samples as possible. In the context of Monte Carlo this means to extract as much information as possible from any rollout (S,A,R) of the Markov decision process.

2.1 First Visit Monte Carlo Policy Evaluation

A way to improve the sample efficiency of the Monte Carlo method, is to do bootstrapping, i.e. we reuse samples, if we have the opportunity to do so. In the example of the Highway from Berlin to Mannheim: Some paths may overlap and thus the time that was estimated on a specific overlapping section can be reused. We to this similarly in order to reuse samples for the Monte Carlo estimation. In order to do so, consider the following:

Definition 2.2: Stopping Time

Let $X := (X_t)_{t\in\mathbb{N}}$ be a stochastic process on $(\Omega, \mathcal{F}, \mathbb{P})$ taking values in E.We call the random variable $T : \Omega \to \mathbb{N}_0 \cup \{\infty\}$ a stopping time wrt. the process X, if

$$\forall B \in \epsilon^{\otimes(n+1)} : \quad \{T = n\} := (X_0, ..., X_n)^{-1}(B) \in \sigma(X_0, ..., X_n) = \mathcal{F}_n^X.$$

Theorem 2.3: Strong Markov Property

Let $X := (X_t)_{t\in\mathbb{N}}$ be a (ν, P)-Markov Chain on the state space E and T a stopping time. Then it holds for all $A \in \epsilon^{\otimes\mathbb{N}_0}$, $F \in \mathcal{F}_T^X$ and $x \in E$ with $\mathbb{P}(F, X_T = x, T < \infty) > 0$ that

$$\mathbb{P}_\nu((X_T, X_{T+1}, ...) \in A \mid F, X_T = x, T < \infty) = \mathbb{P}_x((X_0, X_1, ...) \in A).$$

Here $\{T < \infty\}$ is defined by $X_T(\omega) = X_{T(\omega)}(\omega)$.

Now we can show that Bootstrapping works. Lets consider we take a rollout $(S_t, A_t, R_t)_{t=0,...,2N}$ that is started in $S_0 = s$. Using this rollout we get one sample for $V^\pi(s)$. If we want one estimate for $V^\pi(s')$, but we do not want to get another rollout, i.e. resample, then consider the first hitting time of state s' (which is a stopping time) denoted by

$$T_{s'} := \inf\{t \in \mathbb{N}_0 \mid S_t(\omega) = s'\}.$$

In order to be the first hitting time before time step $n + 1$ we redefine it as

$$T_{s'} := \inf\{t \in \mathbb{N}_0 \mid S_t(\omega) = s', t \le n\}.$$

Now using the same rollout that started in $s \in \mathcal{S}$ we can get one sample for $V^\pi(s')$, because due to the strong Markovproperty, after the first hitting time $T_{s'}$ the Markovchain resets. Note that the Markov Chain is (S_t, A_t). Using this we can now construct the following algorithm that reuses samples:

Algorithm 10: First-visit Monte Carlo policy evaluation of V^π

Input : Policy $\pi \in \Pi$, approximate value function $V \approx V^\pi$, discount γ, initial state distribution μ
Initialize $V(s) = 0$ and $N(s) = 0$ for all states s;
while *not converged* **do**
  $n = n + 1$;
  Sample $T \sim \text{Geo}(1 - \gamma)$;
  Sample $s_0 \sim \mu$;
  Generate a trajectory $(s_0, a_0, r_0, s_1, \dots, s_{2T})$ using policy π;
  for $t = 0$ **to** T **do**
    if $s_t \notin \{s_0, ..., s_{t-1}\}$ **then**
      $v = \sum_{k=t}^{t+T} r_k$;
      $N(s_t) = N(s_t) + 1$;
      $V(s_t) = V(s_t) + \frac{1}{N(s_t)}\big(v - V(s_t)\big)$;

return V **Output:** Value function V

Due to the strong markov property, we can restart a MDP, after reaching a hitting time. Thus if T_s is the first hitting time of state s then

$$V^\pi(s) = \mathbb{E}_s[\sum_{t=0}^{\infty} \gamma^t R_t] = \mathbb{E}[\sum_{t=T_s}^{\infty} \gamma^{t-T_s} R_t \mid T_s < \infty]$$

Let $\tau_1, ..., \tau_n \overset{iid}{\sim} \pi$ be Rollouts from a policy, i.e. $\tau_i := (s_0^i, a_0^i, r_0^i, ..., s_{2N}^i, a_{2N}^i, r_{2N}^i)$. Note that we want to have for every sample of the Policy the same number of steps. We choose this to be $N \sim Geo(1 - \gamma)$ steps. We define the first visit Monte Carlo estimator for the value function as

$$\hat{V}_{n,N}^\pi(s) := \frac{1}{n} \sum_{i=1}^{n} v^{i,N}(s),$$

where $T_s^i := \inf\{t = 1, ..., N \mid S_t^i(\omega) = s\}$ and

$$v^{i,N}(s) := \frac{1}{N} \sum_{j=T_s^i}^{T_s^i+N} (\gamma^{j-T_s^i}) R_j^i \mathbf{1}_{T_s^i \leq N}.$$

Note that the $v^{i,N}(s)$ are just zero due to the indicator in the case that the state $s \in \mathcal{S}$ is not reached. Because every Rollout is iid it follows that all $v^{1;N}(s), ..., v^{n,N}(s)$ are iid, as each estimator only depends on one roullout. Thus for every $s \in \mathcal{S}$ we give the MDP at most N time steps to reach the respective state. Every visit Monte Carlo is then defined for the k^{th} hitting time of state $s \in \mathcal{S}$

$$T_s^i(k) := \inf\{T_s^i(k-1) < t \leq N \mid S_t^i(\omega) = s\}$$

and the sample for the value function, always when state s is hit

$$\tilde{V}_{n,N}^{\pi}(s) := \frac{1}{\sum_{i,k=1}^{n} \mathbf{1}_{T_s^i(k)<N}} \sum_{i,k=1}^{n} v_k^{i,N}(s),$$

where

$$v_k^{i,N}(s) = \frac{1}{N} \sum_{j=T_s^i(k)}^{T_s^i(k)+N} (\gamma^{t-T_s^i(k)}) R_j^i \mathbf{1}_{T_s^i(k)\leq N}.$$

Again, due to the indicators $v_k^{i,N}(s)$ are zero if the state $s \in \mathcal{S}$ is not reached for the k^{th} time. Here the we have more samples, but for all $i = 1, ..., n$ each estimator $v_1^{i,N}, ..., v_{\max_k T_s^i(k)}^{i,N}$ are dependend, as they come from the same roullout. Here again, we give the MDP at most N time steps to reach the state $s \in \mathcal{S}$ once or more.

Remark 2.2

The first visit monte Carlo estimator satisfies

$$\mathbb{E}[\hat{V}_{n,N}^{\pi}(s)] = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}[v^{i,N}(s)] = V^{\pi}(s) \iff N = \infty$$

and also if we use leave the γ and make $N \sim Geo(1-\gamma)$. Then variance is given by

$$\mathbb{V}[\hat{V}_{n,N}^{\pi}(s)] = \frac{1}{n^2} \sum_{i=1}^{n} \mathbb{V}[v^{i,N}(s)]$$

The every visit Monte Carlo estimator satisfies

$$\mathbb{E}[\tilde{V}_{n,N}^{\pi}(s)] := \frac{1}{\sum_{i,k=1}^{n} \mathbf{1}_{v_k^{i,N}(s)>0}} \sum_{i,k=1}^{n} \mathbb{E}[v_k^{i,N}(s)] = V^{\pi}(s) \iff N = \infty$$

or using the trick again with the geometric time horizon after every hitting time. The variance is higher for the every visit Monte Carlo estimator as, due to the dependence, the covariance is not zero

$$\mathbb{V}[\tilde{V}_{n,N}^{\pi}(s)] = \left(\frac{1}{\sum_{i,k=1}^{n} \mathbf{1}_{T_s^i(k)<N}} \right)^2 \sum_{i,k=1}^{n} \mathbb{V}[v_k^{i,N}(s)] + \sum_{\substack{l,L,m,M=1 \\ l \neq L \neq m \neq M}}^{n} Cov(v_l^{m,N}(s), v_L^{M,N}(s)).$$

Theorem 2.4

If we visit every state infinitely often for the (a, s) with first visit Monte carlo method with the policy and in each step choose $N \sim Geo(1-\gamma)$ then $\hat{V}_n^\pi(s) \to V^\pi(s)$ for $n \to \infty$ $a.s.$

Proof. LLN, because every sample is iid and we assume bounded rewards. □

For first visit Monte Carlo, thus we need to force exploration in the policy π and initial distribution μ in order to ensure convergence. Thus we can never really evaluate greedy policies. As in bandits, one will first force exploration and then let this forced exploration go to zero, i.e.

$$\pi + explorationBonus(n), \quad explorationBonus(n) \to 0, n \to \infty.$$

2.2 Generalised policy iteration with first visit Monte Carlo estima- tion

CHAPTER 4

FINITE TIME MDPS

Now the optimization goal is

$$\pi^* \in \arg\max_{\pi} \mathbb{E}^{\pi}[\sum_{t=0}^{T} R_t]$$

for fixed $T \in \mathbb{N}$. Here the idea of dynamic programming becomes much clearer, as its idea was to reduce a big problem into smaller subproblems. It will turn out the equations are backwards recursions, beginning from the End.
Π_t^T denotes the set of polices $(\pi_i)_{i=t,\dots,T-1}$. Note that the following definition might seem a little bit cruel, but it is neseccary in order to avoid conditioning on an event with probability zero.

Definition 0.1: Value functions

For any $\pi \in \Pi_t^T$ we define $V_{T,T} \equiv 0$ and

$$V_{t,T}^{\pi}(s) = \mathbb{E}_s^{\tilde{\pi}}[\sum_{i=0}^{T-t-1} R_i] =: \mathbb{E}^{\pi}[\sum_{i=t}^{T} R_i \mid S_t = s]$$

the time-state value function from time t to T, and $\tilde{\pi} := \pi_{\cdot+t}$ is the shifted policy by t. $V_{0,T}$ is the state value function.
We define $Q_{T,T} \equiv 0$ and

$$Q_{t,T}^{\pi}(s,a) = \mathbb{E}_{s,a}^{\tilde{\pi}}[\sum_{i=0}^{T-1-t} R_i] =: \mathbb{E}^{\pi}[\sum_{i=t}^{T} R_i \mid S_t = s, A_t = a].$$

Proposition 0.2

Again it holds that

$$V_{t,T}(s) = \sum_{a \in \mathcal{A}_s} \pi_t(a;s) Q_{t,T}(s,a)$$

and the Q-V transfer

$$Q_{t,T}^{\pi}(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V_{t+1,T}^{\pi}(s').$$

The Bellman expectation operators are backwards recursions:

$$V^{\pi}_{t,T}(s) = \sum_{a \in \mathcal{A}_s} \pi_t(a;s)(r(s,a) + \sum_{s' \in \mathcal{S}} p(s';s,a) V^{\pi}_{t+1;T}(s'))$$

and

$$Q^{\pi}_{t,T}(s,a) = r(s,a) + \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} p(s';s,a)\pi_{t+1}(a';s') Q^{\pi}_{t+1,T}(s',a')$$

The optimal value function now depends on the remaining time

Definition 0.3: Optimality

For all $t \leq T$
The optimal state value function

$$\forall s \in \mathcal{S}: \quad V^{*}_{t,T}(s) := \sup_{\pi \in \Pi^T_t} V^{\pi}_{t,T}(s)$$

The optimal state action value function

$$\forall (s,a) \in (\mathcal{S} \times \mathcal{A}): \quad Q^{*}_{t,T}(s,a) := \sup_{\pi \in \Pi^T_t} Q^{\pi}_{t,T}(s,a)$$

A policy π^* is optimal if its value function equals the optimal value function for every state and time:

$$\forall s \in \mathcal{S}\ t \leq T: \quad V^{\pi^*}_{t,T}(s) = V^{*}_{t,T}(s)$$

Theorem 0.4: Dynamic Programming

Let $v_t : \mathcal{S} \to \mathbb{R}$ and $q_t : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be seqeuneces $(v_t)_{t \leq T}$ and $(q_t)_{t \leq T}$ that satisfy the Bellman optimality equations as backwards recursions. Then it follows that

$$\forall t \leq T: \quad v_t = V^{*}_{t,T} \quad \text{and} \quad q_t = Q^{*}_{t,T}.$$

Further, a optimal non stationary policy is given by the greedy polciy

$$\pi_t(a;s) := \begin{cases} 1 & , a \in \arg\max_a q_t(s,a) \\ 0 & , else. \end{cases}$$

1 Stochastic Fixed Point Iterations

Suposse we want to solve a fixed point Iteration $F(x) = x$ using the Banach fixed point theorem, but we do not know F and can only sample from it and thus can only work with stochastic approximations $\hat{F(\cdot)} = F(\cdot) + \epsilon(\cdot)$. The question is, does the iteration

$$x_{n+1} = \hat{F(x_n)}$$

converge to x^*. The answer is in general no, due to the fact the errors are too large, if we do Monte Carlo approximations of F. One can show that the iteration

$$x_{n+1} = (1 - \alpha_n)x_n + \alpha_n F(\hat{x}_n)$$

converges under certain conditions, where the Errors are unbiased and small enough and the step size decays in a certrain speed. Note that for $\alpha \equiv 1$ we have Banach fixed point iteration and for $\alpha \equiv 0$ we have constant sequence.

Remark 1.1

In order to get a little bit of a feeling on how to build algroithms in a model free way, we will now focus on finding roots of functions without any derivatives. These functions can also be expectations.

The idea is as follows. Consider we want to find the root x^* of $G(x)$, but we only have access to stochastic approximations

$$\tilde{G}(x) = G(x) + \epsilon,$$

where the error are centered, i.e. mean zero. The most generic Robbins-Monro iteration scheme is

$$x_{n+1} = x_n + \alpha_n(G(x) + \epsilon)$$

where α_n is our step size. This is more or less a stochastic version of the Newton method.

Example 1.1

The Problem is that we want to find for the function

$$G(x) = \mathbb{E}[g(x, Y)], \quad Y \sim P$$

the root. We define the approximation as

$$\Psi^{n,n+1}x_0 := x_\Delta^{n+1} := x_\Delta^n + \alpha_n \hat{G}(x_\Delta^n), \quad x_\Delta^0 := x_0$$

and the true path

$$\Phi^{n,n+1}x_0 := x^{n+1} := x^n + \alpha_n G(x^n)$$

where

$$\hat{G}(x) = G(x) + \epsilon, \quad \epsilon \sim (0, C), \text{where } C = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ \cdots & \cdots & & \cdots \\ 0 & \cdots & & \sigma_d^2 \end{pmatrix}$$

As the errors accumulate over each step of the discrete evolution, we have to consider the following: If $\Phi^{\cdot,\cdot}$ is asymptotically stable in different regions of non unique fixed points $x^* \in A^*$ (but only some are desired to be reached), does the discrete evolution $\Psi^{\cdot,\cdot}$ inherit this property in finite time? This is a to complicated Problem! We only consider the that we have infinite time and $\Phi^{\cdot,\cdot}$ is globally asymptotically stable, i.e. only one unique fixed point. This can also be stated in a different perspective, i.e. let G be a contraction and force the random step size to decrease as it the step size goes to infinity. In this setting, considering the error that the discrete evolution does wrt. the true evolution in each step does not matter anymore and we can just formulate it as

$$x_{n+1} := x_n + \alpha_n \hat{G}(x_n).$$

Theorem 1.1: Simplest Robins Monro Algorithm

Let $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{N}}, \mathbb{P})$ be a filtered probability space on which all the following random variables are defined. Suposse the function $G : \mathbb{R} \to \mathbb{R}$ has a root x^* and satisfies

$$\exists \kappa > 0 : \quad G(x) \geq \kappa(x - x^*).$$

Then we define the recursive stochastic process

$$x_{n+1} = x_n - \alpha_n \underbrace{(G(x_n) + \epsilon_n)}_{=:y_n}$$

where the following conditions are met

- The deterministic step-sizes satisfy

$$\sum_{n=1}^{\infty} \alpha_n = \infty \quad \text{and} \quad \sum_{n=1}^{\infty} \alpha_n^2 < \infty$$

- The errors ϵ_n are $\mathcal{F}_{n+1}$-measurable and are conditionally on the past unbiased, i.e. $\mathbb{E}[\epsilon_n \mid \mathcal{F}_n] = 0$
- The stochastic approximation satisfies

$$\mathbb{E}[y_n^2] \leq A + B\mathbb{E}[x_n - x^*]^2.$$

Then $x_n \overset{L^2}{\to} x^*$ for $n \to \infty$.

Proof. Only Idea:
We define $z_n := \mathbb{E}[(x_n - x^*)^2]$ and show that

$$z_{n+1} \leq z_n(1 - \underbrace{2\alpha_n\kappa}_{=:a_n}) + \underbrace{\alpha_n^2 e_n}_{=:c_n}$$

because

$$\sum_{n=1}^{\infty} \alpha_n^2 e_n \leq \underbrace{\sum_{n=1}^{\infty} \alpha_n^2}_{<\infty} \underbrace{\sum_{n=1}^{\infty} e_n}_{=\sum_{n=1}^{\infty} A + B\sum_{n=1}^{\infty} z_n \overset{???}{<} \infty}$$

and

$$\sum_{n=1}^{\infty} 2\kappa\alpha_n = \infty$$

We will now show that for an arbitrary positive sequence $(z_n)_{n\in\mathbb{N}}$ that satisfies $z_n \leq (1 - a_n)z_n + c_n$ where

$$\sum_{n=1}^{\infty} a_n = \infty, \quad \sum_{n=1}^{\infty} c_n < \infty,$$

that $\lim_{n\to\infty} z_n = 0$.
We first define some complicated $d_n := z_n + \sum_{k=1}^{n-1} a_k z_k - \sum_{k=1}^{n-1} c_k$ and then show that

$$z_n \geq \ldots \geq z_{n+1}.$$

Using this we get that the sequence is not only monotonely decreasing but also bounded by below because

$$z_n \geq -\sum_{k=1}^{n-1} c_k > -\infty$$

which is finite by definition. Thus combining montonicity and boundedness implies convergence. Finally because $(z_n)_{n\in\mathbb{N}}$ is non negative and we can rewrite the definition of d_n as

$$d_n := z_n + \sum_{k=1}^{n-1} a_k z_k - \sum_{k=1}^{n-1} c_k \iff z_n := d_n - \sum_{k=1}^{n-1} a_k z_k + \sum_{k=1}^{n-1} c_k$$

and get that

$$0 \leq z_n = \underbrace{x_n}_{convergent} + \underbrace{\sum_{k=1}^{n-1} c_k}_{<\infty} - \sum_{k=1}^{n-1} a_k z_k$$

which implies that $\sum_{k=1}^{n-1} a_k z_k < \infty$ has to be bounded. Because $\sum_{k=1}^{n-1} a_k = \infty$ it follows that $\liminf_{n\to\infty} z_n = 0$. We can now show that $(z_n)_{n\in\mathbb{N}}$ is a cauchy sequence and then it follows that $\lim_{n\to\infty} z_n = 0$. □

Note that we have hardly any assumptions on the stochastic approximation y_n, but also only get L^2-convergence. The simplest step sizes that satisfy the conditions in this theorem are

$$\alpha_n = \frac{1}{n^p}, \quad p \in (1/2, 1].$$

Note, that here the third condition is the most restrictive and in many cases not satisfied. Lets consider a deterministic approximation, i.e. $\epsilon_n = 0$, then it should hold that

$$G(x_n) \leq A + B(x_n - x^*)$$

In a case that $x^* = 0$ and $G(x) = x^2$(schlechtesd Beispiel. Erfüllt schon das erste nicht!), then

$$x_n^2 \leq A + Bx_n$$

is not satisfied, for negative values for x_n.
We call all vectors $u \in \mathbb{R}^n$

$$G(x) - G(y) \geq \langle u, y - x\rangle$$

subderivatives. This set is defined by

$$\partial G(y) := \{u \in \mathbb{R}^n \mid G(x) - G(y) \geq \langle u, y - x\rangle, x \in \mathbb{R}^n\}$$

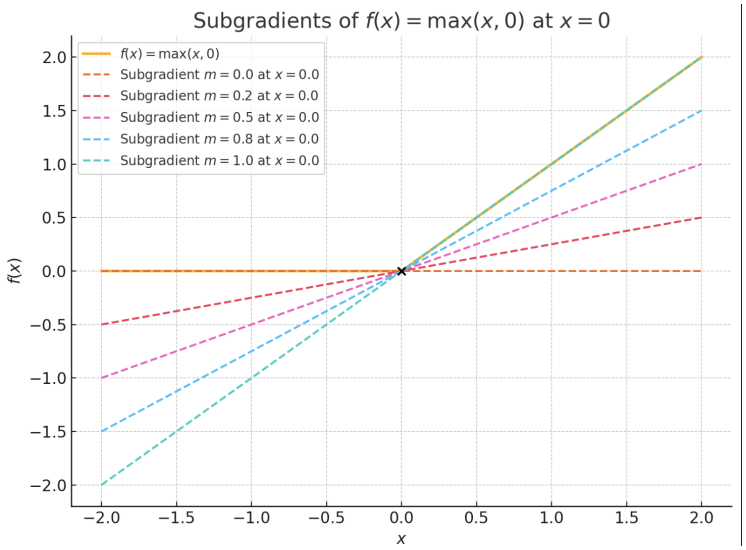


Figure 4.1

The Robins-Monro theorem implies that the root x^* to be subdifferentiable and to contain at least one element:

$$\{\kappa\} \subseteq \partial G(x^*) = \{u \in \mathbb{R}^n \mid G(x) - \underbrace{G(x^*)}_{=0} \geq \langle u, y - x \rangle, x \in \mathbb{R}^n\}.$$

Note that non emptiness of the subdifferential does not imply the condition in the theorem!

Example 1.2

Define $G(x) = x - \mu$ and assume iid centered errors $\epsilon_n := \mu - Z_{n+1}$, where $(Z_n)_{n\in\mathbb{N}}$ is an iid sequence with expectation μ and finite moments, i.e. the conditions of the simplest Robins Monro Algorithm are satisfied, i.e. ϵ_n are centered, conditionally unbiased, $\mathbb{E}[y_n^2] \leq ....$ Choosing α_n such that the conditions are satisfied yields

$$\begin{aligned} x_{n+1} = x_n - \alpha(x_n - \mu + \epsilon_n) &= x_n - \alpha(x_n - \mu + \mu - Z_{n+1}) \\ &= x_n + \alpha(Z_{n+1} - x_n) \end{aligned}$$

If we now apply the memory trick and choose $\alpha_n = \frac{1}{n+1}$, we get that

$$x_{n+1} = x_n + \frac{1}{n+1}(Z_{n+1} - x_n) = \frac{1}{n+1}\sum_{i=1}^{n} Z_{n+1} \overset{a.s.}{\to} \mu = x^*$$

using the LLN. This is due to

$$x_1 = x_0 + 1 \cdot (Z_1 - x_0) = Z_1, x_2 = x_1 + \frac{1}{2}(Z_2 - x_1) = \frac{1}{2}\sum_{i=1}^{2} Z_i, \quad ...$$

This iteration satisfies all condition from the simplest Robins Monro theorem. For example, because $(Z_n)_{n\in\mathbb{N}}$ are the only random variables appearing, it follows that $\mathcal{F}_n = \sigma(Z_1, ..., Z_n)$ and because of iid it holds that

$$\mathbb{E}[\epsilon_n \mid \mathcal{F}_{n+1}] = \mathbb{E}[\mu - Z_n \mid \mathcal{F}_{n+1}] = \mu - \mathbb{E}[Z_n] = 0.$$

We know that $a.s.$ convergence is very slow, but the theorem only has L^2-convergence, which is even worse. Therefore, one good example of this algorithms still performs very slowly.

Example 1.3

One of the original Ideas of Robins Monro was the following. Consider

$$G(X) = \mathbb{E}[g(Y)], \quad Y \sim P_X$$

then

$$\epsilon_n := g(Y_{n+1}) - G(X_n), \quad Y \sim P_{X_n}$$

und jetzt???

We define a generalization of the infinity norm

Definition 1.2

A vector $\vartheta \in \mathbb{R}^d_+ \setminus \{0\}$ we define the weighted maximum norm $\|\cdot\|_\vartheta$ as

$$\|x\|_\vartheta := \max_{i=1,\dots,d} \frac{|x_i|}{\vartheta_i}$$

If we choose $\vartheta = \mathbf{1}$ then we have the usual maximums norm. Now we also generalize the concept of a contraction.

Definition 1.3: Pseudo-Contraction

We call a mapping $F : \mathbb{R}^d \to \mathbb{R}^d$ a weighted maximums norm pseudo contraction if

$$\exists x^* \in \mathbb{R}^d, \vartheta \in \mathbb{R}^d_+ \setminus \{0\}, \lambda \in [0,1) \forall x \in \mathbb{R}^d : \quad \|F(x) - x^*\|_\vartheta \leq \lambda \|x - x^*\|_\vartheta$$

A pseudo contraction wrt. $\|\cdot\|_\vartheta$ is a special case of a contraction wrt. $\|\cdot\|_\vartheta$, i.e. choose $x_2 = x^* = F(x^*)$ as the fixed point.

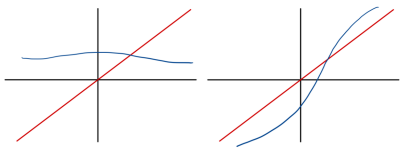


Figure 4.2: Contraction vs. no contraction

A contraction can never grow faster than linear. This is the case in the right plot. For a psuedo contraction it holds that if ϑ_i is chosen huge, then the i^{th} coordinate is not weighted as heavily. If $\vartheta_i \to 0$ then in this coordinate, the pseudo contraction forces it not to grow heavily. We will now look into stochastic fixed point iterations. We motivate this using the Banach fixed point theorem

$$x_{n+1} = F(x_n) = x_n + F(x_n) - x_n \approx x_n + \hat{F}(x_n) + \epsilon_n - x_n,$$

where $\hat{F}$ is an stochastic approximation of F and ϵ_n are errors that do not decrease. This method clearly does not converge, because the errors of the approximation will dominate and thus it will only fluctuate.

Introducing a step size (which will be random), when updating, will alleviate this problem, if it decreases fast enough. Thus we get

$$x_{n+1} \approx x_n + \alpha_n(\hat{F}(x_n) + \epsilon_n - x_n) = (1-\alpha_n)x_n + \alpha_n(\hat{F}(x_n) + \epsilon_n).$$

Thus we define the mutlidimensional scheme for all $i = 1, ..., d$

$$x_i(n+1) := (1-\alpha_i(n))x_i(n) + \alpha_i(n)(F_i(x(n)) + \epsilon_i(n)).$$

Here we interpret the old value $x_i(n)$ as the best known estimate so far that we choose with probability $(1-\alpha_i(n)))$ and $(F_i(x(n))+\epsilon_i(n))$ as the new estimate that we choose with probability $\alpha_i(n)$. The decreasing step size can be interpreted as, increase in the confidence of the best guess so far.

Definition 1.4

An multivariate fixed point iteration algorithm is called totally synchronous if

$$\forall n \in \mathbb{N}: \quad \alpha_1(n) = ... = \alpha_d(n).$$

Else it is called asynchronous. If in each step there is in each step only one $\alpha_i(n) = 1$ then it is called totally asynchronous.

If the above iteration scheme is updated totally synchronous with $\alpha_i(n) = 1$ for all $i = 1, ..., d$ and $n \in \mathbb{N}$ we just have the Banach fixed point iteration scheme.

Theorem 1.5: Simple stochastic fixed point iteration on $\mathbb{R}$

Let $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{N}}, \mathbb{P})$ be a filtered probability space on which all following random variables are defined. Further, let

- $F : \mathbb{R} \to \mathbb{R}$ be a contraction
- ϵ_n be $\mathcal{F}_{n+1}$ measurable for all $n \in \mathbb{N}$ and

$$\forall n \in \mathbb{N}: \quad \mathbb{E}[\epsilon_n \mid \mathcal{F}_n] = 0 \quad \text{and} \quad \mathbb{E}[\epsilon_n^2 \mid \mathcal{F}_n] \leq C.$$

- The random step size $\alpha_n \in [0, 1]$ be $\mathcal{F}_n$ measurable with

$$\sum_{k=1}^{\infty} \alpha_k = \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty \quad a.s.$$

Then the stochastic process $(x_n)_{n\in\mathbb{N}}$ where $\mathcal{F}_0$-measurable (for some initial condition) and

$$x_{n+1} := x_n + \alpha_n(F(x_n) + \epsilon_n - x_n) \overset{a.s.}{\to} x^* \overset{unique}{=} F(x^*).$$

Proof. Skipped, as assumptions are too strong. □

Now we have a little less restrictive assumptions and do it multidimensional.

Theorem 1.6: Stochastic fixed point iteration for contractions on $\mathbb{R}^d$

Let $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{N}}, \mathbb{P})$ be a filtered probability space on which all following random variables are defined. Further, we define the sequence $\forall i = 1, ..., d$

$$x_i(n+1) := (1-\alpha_i(n))x_i(n) + \alpha_i(F_i^n(x(n)) + \epsilon_i(n) + b_i(n)), \quad n \in \mathbb{N}$$

where we have the following assumptions

- All $F^n : \mathbb{R}^d \to \mathbb{R}^d$ are pseudo contractions for the same norm $\|\cdot\|_\vartheta$ with the same $\lambda \in [0, 1)$ and x^*, i.e.
$$\forall n \in \mathbb{N} \forall x \in \mathbb{R}^d : \quad \|F^n(x) - x^*\|_\vartheta \leq \lambda \|x - x^*\|_\vartheta$$
- The random step sizes $((\alpha_i(n))_{i=1,...,n})_{n\in\mathbb{N}}$ are $\mathcal{F}_n$-adapted and satisfy
$$\forall i = 1, ..., d : \quad \sum_{n=1}^{\infty} \alpha_i(n) = \infty \quad \text{and} \quad \sum_{n=1}^{\infty} \alpha_i(n)^2 < \infty \quad a.s.$$
- The errors $\epsilon_i(n)$ satisfy the following for every coordinate
  - $\epsilon_i(n)$ is $\mathcal{F}_{n+1}$-measurable
  - $\mathbb{E}[\epsilon_i(n) \mid \mathcal{F}_n] = 0$
  - The conditional second moment satisfies
$$\forall i = 1, ..., d \exists A, B \geq 0 : \quad \mathbb{E}[\epsilon_i(n)^2 \mid \mathcal{F}_n] \leq A + B\|x(n)\|_\vartheta^2$$
- There exists a sequence of random variables $(\omega_n)_{n\in\mathbb{N}}$ that satisfies
  - $\lim_{n\to\infty} \omega_n = 0$
  - $\forall i = 1, ..., d : |b_i(n)| \leq \omega_n \cdot (1 + ||x(n)||_\vartheta)$, $n \in \mathbb{N}$.

Then it follows that $\lim_{n\to\infty} x(n) = x^*$ $a.s.$

We will only prove this for the following conditions

- $d = 1$
- $b \equiv 0$
- $\forall n \in \mathbb{N} : F^n \equiv F$
- $B = 0$

2 Proof of the general stochastic fixed point iteration

To keep it even simpler we set $F(x) = \lambda x$, $\lambda \in (0,1)$. Then $x^* = 0$ and we can simplify the recursion even further by

$$\begin{aligned} x_{n+1} &= (1-\alpha_n)x_n + \alpha_n(\lambda x_n + \epsilon_n) \\ &= (1-\alpha_n+\alpha_n\lambda)x_n + \alpha_n\epsilon_n \\ &= (1+(\lambda-1)\alpha_n)x_n + \alpha_n\epsilon_n \\ &= (1-(1-\lambda)\alpha_n)x_n + \frac{1-\lambda}{1-\lambda}\alpha_n\epsilon_n \\ &= (1-\bar{\alpha_n})x_n + \bar{\alpha_n}\bar{\epsilon_n}, \end{aligned}$$

where $\bar{\alpha_n} := (1-\lambda)\alpha_n$ and $\bar{\epsilon_n} := \frac{\epsilon_n}{1-\lambda}$.

We will first proove the boundedness and then the proof of convergence will follow very easily.

2.1 Proof of boundedness

First the idea:

1. In order to show boundedness of the sequence $x(n)$ we first show that the contraction property of F_i^n implies linear growth. Using this property in the proof will be easier.

2. Rescaling: We show that the sequence $(\tilde{x}_i(n) := \frac{x_i(n)}{\vartheta_i})_{n\in\mathbb{N}}$ solves the same recursion as $(x(n))_{n\in\mathbb{N}}$ but only wrt.

$$\tilde{F}_i^n(x) := \frac{F_i^n(\vartheta_i x)}{\vartheta_i}, \quad \tilde{\epsilon}_i(n) := \frac{\epsilon_i(n)}{\vartheta_i}$$

and the $\|\cdot\|_\infty$. Therefore we can assume wlog $\vartheta = \mathbf{1}$.

3. $\exists (G_n)_{n\in\mathbb{N}}$ and $\exists \epsilon > 0$ in a specific way such that

$$\|x(n)\|_\infty \le (1+\epsilon)G_n, \text{ and } \|F^n(x(n))\|_\infty + \omega_n(\|x(n)\|_\infty + 1) \le G_n$$

where G_n is a **non**-decreasing sequence, i.e. it either stays constant, if at time $n \in \mathbb{N}$ it holds that $\|x(n)\|_\infty \le (1+\epsilon)G_n$ and else it increases, i.e. choose $\kappa(n)$ such that

$$(1+\epsilon)^{\kappa(n)-1}G_{n_0} < \|x(n+1)\|_\infty \le (1+\epsilon)^{\kappa(n)}G_{n_0} =: G_{n+1}.$$

Intuition für dei Bedinung von F???

4. Show that $(G_n)_{n\in\mathbb{N}}$ stops growing.

a) Rescale the errors such that

$$\mathbb{E}[\tilde{\epsilon}_i^2(n) \mid \mathcal{F}_n] \le \underbrace{\frac{A}{G_n}}_{\le A} + B(1+\epsilon)^2$$

b) Apply Lemma of the convergence of the cumulative biased convexcombination of errors sequence:

$$\tilde{W}_i(n+1:n_0) = (1-\alpha_i(n))W_i(n:n_0) + \alpha_i(n)\tilde{\epsilon}_i(n), \quad \tilde{W}_i(n_0:n_0) = 0.$$

is bounded if started in n_0 large enough. This says, that if we start the cumulative convexcombination of the errors for n_0 large enough, then the entire convexcombination of all errors for $n \ge n_0$ to infinity will be bounded. This statement uses that $\lim_{n\to\infty} \tilde{W}_i(n:0) = 0$, as all conditions of Lemma 2.1 are satisfied.

c) Assume $(x(n))_{n\in\mathbb{N}}$ is unbounded. Then by construction of the G_n it follows that $G_n \to \infty$ and $\|x(n)\|_\infty \le G_n$ holds infinitely often. Now choose $\forall \xi > 0$ a $n_0 \in \mathbb{N}$ large enough such that

- $\|x(n_0)\|_\infty \leq G_{n_0}$
- $\forall n \geq n_0 : \|\tilde{W}(n:n_0)\|_\infty \leq \xi$
- $\forall n \geq n_0 \; \exists \omega^* : \omega_n \leq \omega^*$

Then it follows that

$$\forall n \geq n_0 \text{ it holds that } G_n = G_{n_0} \quad \text{and} \quad \forall n \geq n_0, \forall \xi > 0 : \quad |x_i(n)| \leq G_{n_0}(1+\xi).$$

The second condition follows by induction. Because $\xi > 0$ was chosen arbitrary it follows by the construction of the G_n that the first case, i.e. $G_n = G_{n_0}$ for all $n \geq n_0$, always holds, i.e. the second condition. This now implies boundedness of $(x(n))_{n\in\mathbb{N}}$ which is a contradiction! Thus the opposite holds!

Theorem 2.1

Let $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{N}}, \mathbb{P})$ be a filtered probability space on which all following random variables are defined. Assume that all assumptions from Theorem 1 hold and let $x(n)$ be generated from that sequence. Then if we firhter assume that the following condition, that is called linear growth condition is met, i.e.

$$\exists \lambda \in (0,1), D \in \mathbb{R} : \quad \|F^n(x)\|_\vartheta \leq \lambda \|x(n)\|_\vartheta + D$$

then $(x(n))_{n\in\mathbb{N}}$ is bounded $a.s.$, i.e.

$$\forall \omega \in \Omega \exists (c_n(\omega))_{n\in\mathbb{N}} \forall n \in \mathbb{N} : x(n)(\omega) \leq c_n(\omega),$$

i.e. we have for every realization a bound.

This theorem will be prooven in the very end of this section. ???

Theorem 2.2: Robins-Siegmund Theorem

Let $(\Omega, \mathcal{F}, (\mathcal{F}_n)_{n\in\mathbb{N}}, \mathbb{P})$ be a filtered probability space on which the sequences $(Z_n)_{n\in\mathbb{N}}$, $(A_n)_{n\in\mathbb{N}}$, $(B_n)_{n\in\mathbb{N}}$ and $(C_n)_{n\in\mathbb{N}}$ are defined that are non negative stochastic processes that are adapted and that satisfy

$$\sum_{k=1}^{\infty} A_k < \infty \quad \text{and} \quad \sum_{k=1}^{\infty} B_k < \infty \quad a.s.$$

and let

$$\mathbb{E}[Z_{n+1} \mid \mathcal{F}_n] \leq Z_n(1+A_n) + B_n - C_n, \quad n \in \mathbb{N}.$$

Then it holds that

- The limit
$$\lim_{n\to\infty} Z_n =: Z_\infty < \infty \quad a.s.$$
exists.
- It holds that
$$\sum_{k=0}^{\infty} C_k < \infty \quad a.s.$$

Proof. For the first part, we will apply Doobs Convergence theorem. We will show: A supermartingal $(X_n)_{n\in\mathbb{N}}$ that satisfies

$$\sup_{t\in\mathbb{N}} \mathbb{E}[X_t^-] < \infty, \quad \text{where } X_t^- := -\min\{X_t, 0\},$$

the limit $\lim_{t\to\infty} X_t =: X$ exists *a.s.* with $\mathbb{E}[X] < \infty$.
We will now construct this supermartingal. Define

$$\hat{Z}_n := \frac{Z_n}{\prod_{k=0}^{n-1}(1+A_k)}, \quad \hat{B}_n := \frac{B_n}{\prod_{k=0}^{n-1}(1+A_k)}, \quad \hat{C}_n := \frac{C_n}{\prod_{k=0}^{n-1}(1+A_k)}.$$

and

$$M_n := \hat{Z}_{n+1} - \sum_{k=0}^{n-1}(\hat{B}_k - \hat{C}_k).$$

Then we can show that $(M_n)_{n\in\mathbb{N}}$ is a supermatringal. Here we use that for $k \le n$ it holds that $\hat{C}_k$ and $\hat{B}_k$ are $\mathcal{F}_n$ measurable, because they are $\mathcal{F}_k$ measurable and $\mathcal{F}_0 \subseteq ... \subseteq \mathcal{F}_n$. This yieds

$$\begin{aligned}
\mathbb{E}[M_{n+1} \mid \mathcal{F}_n] &= \mathbb{E}[\hat{Z}_{n+1} \mid \mathcal{F}_n] - \sum_{k=0}^{n}(\mathbb{E}[\hat{B}_k \mid \mathcal{F}_n] - \mathbb{E}[\hat{C}_k \mid \mathcal{F}_n]) \\
&= \mathbb{E}[\frac{Z_{n+1}}{\prod_{k=0}^{n-1}(1+A_k)} \mid \mathcal{F}_n] - \sum_{k=0}^{n}(\mathbb{E}[\hat{B}_k \mid \mathcal{F}_n] - \mathbb{E}[\hat{C}_k \mid \mathcal{F}_n]) \\
&= \frac{1}{\prod_{k=0}^{n-1}(1+A_k)}\mathbb{E}[Z_{n+1} \mid \mathcal{F}_n] - \sum_{k=0}^{n}(\hat{B}_k - \hat{C}_k) \\
&\overset{VSS}{\le} \frac{1}{\prod_{k=0}^{n-1}(1+A_k)}(Z_n(1+A_n) + B_n - C_n) - \sum_{k=0}^{n}(\hat{B}_k - \hat{C}_k) \\
&\overset{Def}{\le} (\hat{Z}_n + \hat{B}_n - \hat{C}_n) - \sum_{k=0}^{n}(\hat{B}_k - \hat{C}_k) \\
&= \hat{Z}_n - \sum_{k=0}^{n-1}(\hat{B}_k - \hat{C}_k) = M_n
\end{aligned}$$

Thus $(M_n)_{n\in\mathbb{N}}$ is a supermartingal. Now in order to apply the Doops theorem, we need to verify, that

$$\sup_{t\in\mathbb{N}} \mathbb{E}[M_t^-] < \infty, \quad \text{where } M_t^- := -\min\{M_t, 0\}.$$

We will first show this locally. For this we define

$$\forall \epsilon > 0: \quad \tau_\epsilon := \inf\{n \in \mathbb{N} \mid \sum_{k=0}^{n} \hat{B}_k > \epsilon\}$$

then τ_ϵ is a stopping time wrt. the filtration $(\mathcal{F}_n)_{n\in\mathbb{N}}$ (???).
$(M_{n\wedge\tau_\epsilon})_{n\in\mathbb{N}}$ is still a supermartingal and is uniformly bounded ($\exists const \forall n \in \mathbb{N}$: $|x_n| \le M$), i.e. $\forall n \in \mathbb{N}$

$$M_{n\wedge\tau_\epsilon} = \hat{Z}_{n\wedge\tau_\epsilon} - \sum_{k=0}^{(n\wedge\tau_\epsilon)-1} \hat{B}_k + \sum_{k=0}^{(n\wedge\tau_\epsilon)-1} \hat{C}_k \ge - \sum_{k=0}^{(n\wedge\tau_\epsilon)-1} \hat{B}_k \overset{\text{construction } \tau_\epsilon}{\ge} -\epsilon.$$

Using this we get directly that

$$\sup_{n\in\mathbb{N}} \mathbb{E}[|M_{n\wedge\tau_\epsilon}|] < \infty.$$

Thus we can use Doops Matringal theorem and get that

$$\forall \epsilon > 0 \exists M_\infty^\epsilon := \lim_{n\to\infty} M_{n\wedge\tau_\epsilon} \text{ finite.} \quad a.s.$$

(Note boundedness is not sufficient for convergence. We need monotonicity, which we get from the supermartingal Property).
To show: $\lim_{n\to\infty} M_n < \infty$ $a.s.$
Let $(\epsilon_k)_{k\in\mathbb{N}}$ be an increasing sequence with $\lim_{k\to\infty} \epsilon_k = \infty$.
Because we showed the above only $a.s.$ for every $\epsilon > 0$ we have that for Null sets $N \subseteq \Omega$, that

$$\forall k \in \mathbb{N} \forall \omega \in \Omega \setminus N : \quad \lim_{n\to\infty} M_{n\wedge\tau_{\epsilon_k}}(\omega) =: M_\infty^{\epsilon_k}(\omega) \text{ exists.}$$

The set

$$\Omega^{\hat{B}} := \{\omega \in \Omega \mid \sum_{k=0}^{\infty} \hat{B}_k(\omega) < \infty\} \subseteq \Omega$$

is no null set due to

$$\begin{aligned}\mathbb{P}\left(\sum_{k=0}^{\infty} \hat{B}_k < \infty\right) &= \mathbb{P}\left(\sum_{k=0}^{\infty} \frac{B_k}{\prod_{i=0}^{k}(1+A_i)} < \infty\right) \\ &\geq \mathbb{P}\left(\sum_{k=0}^{\infty} \frac{B_k}{e^{\sum_{i=0}^{k}(1+A_i)}} < \infty\right)\end{aligned}$$

although we used that $e^{-x} \geq \frac{1}{1+x}$. Finally, using that e^{-x} is monotonically decreasing and $A_i \geq 0$, it follows that

$$\mathbb{P}\left(\sum_{k=0}^{\infty} \frac{B_k}{e^{\sum_{i=0}^{k}(1+A_i)}} < \infty\right) \geq \mathbb{P}\left(\sum_{k=0}^{\infty} \frac{B_k}{e^0} < \infty\right) = \mathbb{P}\left(\sum_{k=0}^{\infty} B_k < \infty\right) \overset{VSS}{>} 0.$$

Thus $\forall \omega \in \Omega^{\hat{B}}$ it holds that if we choose $N \in \mathbb{N}$ such that ϵ_N large enough, then $\omega \in \{\tau_{\epsilon_N} = \infty\}$. We have

$$\forall \omega \in \Omega^{\hat{B}} \forall n \in \mathbb{N} : \quad M_{n\wedge\tau_{\epsilon_N}}(\omega) = M_n(\omega).$$

This leads using

$$\sup_{n\in\mathbb{N}} \mathbb{E}[|M_n|] = \sup_{n\in\mathbb{N}} \mathbb{E}[|M_{n\wedge\tau_{\epsilon_N}}|] < \infty$$

with Doobs convergence theorem to

$$\forall \omega \in \Omega^{\hat{B}} : \quad \lim_{n\to\infty} M_n(\omega) = \lim_{n\to\infty} M_{n\wedge\tau_{\epsilon_N}}(\omega) = M_\infty^{\epsilon_N} < \infty \text{ exists.}$$

Now we can show the two assertion:

- Z_n converges almoust surely:
  Using the fact that $\forall i$ it holds that $\hat{B}_i \leq B_i$ it follows

$$-\infty \overset{VSS}{<} -\sum_{k=0}^{\infty} B_k \leq -\sum_{k=0}^{\infty} \hat{B}_k$$
$$\leq \lim_{n\to\infty} \left(\underbrace{\hat{Z}_n}_{\geq 0} - \sum_{k=0}^{n-1} \hat{B}_k + \underbrace{\sum_{k=0}^{n-1} \hat{C}_k}_{\geq 0} \right) \overset{Def.}{=} \lim_{n\to\infty} M_n \overset{above}{<} \infty \quad a.s.$$

  Now using the non-negativity (???) we get that

$$\lim_{n\to\infty} \hat{Z}_n \text{ and } \sum_{k=0}^{\infty} \hat{C}_k \text{ exist} \quad a.s.$$

  Finally, using that

$$\prod_{k=0}^{n-1}(1+A_k) \leq e^{\sum_{k=0}^{n-1}(1+A_k)} \leq e^{\sum_{k=0}^{\infty}(1+A_k)} \overset{VSS}{<} \infty$$

  it follows that

$$\lim_{n\to\infty} Z_n \overset{Def.}{=} \lim_{n\to\infty} \hat{Z}_n \prod_{k=0}^{n-1}(1+A_k) < \infty \quad a.s.$$

- Using the a.s. existence of $\lim_{n\to\infty} Z_n$ leads to

$$\lim_{n\to\infty} \sum_{k=0}^{n} C_k \overset{Def.}{=} \lim_{n\to\infty} \sum_{k=0}^{n} \hat{C}_k \prod_{j=0}^{k}(1+A_j) \overset{\text{non neg.}}{\leq} \lim_{n\to\infty} \sum_{k=0}^{n} \hat{C}_k \prod_{j=0}^{\infty}(1+A_j)$$
$$\leq \lim_{n\to\infty} \sum_{k=0}^{n} \hat{C}_k e^{\sum_{j=0}^{\infty}(1+A_j)} < \infty \quad a.s.$$

□

Corollary 2.3

Suppose that $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_n)_{n\in\mathbb{N}})$ is a filtered probability space on which the non negative adapted stochastic process $(Z_n)_{n\in\mathbb{N}}$, $(a_n)_{n\in\mathbb{N}}$ $(b_n)_{n\in\mathbb{N}}$ $(c_n)_{n\in\mathbb{N}}$ are defined. If

$$\forall n \in \mathbb{N}: \quad \mathbb{E}[Z_{n+1} \mid \mathcal{F}_n] \leq (1 - a_n + b_n)Z_n + c_n$$

and

$$\sum_{k=1}^{\infty} a_k = \infty, \quad \sum_{k=1}^{\infty} b_k < \infty, \quad \sum_{k=1}^{\infty} c_k < \infty, \quad a.s.$$

then it follows that $\lim_{n\to\infty} Z_n = 0$ $a.s.$.

Proof. We define $A_n := b_n$, $B_n := c_n$ and $C_n := Z_n a_n$. With the assumption it follows that

$$\forall n \in \mathbb{N}: \quad \mathbb{E}[Z_{n+1} \mid \mathcal{F}_n] \leq (1 - A_n + b_n)Z_n + B_n - C_n$$

and

$$\sum_{k=1}^{\infty} A_k < \infty, \quad \sum_{k=1}^{\infty} B_k < \infty, \quad a.s.$$

Using the Robins Siegmund theorem it follows that

$$\lim_{n\to\infty} Z_n \text{ and } \sum_{k=0}^{\infty} Z_k a_k \text{ exist} \quad a.s.$$

Because by assumption $\sum_{k=0}^{\infty} a_k = \infty$ it follows that

$$\liminf_{n\to\infty} Z_n = 0 \quad a.s.,$$

but because the limit exists, we immediately get that

$$\lim_{n\to\infty} Z_n = 0 \quad a.s.$$

□

Lemma 2.4

Suppose that $(\Omega, \mathcal{F}, \mathbb{P}, (\mathcal{F}_n)_{n\in\mathbb{N}})$ is a filtered probability space on which all the following random variabels are defined. Assume that

- ϵ_n are $\mathcal{F}_{n+1}$ measurable with $\mathbb{E}[\epsilon_n \mid \mathcal{F}_n] = 0$ and

  $\exists$ pathwise bounded adapted process $(D_n)_{n\in\mathbb{N}}$, i.e.$\forall n \in \mathbb{N} \forall \omega \in \Omega \setminus N : D_n(\omega) \leq \sup_{n\in\mathbb{N}} D_n(\omega) < \infty$,

  that satisfies

  $$\forall n \in \mathbb{N} : \mathbb{E}[\epsilon_n^2 \mid \mathcal{F}_n] \leq D_n.$$

- $\alpha_n \in [0, 1]$ are $\mathcal{F}_n$ measurable random variables satisfying

  $$\sum_{k=1}^{\infty} a_k = \infty, \quad \sum_{k=1}^{\infty} a_k^2 < \infty \quad a.s.$$

Then for the stochastic process $(W_n)_{n\in\mathbb{N}}$ where W_0 is $\mathcal{F}_0$-measurable for some initial condition and satisfying

$$W_{n+1} = (1 - \alpha_n)W_n + \alpha_n \epsilon_n$$

holds that

$$\lim_{n\to\infty} W_n = 0 \quad a.s.$$

Proof. We will apply the corollary above to $(W_n^2)_{n\in\mathbb{N}}$.

$$\begin{aligned}\mathbb{E}[W_{n+1}^2 \mid \mathcal{F}_n] &= \mathbb{E}[(1-\alpha_n)^2 W_n^2 + \alpha_n^2\epsilon_n^2 + 2\alpha_n(1-\alpha_n)W_n\epsilon_n \mid \mathcal{F}_n] \\ &= (1-\alpha_n)^2 W_n^2 + \alpha_n^2\mathbb{E}[\epsilon_n^2 \mid \mathcal{F}_n] + 2\alpha_n(1-\alpha_n)W_n \underbrace{\mathbb{E}[\epsilon_n \mid \mathcal{F}_n]}_{=0} \\ &= (1 - \underbrace{2\alpha_n}_{=:a_n} + \underbrace{\alpha_n^2}_{=:b_n})W_n^2 + \alpha_n^2\mathbb{E}[\epsilon_n^2 \mid \mathcal{F}_n] \\ &\leq (1 - a_n + b_n)W_n^2 + \underbrace{\alpha_n^2 D_n}_{=:c_n}\end{aligned}$$

Thus we get with the corollary above that

$$\lim_{n\to\infty} W_n^2 = 0 \quad \Rightarrow \quad \lim_{n\to\infty} W_n = 0.$$

□

Note that ϵ_n is $\mathcal{F}_{n+1}$ measurable and α_n only $\mathcal{F}_n$ as the error accurs between step n to $n+1$, if ϵ_n was $\mathcal{F}_{n+1}$ measureable then

$$\mathbb{E}[\epsilon_n \mid \mathcal{F}_n] = \epsilon_n \stackrel{!}{=} 0$$

which would be errors that are zero. ??? **Main Proof of the Robins Monro Theorem:**

Proof. W.l.o.g. we can get rid of the weighted norms, i.e. $\vartheta = \mathbf{1}$, because if we transform the space

$$\tilde{x_i} := \frac{x_i}{\vartheta_i}, \quad \tilde{F_i^n}(x) := \frac{F^n(\vartheta x)}{\vartheta_i}, \quad b_i := \frac{b_i}{\vartheta_i}, \quad \text{and} \quad \tilde{\epsilon_i}(n) := \frac{\epsilon_i(n)}{\vartheta_i}.$$

It follows that $\tilde{x}(n)$ solves the same recursion as $x(n)$ only wrt. $\tilde{F}$ and $\tilde{\epsilon}$, i.e.

$$\|\tilde{F}^n(\tilde{x}(n))\|_\infty = \|\frac{F^n(\vartheta_i \frac{x_i}{\vartheta_i})}{\vartheta_i}\|_\infty = \|F^n(x(n))\|_{\vartheta_i} \leq \lambda\|x(n)\|_{\vartheta_i} + D = \lambda\|\tilde{x}(n)\|_\infty + D.$$

Therefore we can now work with the $\|\cdot\|_\infty$ norm instead of the $\|\cdot\|_\vartheta$ norm.!
We will now find a very clean bound, such that the proof of convergence will be very easily. If we would find a not so nice upper bound the second part, i.e. the proof of convergence would be harder.
To show:

- There exists an adapted process $(G_n)_{n\in\mathbb{N}}$ such that

$$\exists\epsilon > 0 : \quad \|x(n)\|_\infty \leq (1+\epsilon)G_n$$

and

$$\|F^n(x(n))\|_\infty + \omega_n(\|x(n)\|_\infty + 1) \leq G_n.$$

Let λ and D be the constants from the Robins Monro theorem. Choose a constant $G \geq 1$ such that $\lambda G + D < G$, for example $G := \max\{\frac{D+1}{1-\lambda}, 1\}$. Next, choose $\eta \in [0,1)$ such that $\lambda G + D = \eta G$. Because $D > 0$ it follows that

$$\lambda G < \lambda G + D = \eta G \quad \Longleftrightarrow \quad \lambda < \eta.$$

Finally, choose $\epsilon > 0$ such that $(1+\epsilon)\eta = 1$.
Next, we define the random recursive sequence

$$G_{n+1} := \begin{cases} G_n & , \|x(n+1)\|_\infty \leq (1+\epsilon)G_n \\ G_0(1+\epsilon)^{\kappa(n)} & , else \end{cases}, \quad \text{from the choice of } \kappa(n)$$

where $G_0 := \max\{\|x(0)\|_\infty, G\}$ and choose $\kappa(n) \in \mathbb{R}$ such that

$$G_0(1+\epsilon)^{\kappa(n)-1} < \|x(n+1)\|_\infty \leq G_0(1+\epsilon)^{\kappa(n)}.$$

This is possible because a norm is non negative and with the mapping $const.^x$ we hit all non-negative values for $x \in \mathbb{R}$.
Further, because the norm $\|\cdot\|_\infty$ is continuous if follows that $\|x(n)\|_\infty$ is $\mathcal{F}_n$ measurable. Becuase G_n is only determind by indikators, norms and constants this is sequence is adapted to the given Filtration. The sequence $(G_n)_{n\in\mathbb{N}}$ is non-necreasing, because it either stays the same or it decreases $G_0(1+\epsilon)^{\kappa(n+1)-1} < G_0(1+\epsilon)^{\kappa(n+1)}$.
Further it satisfies by construction that

$$\|x(n)\|_\infty \leq \begin{cases} (1+\epsilon)G_n \\ G_n = G_0(1+\epsilon)^{\kappa(n)} \quad , G_{n-1} < G_n \end{cases}$$

Next, because $\lambda < \eta$ it follows that

$$\lambda\epsilon + \eta < \eta\epsilon + \eta = (1+\epsilon)\eta \overset{construction}{=} 1.$$

Thus there exists a $\omega^* > 0$ such that

$$\lambda\epsilon + \eta + \omega^*(2+\epsilon) \leq 1.$$

By assumption of the theorem the sequence $(\omega_n)_{n\in\mathbb{N}}$ converges to zero almoust surely, therefore

$$\exists n^* \in \mathbb{N} \forall n \geq n^* : \quad \omega_n \leq \omega^*, \quad a.s.$$

We will now prove as part of the existence of the adapted process $(G_n)_{n\in\mathbb{N}}$ that

$$\forall n \geq n^* : \quad \|F^n(x(n))\|_\infty + \omega_n \cdot (\|x(n)\|_\infty + 1) \leq G_n.$$

Now using the linear growth condition we showed holds in the beginning of this section, we get that

$$\begin{aligned} \|F^n(x(n))\|_\infty &\overset{lin.Gr.Cond.}{\leq} \lambda\|x(n)\|_\infty + D \\ &\leq \lambda(1+\epsilon)G_n + D \\ &\overset{Construction}{=} \lambda(1+\epsilon)G_n + (\eta - \lambda)G \\ &\overset{\text{non-decreasing}}{\leq} \lambda(1+\epsilon)G_n + (\eta-\lambda)G_n \\ &= (\lambda + \lambda\epsilon + \eta - \lambda)G_n = (\lambda\epsilon + \eta)G_n \end{aligned}$$

Thus we get for all $n \geq n^*$

$$\begin{aligned} \|F^n(x(n))\|_\infty + \omega_n \cdot (\|x(n)\|_\infty + 1) &\leq (\lambda\epsilon + \eta)G_n + \omega^*((1+\epsilon)G_n + 1) \\ &\leq (\lambda\epsilon + \eta)G_n + \omega^*((1+\epsilon)G_n + G_n) \\ &= (\lambda\epsilon + \eta)G_n + \omega^*((2+\epsilon)G_n) \\ &= \underbrace{(\lambda\epsilon + \eta + \omega^*((2+\epsilon))}_{\leq 1, construction} G_n \\ &\leq G_n \end{aligned}$$

What we did is construct a set of boxes that is non decreasing in each step and bound the sequence $(x_n)_{n\in\mathbb{N}}$. We now have to show that this set of boxes ultimately stop growing! We do this first using the error term, i.e. we show that the convex combination ??? of the adjusted error in time step before to the current adjusted error is bounded. We define the adjusted error by

$$\tilde{\epsilon}_i(n) := \frac{\epsilon_i(n)}{G_n}$$

and the iterative convex combination

$$\tilde{W}_i(n+1:n_0) := (1-\alpha_i(n))\tilde{W}_i(n:n_0) + \alpha_i(n)\tilde{\epsilon}_i(n)$$

started in $\tilde{W}_i(n_0:n_0) = 0$.
To show:

- $$\forall\delta > 0 \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \quad |\tilde{W}_i(n:n_0)| \leq \delta$$

Step 1:
We begin by showing

$$\forall\delta \exists n_0 \in \mathbb{N} \forall n \geq n_0 : \quad |W_i(n:n_0)| \leq \delta, \quad a.s., \quad i = 1, ..., d.$$

Because the random sequence $(\tilde{W}_i(n:0))_{n\in\mathbb{N}}$ is generated by only the random sequence $(\epsilon_i(n))_{n\in\mathbb{N}}$ it follows that we can write

$$\begin{aligned}
\tilde{W}_i(n:0) &= (1-\alpha_i(n-1))\tilde{W}_i(n-1:0) + \alpha_i(n-1)\tilde{\epsilon}_i(n-1) \\
&= (1-\alpha_i(n-1))\tilde{W}_i(n-1:0) + \alpha_i(n-1)\tilde{\epsilon}_i(n-1) + (1-\alpha_i(n-1))\underbrace{\tilde{W}_i(n-1:n-1)}_{=0} \\
&= (1-\alpha_i(n-1))\tilde{W}_i(n-1:0) + \tilde{W}_i(n:n-1) \\
&= (1-\alpha_i(n-1))\Big((1-\alpha_i(n-2))\tilde{W}_i(n-2:0) + \alpha_i(n-2)\tilde{\epsilon}_i(n-2)\Big) + \tilde{W}_i(n:n-1) \\
&= \prod_{i=1}^{2}(1-\alpha_i(n-i))\tilde{W}_i(n-2:0)) + (1-\alpha_i(n-1))\underbrace{\alpha_i(n-2)\tilde{\epsilon}_i(n-2)}_{=(1-\alpha_i(n-2))0+\alpha_i(n-2)\tilde{\epsilon}_i(n-2)=\tilde{W}_i(n-1:n-2)} + \tilde{W}_i(n:n-1) \\
&= \prod_{i=1}^{2}(1-\alpha_i(n-i))\tilde{W}_i(n-2:0)) + (1-\alpha_i(n-1))\tilde{W}_i(n-1:n-2) + \tilde{W}_i(n:n-1) \\
&= \prod_{i=1}^{2}(1-\alpha_i(n-i))\tilde{W}_i(n-2:0)) + (1-\alpha_i(n-1))\tilde{W}_i(n-1:n-2) + 0 + \alpha_i(n-1)\tilde{\epsilon}_i(n-1) \\
&= \prod_{i=1}^{2}(1-\alpha_i(n-i))\tilde{W}_i(n-2:0)) + \tilde{W}_i(n:n-2) \\
&= \prod_{i=1}^{s}(1-\alpha_i(n-s))\tilde{W}_i(s:0)) + \tilde{W}_i(n:s), \quad s \leq n
\end{aligned}$$

Because every $\alpha_i(n) \leq 1$ we have that for all $s \leq n$:

$$\tilde{W}_i(n:0) \leq \tilde{W}_i(s:0)) + \tilde{W}_i(n:s)$$

which is equivalent to

$$|\tilde{W}_i(n:s)| \leq |\tilde{W}_i(s:0))| + |\tilde{W}_i(n:0)|.$$

Step 2:
The sequence $(W(n:0))_{n\in\mathbb{N}}$ is of the form of Lemma 2.1 and because ϵ_n and α_n satisfy their respective conditions from Lemma 2.1 it follows that

$$\lim_{n\to\infty} \tilde{W}_i(n:0) = 0.$$

Thus it follows that for every $\delta > 0$ there exists $n_0 \in \mathbb{N}$ such that it holds that

$$\forall n \geq n_0 : \quad |\tilde{W}_i(n:0)| \leq \frac{\delta}{2}.$$

This now implies that for $n_0 \in \mathbb{N}$ large enough it holds that for $n \geq n_0$:

$$|\tilde{W}_i(n:n_0)| \leq |\tilde{W}_i(n_0:0))| + |\tilde{W}_i(n:0)| \leq \delta.$$

We now want to show that the sequence $(x(n))_{n\in\mathbb{N}}$ is almoust surely unbounded. For this assume that

$$\Omega_+ := \{\omega \in \Omega \mid (x(n)(\omega))_{n\in\mathbb{N}} \text{ is unbounded } \}$$

is no Nullset. By construction we have that

$$\|x(n)\|_\infty = \begin{cases} (1+\epsilon)G_n & , else \\ G_n & , G_{n-1} < G_n \end{cases}$$

which implies that $(G_n)_{n\in\mathbb{N}}$ is unbounded, due to the unboundedness of $x(n)$. Because $(G_n)_{n\in\mathbb{N}}$ is unbounded, we have $G_{n-1} < G_n$ infinitely often, i.e. $\|x(n)\|_\infty \leq G_n$ infinitely often. This implies that

$$\forall \epsilon > 0 \exists n_0 \in \mathbb{N} : \quad \|x(n)\|_\infty \leq G_{n_0} \tag{4.1}$$

and for all $n \geq n_0$ it holds that

$$\|\tilde{W}(n:n_0)\|_\infty \leq \epsilon. \tag{4.2}$$

We can choose n_0 even larger if necessary such that

$$\exists \omega^* > 0 \forall n \geq n_0 : \quad \omega_n \leq \omega^*. \tag{4.3}$$

We will now show via induction the following two results: First $\forall n \geq n_0$

$$G_n = G_{n_0}$$

and that

$$\begin{aligned} -G_{n_0}(1+\epsilon) &\leq -G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0} \\ &\leq x_i(n) \\ &\leq G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0} \\ &\leq G_{n_0}(1+\epsilon) \end{aligned}$$

But this will imply that $(x(n))_{n\in\mathbb{N}}$ is almoust surely bounded, i.e. a contraction! Therefore, it is bounded. What is left to show is that the equations above by induction.

Induction Start:
We have $n = n_0$ thus

$$\begin{aligned}
-G_{n_0}(1+\epsilon) &\leq -G_{n_0} + \underbrace{\tilde{W}_i(n:n_0)}_{=0} G_{n_0} \\
&\leq -\|x(n)\|_\infty \\
&\leq x_i(n) \\
&\leq G_{n_0} + \underbrace{\tilde{W}_i(n:n_0)}_{=0} G_{n_0} \\
&\leq G_{n_0}(1+\epsilon)
\end{aligned}$$

and $G_n = G_{n_0}$ clearly. Induction Step:

$$\begin{aligned}
x_i(n+1) &\overset{Def.}{=} (1-\alpha_i(n))x_i(n) + \alpha_i(n)F_i^n(x(n)) + \alpha_i(n)\epsilon_i(n) + \alpha_i b_i(n) \\
&\overset{IV}{\leq} (1-\alpha_i(n))\left(G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0}\right) + \alpha_i(n)F_i^n(x(n)) + \alpha_i(n)\epsilon_i(n) + \alpha_i b_i(n) \\
&\overset{constr.}{=} (1-\alpha_i(n))\left(G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0}\right) + \alpha_i(n)F_i^n(x(n)) + \alpha_i(n)\tilde{\epsilon}_i(n)G_{n_0} + \alpha_i b_i(n) \\
&\overset{constr.}{=} (1-\alpha_i(n))\left(G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0}\right) + \alpha_i(n)F_i^n(x(n)) + \alpha_i(n)\tilde{\epsilon}_i(n)G_{n_0} + \alpha_i b_i(n) \\
&\leq (1-\alpha_i(n))\left(G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0}\right) + \alpha_i(n)\tilde{\epsilon}_i(n)G_{n_0} + \alpha_i(n)F_i^n(x(n)) + \alpha_i|b_i(n)| \\
&\overset{assumption}{\leq} (1-\alpha_i(n))\left(G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0}\right) + \alpha_i(n)\tilde{\epsilon}_i(n)G_{n_0} + \alpha_i(n)F_i^n(x(n)) + \alpha_i\omega_n(\|x(n)\|_\infty + 1) \\
&\overset{above}{\leq} (1-\alpha_i(n))\left(G_{n_0} + \tilde{W}_i(n:n_0)G_{n_0}\right) + \alpha_i(n)\tilde{\epsilon}_i(n)G_{n_0} + \alpha_i G_{n_0} \\
&= (1-\alpha_i(n))\tilde{W}_i(n:n_0)G_{n_0} + \alpha_i(n)\tilde{\epsilon}_i(n)G_{n_0} + G_{n_0} \\
&= \tilde{W}_i(n+1:n_0)G_{n_0} + G_{n_0}
\end{aligned}$$

Due to symmetry is also holds that $\tilde{W}_i(n+1:n_0)G_{n_0} - G_{n_0} \leq x_i(n+1)$. Finally, using the above we have that for all $\epsilon > 0$

$$|x_i(n+1) \leq G_{n_0}(1+\epsilon)|.$$

Because $\epsilon > 0$ was chosen arbitrary and also $\|x(n+1)\|_\infty \leq (1+\epsilon)G_{n+1}$ holds, it follows that $G_{n+1} = G_{n_0}$.

2.2 Proof of convergence

First the Idea:
1. Rescale the Coordinate System, i.e. w.l.o.g. let $F^n(x^*) = x^* = 0$.
2. In the first part we showed that there is an a.s. upper bound $D_0 > 0$ with

$$\forall n \in \mathbb{N}: \quad \|x(n)\|_\infty \leq D_0.$$

When considering the sequence for $n \geq n_0$

$$W_i(n+1:n_0) := (1-\alpha_i(n))W_i(n:n_0) + \alpha_i(n)\epsilon_i(n), \quad W_i(n_0:n_0) = 0$$

then we do not need to rescale it this time, because it already satisfies the assumption of Lemma 2.1:

$$\forall n \in \mathbb{N}: \quad \mathbb{E}[\epsilon_n^2 \mid \mathcal{F}_n] \leq A + B\|x(n)\|_\infty^2 \leq A + BD_k^2 < \infty, \quad a.s.$$

Thus $(W(n:n_0))_{n\in\mathbb{N}}$ converges to zero for all $n_0 \in \mathbb{N}$.
3. We define a decreasing sequence $(D_k)_{k\in\mathbb{N}}$ and show via induction, i.e. for all $k \in \mathbb{N}$ that

$$\exists n_k \in \mathbb{N} \forall n \geq n_k: \quad \|x(n)\|_\infty \leq D_k \quad a.s.$$

Proof:
Because the shifted recursion $\tilde{F}^n(\cdot) := F(\cdot + x^*) - x^*$ is a contraction:

$$\|\tilde{F}^n(x) - 0\|_\infty = \|F(x + x^*) - x^*\|_\infty \leq \lambda\|x + x^* - x^*\|_\infty = \lambda\|x - 0\|_\infty.$$

Suppose that the recursion $(\tilde{x}_n)_{n\in\mathbb{N}}$ generated by $\tilde{F}^n$ is proven to converge to 0. Then $\tilde{x}_n + x^* \to x^*$ implies that $x_n \to x^*$, because

$$\begin{aligned}
\tilde{x}_i(n+1) + x^* &= (1 - \alpha_i(n))(\tilde{x}_i(n) + x^*) + \alpha_i(n)\left(\tilde{F}_i^n(x(n)) + x^* + \epsilon_i(n) + b_i(n)\right) \\
&= (1 - \alpha_i(n))(\tilde{x}_i(n) + x^*) + \alpha_i(n)\left(F_i^n(x(n)) - x^* + x^* + \epsilon_i(n) + b_i(n)\right) \\
&= x_i(n+1) + x^*.
\end{aligned}$$

Thus we can assume without loss of generality that

$$F^n(x^*) = x^* = 0.$$

Because $(x(n))_{n\in\mathbb{N}}$ is almoust surely bounded, there exists an almoust surely upper bound

$$\|x(n)\|_\infty \leq D_0 \quad a.s.$$

The sequence

$$W_i(n+1:n_0) = (1 - \alpha_i(n))W_i(n:n_0) + \alpha_i(n)\epsilon_i(n), \quad W_i(n_0:n_0) = 0$$

By assumption it holds that

$$\mathbb{E}[\epsilon_i^2(n) \mid \mathcal{F}_n] \leq A + B\|x(n)\|_\infty \leq A + BD_0 < \infty.$$

Thus we can now apply Lemma 2.1 and get that the sequence $(W(n:n_0))_{n\in\mathbb{N}}$ converges to 0 for all $n_0 \in \mathbb{N}$.
Outer Induction: To show: There exists a sequence $(n_k)_{k\in\mathbb{N}}$ such that for all $n \geq n_k$ it holds that $\|x(n)\|_\infty \leq D_k$.
Induction Start: We already know that $\|x(n)\|_\infty \leq D_0$ a.s., i.e. $n_0 = 0$.
Induction Voraussetzung: We assume for all $n \geq n_k$ that $\|x(n)\|_\infty \leq D_k$.
Induction Step: We begin with the bias term. Using the assumption we get that

$$\forall n \geq n_k: \quad |b_i(n)| \leq \omega_n(\|x(n)\|_\infty + 1) \leq \omega_k(D_k + 1),$$

because ω_k converges to zero, it follows that the bias disappears. Thus

$$\forall \epsilon > 0 \exists \tau_k \geq n_k \forall n \geq \tau_k: \quad \|b(n)\|_\infty \leq \epsilon D_k$$

Using the λ from the theorem (i.e. the contraction factor) we define the stochastic process $(Y_n)_{n\geq\tau_k}$ by

$$Y_i(n+1) := (1 - \alpha_i(n))Y_i(n) + \alpha_i(n)(\lambda + \epsilon)D_k, \quad Y_i(\tau_k) = D_k.$$

We can show that this process converges to $(\lambda+\epsilon)D_k$, due to the following argument. Define $V_i(n) := Y_i(n) - (\lambda+\epsilon)D_k$ that satisfies

$$\begin{aligned} V_i(n+1) &= Y_i(n+1) - (\lambda+\epsilon)D_k \\ &= (1-\alpha_i(n))Y_i(n) + \alpha_i(n)(\lambda+\epsilon)D_k - (\lambda+\epsilon)D_k \\ &= (1-\alpha_i(n))(Y_i(n) - (\lambda+\epsilon)D_k) \\ &= (1-\alpha_i(n))V_i(n) \end{aligned}$$

which converges to zero due to

$$V_i(n+1) = \prod_{j=1}^{n+1}(1-\alpha_i(j))V_i(0)$$

and because by assumption $\sum_{j=1}^{\infty}\alpha_i(j)^2 < \infty$ it follows that $(\alpha_i(j))_{j\in\mathbb{N}}$ is a zero sequence therefore, finitely many values are greater than one and infinite many are smaller, thus

$$\prod_{j=1}^{n}(1-\alpha_i(j)) \to 0, \quad n\to\infty.$$

In order to show the outer induction, we show via an inner induction that we can sandwich the sequence $(x(n))_{n\in\mathbb{N}}$ as follows **Inner induction:** For fixed k we have to show that $\forall n \geq \tau_k$:

$$-Y_i(n) + W_i(n:\tau_k) \leq x_i(n) \leq Y_i(n) + W_i(n:\tau_k).$$

Induction start: With definition $Y_i(\tau_k) = D_k$ and $W(\tau_k:\tau_k) = 0$ and $|x_i(n)| \leq D_k$ already holds.
Induction Step: Note that we were able to assume w.l.o.g. that $x^* = 0$ and thus

$$\|F^n(x(n))\|_\infty \leq \lambda\|x(n)\|_\infty \leq \lambda D_k.$$

It now holds that

$$\begin{aligned} x_i(n+1) &= (1-\alpha_i(n))x_i(n) + \alpha_i(n)\left(F_i^n(x(n)) + \epsilon_i(n) + b_i(n)\right) \\ &\overset{IV}{\leq} (1-\alpha_i(n))(Y_i(n) + W_i(n:\tau_k)) + \alpha_i(n)\left(F_i^n(x(n)) + \epsilon_i(n) + b_i(n)\right) \\ &\leq (1-\alpha_i(n))(Y_i(n) + W_i(n:\tau_k)) + \alpha_i(n)\left(\lambda D_k + \epsilon_i(n) + \|b(n)\|_\infty\right) \\ &\leq (1-\alpha_i(n))(Y_i(n) + W_i(n:\tau_k)) + \alpha_i(n)\left(\lambda D_k + \epsilon_i(n) + \epsilon D_k\right) \\ &= (1-\alpha_i(n))Y_i(n) + \alpha_i(n)\left(\lambda D_k + \epsilon D_k\right) + (1-\alpha_i(n))W_i(n:\tau_k) + \alpha_i(n)\epsilon_i(n) \\ &= (1-\alpha_i(n))Y_i(n) + \alpha_i(n)\left(\lambda+\epsilon\right)D_k + W_i(n+1:\tau_k) \\ &= Y_i(n+1) + W_i(n+1:\tau_k) \end{aligned}$$

This concludes the inner induction.
We can now finish, with the sandwitch argument. Since $W(n:n_0)$ converges to zero and Y_i converges to $(\lambda+\epsilon)D_k$ it follows that for large $n \geq n_{k+1}$ it holds that

$$\|x(n)\|_\infty \leq (\lambda+\epsilon)D_k + \epsilon D_k = (\lambda+2\epsilon)D_k =: D_{k+1},$$

because due to the fact that $W(n:n_0)$ is a zero sequence $\forall\bar{\epsilon} > 0\ \exists N : |W(n:n_0)| \leq \bar{\epsilon}\ \forall n \geq N$. Here we choose $\bar{\epsilon} = \epsilon D_k$ and $N =: n_{k+1}$.
The outer induction implies that $\limsup_{n\to\infty}\|x(n)\|_\infty \leq D_k$. Because D_k vanishes, it follows that $\lim_{n\to\infty}\|x(n)\|_\infty = 0$.

□

3 Sample based dynamic Programming

In the Section above we showed that for a recursion

$$x_i(n+1) := (1-\alpha_i(n))x_i(n) + \alpha_i(n)(F_i(x(n)) + \epsilon_i(n) + b_i(n))$$

where

$$F : \mathbb{R}^d \to \mathbb{R}^d, \quad x \mapsto (F_i(x))_{i=1,\dots,d} = (\mathbb{E}_i[f(x,Z)])_{i=1,\dots,d}$$

the approximate recursion

$$\begin{aligned} x_i(n+1) &:= (1-\alpha_i(n))x_i(n) + \alpha_i(n)(f(x(n), Z_i(n))) \\ &= (1-\alpha_i(n))x_i(n) + \alpha_i(n)(f(x(n), Z_i(n)) + F_i(x(n)) - F_i(x(n))) \\ &= (1-\alpha_i(n))x_i(n) + \alpha_i(n)(F_i(x(n)) + \underbrace{f(x(n), Z_i(n)) - F_i(x(n))}_{=:\epsilon_i(n)} + \underbrace{0}_{=:b_i(n)}) \end{aligned}$$

where for all $n \in \mathbb{N}$ are distributed according to $Z_i^{(n)} \sim \mathbb{P}_i$. Then

$$x_i(n) \to x^* a.s. \quad \iff \quad \text{all R-M conditions hold.}$$

We need to show that

$$\mathbb{E}[\epsilon_i(n) \mid \mathcal{F}_n] = 0, \quad \mathbb{E}[\epsilon_i(n)^2 \mid \mathcal{F}_n] \le A + B\|x(n)\|^2, \quad \sum_{i=1}^n \alpha_i(n) = \infty \text{ and } \sum_{i=1}^n \alpha_i(n)^2 < \infty \quad a.s.$$

hold. The first condition is easy (Note we set $b(n) \equiv 0$):

$$\begin{aligned} \mathbb{E}[\epsilon_i(n) \mid \mathcal{F}_n] &= \mathbb{E}[f(x(n), Z_i(n)) \mid \mathcal{F}_n] - \mathbb{E}[F_i(x) \mid \mathcal{F}_n] \\ &= \mathbb{E}[f(x(n), Z_i(n)) \mid \mathcal{F}_n] - F_i(x)\mathbb{E}[1 \mid \mathcal{F}_n] \\ &= \mathbb{E}[f(x(n), Z_i(n)) \mid \mathcal{F}_n] - \mathbb{E}_i[f(x,Z)] \\ &\overset{\text{stochastic Processes}}{=} 0. \end{aligned}$$

The second condition is a lot harder to show. The choices of the step sizes, is up to us to choose, but the choice needs to satisfy these two conditions.
Now we will connect to the dynamic Programming algorithm approach of doing Value Iteration and Policy Iteration. We can write the Bellman operators as expectations

Remark 3.1

$$\begin{aligned} T^\pi V(s) &:= \sum_{a \in \mathcal{A}_s} \pi(a;s) \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V(s') \right) \\ &= \sum_{a \in \mathcal{A}_s} \pi(a;s) \sum_{r \in \mathcal{R}} r \cdot p(r;s,a) + \sum_{a \in \mathcal{A}_s} \pi(a;s) \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V(s') \\ &= \sum_{a \in \mathcal{A}_s} \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} \pi(a;s) p(\{r\} \times \{s'\}; s, a) \cdot r + \sum_{a \in \mathcal{A}_s} \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} \pi(a;s) p(\{r\} \times \{s'\}; s, a) \cdot \gamma V(s') \\ &= \mathbb{E}_s^\pi [R_0 + \gamma V(S_1)]. \end{aligned}$$

We want to write this now in the form of the Robins Monro recursion, i.e. in the form

$$\mathbb{E}_s^\pi[R_0 + \gamma V(S_1)] = \mathbb{E}_s[f(x, Z)]$$

where $x \in \mathbb{R}^{|\mathcal{S}|}$ is some vector and $Z \sim \mathbb{P}_s$. Now the question is how f and Z look like. We have that

$$Z := (Z_1, Z_2, Z_3) = (A, R, S'), \quad A \sim \pi(\cdot; s), R \sim p(\{\cdot\} \times \mathcal{S}; s, A), S' \sim p(\{R\} \times \{\cdot\}; s, A)$$

and thus $f(x, Z) := Z_2(s, Z_1) + \gamma x_{Z_3}$.

For the Bellman expectation of the Q value, the approach is identical, only that we have $\mathbb{P}_{s,a}^\pi$, thus the first action fixed.

Remark 3.2

$$\begin{aligned}
T^\pi Q(s,a) &:= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} p(s'; s, a)\pi(a'; s')Q^\pi(s'; a') \\
&= \sum_{r \in \mathcal{R}} r \cdot p(r; s, a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} p(s'; s, a)\pi(a'; s')Q^\pi(s'; a') \\
&= \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} \pi(a; s)p(\{r\} \times \{s'\}; s, a) \cdot r + \gamma \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} p(\{r\} \times \{s'\}; s, a)\pi(a'; s')Q^\pi(s'; a') \\
&= \mathbb{E}_{s,a}^\pi[R_0 + \gamma Q^\pi(S_1, A_1)].
\end{aligned}$$

We want to write this now in the form of the Robins Monro recursion, i.e. in the form

$$\mathbb{E}_{s,a}^\pi[R_0 + \gamma Q^\pi(S_1, A_1)] = \mathbb{E}_{s,a}[f(x, Z)]$$

where $x \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ is some matrix and $Z \sim \mathbb{P}_{s,a}$. Now the question is how f and Z look like. We have that

$$Z := (Z_1, Z_2, Z_3) = (R, S', A'), \quad R \sim p(\{\cdot\} \times \mathcal{S}; s, a), S' \sim p(\{R\} \times \{\cdot\}; s, a), A' \sim \pi(\cdot; S')$$

and thus $f(x, Z) := Z_1(s, a) + \gamma x_{Z_2, Z_3}$.

Analogously we can show this for the Bellman optimality operator for Q, but not for V as we will see in the following:

Remark 3.3

$$\begin{aligned}
T^*Q(s,a) &:= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) \max_{a' \in \mathcal{A}_{s'}} Q^\pi(s';a') \\
&= \sum_{r \in \mathcal{R}} r \cdot p(r;s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a'' \in \mathcal{A}_{s'}} p(s';s,a)\pi(a'';s') \max_{a' \in \mathcal{A}_{s'}} Q^\pi(s';a') \\
&= \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} \pi(a;s) p(\{r\} \times \{s'\};s,a) \cdot r + \gamma \sum_{r \in \mathcal{R}} \sum_{s' \in \mathcal{S}} \sum_{a'' \in \mathcal{A}_{s'}} p(\{r\} \times \{s'\};s,a)\pi(a'';s') \max_{a' \in \mathcal{A}_{s'}} Q^\pi(s';a') \\
&= \mathbb{E}^\pi_{s,a}[R_0 + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q^\pi(S_1,a')].
\end{aligned}$$

We want to write this now in the form of the Robins Monro recursion, i.e. in the form

$$\mathbb{E}^\pi_{s,a}[R_0 + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q^\pi(S_1,a')] = \mathbb{E}_{s,a}[f(x,Z)]$$

where $x \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}|}$ is some matrix and $Z \sim \mathbb{P}_{s,a}$. Now the question is how f and Z look like. We have that

$$Z := (Z_1, Z_2) = (R, S'), \quad R \sim p(\{\cdot\} \times \mathcal{S}; s, a), S' \sim p(\{R\} \times \{\cdot\}; s, a)???$$

and thus $f(x,Z) := Z_1(s,a) + \gamma \max_{a' \in \mathcal{A}_{Z_2}} x_{Z_2,a'}$.
Now one would also like to do apply the Robins Monro algorithm for T^*V, but here we get the problem that

$$\begin{aligned}
T^*V(s) &= \max_{a \in \mathcal{A}_s} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s';s,a) V(s') \right) \\
&= \max_{a \in \mathcal{A}_s} \mathbb{E}^\pi_{s,a}[R_0 + \gamma V(S_1)]
\end{aligned}$$

But we have no way to sample this function, as the maximum is outside of the expectation.

In the stochastic control community, they often use T^*V in order to calculate the optimal policy, as the vector V is a lot smaller than the matrix Q. But with our Robin Monro approach, we cannot sample from T^*V, thus we only have the ability to use T^*Q.

3.1 Sample based policy evaluation algorithms

Theorem 3.1

Suppose we have $\pi \in \Pi_s$ and the reward distribution has bounded second moments, i.e.

$$\forall s, s' \in \mathcal{S}, a \in \mathcal{A}_s : \quad \mathbb{E}_{s,a}[R^2] < \infty, \quad R \sim p(\{\cdot\} \times \{s'\}; s, a)$$

then for any initial vector $V_0 \in \mathbb{R}^{|\mathcal{S}|}$ we define the (totally asynchronous) update rule

$$V_s(n+1) := (1 - \alpha_s(n))V_s(n) + \alpha_s(n)(r_n + \gamma V_{s'_n}(n)),$$

where $a \sim \pi(\cdot; s)$, $(s'_n, r_n) \sim p(\cdot; s, a)$ and the step size satisfies

$$\alpha_{s'_n}(n) \in (0,1) \text{ and all other are zero, i.e. } \alpha_s(n) = 0, s \neq s'_n$$

and it satisfies the Robins Monro condition, i.e. for all $s \in \mathcal{S}$

$$\sum_{n=1}^{\infty} \alpha_s(n) = \infty, \quad \text{and} \quad \sum_{n=1}^{\infty} \alpha_s(n)^2 < \infty, \quad a.s.$$

Then it follows that $\lim_{n\to\infty} V(n) = V^*$ $a.s.$

Proof. Zweiter Stichpunkt letzes ungleichheitszeichen $x \leq 1 + x^2$??? □

We can only update totally asynchronously, because we can only simulate one step.

Remark 3.4

It is very important to note that the choice of the $\alpha_s(n)$ satisfying these two conditions theoretically from Robins Monro is necessary, but not sufficient, because the underlying policy has to ensure that every state is explored infinitely often in order for $n \to \infty$ in every state.

Theorem 3.2

Suppose we have $\pi \in \Pi_s$ and the reward distribution has bounded second moments, i.e.

$$\forall s, s' \in \mathcal{S}, a \in \mathcal{A}_s : \quad \mathbb{E}_{s,a}[R^2] < \infty, \quad R \sim p(\{\cdot\} \times \{s'\}; s, a)$$

then for any initial matrix $Q_0 \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ we define the (totally asynchronous) update rule

$$Q_{s,a}(n+1) := (1 - \alpha_s(n))Q_{s,a}(n) + \alpha_s(n)(r_n + \gamma Q_{s'_n,a'_n}(n)),$$

where $(s'_n, r_n) \sim p(\cdot; s, a)$, $a'_n \sim \pi(\cdot; s)$ and the step size satisfies

$$\alpha_{s'_n,a'_n}(n) \in (0, 1) \text{ and all other are zero, i.e. } \alpha_{s,a}(n) = 0, (s, a) \neq (s'_n, a'_n)$$

and it satisfies the Robins Monro condition, i.e. for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$

$$\sum_{n=1}^{\infty} \alpha_{s,a}(n) = \infty, \quad \text{and} \quad \sum_{n=1}^{\infty} \alpha_{s,a}(n)^2 < \infty, \quad a.s.$$

Then it follows that $\lim_{n\to\infty} Q(n) = Q^*$ $a.s.$

Here again the issue is that the policy needs to ensure that all state action pairs are explored infinitely often. The theorems of Robins Monro only tells us to explore infinitely often, but not on how exactly to explore. This leads us again to the bandit problems of exploration vs. exploitation.

Remark 3.5

A reasonable choice for the step size is

$$\alpha_s(n) = \frac{1}{(T_s(n)+1)^p}, \quad \alpha_{a,s}(n) = \frac{1}{(T_{s,a}(n)+1)^p}, \quad p \in (1/2, 1]$$

because then the two Robins Monro conditions hold. Here $T_{s,a}(n)$ is the number of times the state-action pair (s, a) was updated during the first n updates.

Algorithm 11: Totally asynchronous policy evaluation for V^π

Data : Policy $\pi \in \Pi$, discount factor $\gamma \in (0, 1)$, learning rate schedule $\{\alpha(s)\}$
Result: Approximation $V \approx V^\pi$
Initialize $V \equiv 0$
stop $\leftarrow$ False
while *stop* = *False* **do**
  `// Pick a state, sample reward and next state, then update` V`.`
  Choose state s (e.g. randomly or in sequence).
  $a \sim \pi(\cdot\,;\, s)$
  Observe $R(s, a)$ and $s' \sim p(\cdot\,;\, s, a)$
  $\alpha \leftarrow \alpha(s)$
  `// Update rule for approximate policy evaluation`

$$V(s) \leftarrow V(s) \;+\; \alpha\Big(R(s,a) \;+\; \gamma\, V(s') - V(s)\Big)$$

  `// Stopping criterion or convergence test here`
  if *converged* **then**
    stop $\leftarrow$ True
return V

Algorithm 12: Totally asynchronous policy evaluation for Q^π

Data : Policy $\pi \in \Pi$, discount factor $\gamma \in (0, 1)$, learning rate schedule $\{\alpha(s, a)\}$
Result: Approximation $Q \approx Q^\pi$
Initialize $Q \equiv 0$
stop $\leftarrow$ False
while *stop* = *False* **do**
  `// Pick a state-action pair, sample reward and next state, then update` Q`.`
  Choose (s, a) (e.g. randomly or in sequence).
  Observe $R(s, a)$ and $s' \sim p(\cdot\,;\, s, a)$
  $a' \sim \pi(\cdot\,;\, s')$
  $\alpha \leftarrow \alpha(s, a)$
  `// Update rule for approximate Q-value evaluation`

$$Q(s,a) \leftarrow Q(s,a) \;+\; \alpha\Big(R(s,a) \;+\; \gamma\, Q\big(s', a'\big) - Q(s,a)\Big)$$

  `// Stopping criterion or convergence test here`
  if *converged* **then**
    stop $\leftarrow$ True
return Q

These two algorithms are both off policy algorithms, i.e. they do not play according to the Policy, but according to some exploration scheme.

3.2 Q-learning and the SARSA trick

We now come to the most famous tabular control algorithm: Q-learning, which is numerically solving the iteration T^*Q of the Bellman optimality operator.

- Advantage: Flexibility in the exploration of the state action space. We will soon call this off Policy.
- Drawback: Very slow, because a special case of it is the LLN, which is very slow in the L^2-error. Further, as we will show, there will be a problem of overestimation.

Theorem 3.3: Q-Learning

Suppose that the reward distribution has bounded second moments, i.e.

$$\forall s, s' \in \mathcal{S}, a \in \mathcal{A}_s : \quad \mathbb{E}_{s,a}[R^2] < \infty, \quad R \sim p(\{\cdot\} \times \{s'\}; s, a)$$

Then for any initial matrix $Q_0 \in \mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}|}$ we define the (aysnchornous) update rule

$$Q_{s,a}(n+1) := (1 - \alpha_{s,a}(n))Q_{s,a}(n) + \alpha_{s,a}(n)(r_n + \gamma \max_{a' \in \mathcal{A}_{s'}} Q_{s',a'}(n) - Q_{s,a}(n)),$$

where we assume that $(r_n, s') \sim p(\cdot; s, a)$ and $\alpha(n)$ only depends on the past steps and almoust surely satisfies the Robins-Monro Conditions for every $(s, a) \in \mathcal{S} \times \mathcal{A}$. Then it follows that

$$\lim_{n\to\infty} Q(n) = Q^* \quad a.s.$$

Proof. The proof is similiar, with the only difference that we use the Bellman optimality operator

$$T^*Q(s, a) = \mathbb{E}_{s,a}[R_0 + \gamma \max_{a' \in \mathcal{A}_{S_1}} Q(S_1, a')]$$

on $\mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}|}$. ... □

Implimentation:
What we can do is run trough the state action space according to some scheme and and in each step update one $Q_{s,a}(n)$ using totally asynchronous updates. But this very similiar to the bandit situation, as we need a method of exploration, that is smart in order to ensure, that we do not waste our exploration on state action pairs, that are not good.

Remark 3.6

As for convergence the step size sequence $(\alpha_n)_{n\in\mathbb{N}}$ only needs to satisfy the Robins Monro condition and be adapted to the past steps (i.e. only depend on the past and not the future) we have a lot of freedom in choosing it in a certain way. But it is important to note that we need to explore every state infinitely often.

There are four typicall methods that are used for exploration

- Uniformly choose (s, a) in each step.
- Run through (s, a) using some fixed policy π (behaviroial policy)
- Act ϵ-greedy wrt. the current estimate $Q(n)$

- Act soft $\epsilon-$greedy wrt. the current $Q(n)$ (Boltzman exploration)

Similiarly to stochastic bandits we can choose ϵ tending to zero. But we need to be carefull, that we still visit every state action pair infinitely often. We want to find a behavioral policy that explores the state space smart, i.e. avoid state action pairs that are definitely not relevant and instead ones that are relevant. The following algorithm chooses the next action according to some behavioral policy.

Algorithm 13: Q-learning

Input : Behavior policy $\pi \in \Pi_S$, discount factor γ
Output: Approximations $Q \approx Q^*$, $\text{greedy}(Q) = \pi \approx \pi^*$
Initialize Q (e.g. $Q \equiv 0$);
while *not converged* **do**
  Initialise s;
  while s *not terminal* **do**
    Sample action a. (e.g. randomly, according to a behavior policy, ε-(soft)-greedy);
    Sample reward $R(s,a)$;
    Sample $s' \sim p(\cdot \mid s,a)$;
    Determine stepsize α;
    Update

$$Q_{s,a} = (1-\alpha)\, Q_{s,a} + \alpha\big(R(s,a) + \gamma \max_{a' \in A_{s'}} Q_{s',a'}\big)$$

    ;
    Set $s = s'$;
return Q

Definition 3.4

An exploration mechanism is called off-policy, if the current action does not depend on the current policy/Q-Values. It is called on-policy if this exploration mechanism uses current estimated Q-values or information of the policy.

One preffers on-policy exploration mechanism, as they use more information as off-policy methods???
Intuition of Q-Learning:
The algorithm challenges the old estimate $Q(s,a)$ with a new estimate $R(s,a)$ (the reward of taking action a ins state s) plus the value of the current best estimate:

$$Q(s,a) \leftarrow (1-\alpha_n(s,a))Q(s,a) + \alpha_n(s,a)(R(s,a) + \gamma \max_{a'} Q(s',a'))$$

This is due to:

Let π and π' be arbitrary policies then

$$\begin{aligned}
T^{\pi} \max_{a\in\mathcal{A}_s} Q^{\pi'}(s,a) &= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \sum_{a'\in\mathcal{A}_{s'}} \pi(a';s') \max_{a'\in\mathcal{A}_{s'}} Q^{\pi'}(s',a') \\
&= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} Q^{\pi'}(s',a') \underbrace{\sum_{a'\in\mathcal{A}_{s'}} \pi(a';s')}_{=1} \\
&= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} Q^{\pi'}(s',a') \\
&= r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} \max_{a'\in\mathcal{A}_{s'}} Q^{\pi'}(s',a') \\
&= T^{*} \max_{a\in\mathcal{A}_s} Q^{\pi'}(s,a).
\end{aligned}$$

Now let $\pi := greedy(Q^{\pi'})$, then $Q^{\pi}(s,a) = \max_{a\in\mathcal{A}_s} Q^{\pi'}(s,a)$ for (s,a), because with the Lemma that $T^{\pi}q = T^{*}q$ is eqivalent to that π is greedy wrt. q, it follows that

$$Q^{\pi}(s,a) = T^{\pi}Q^{\pi}(s,a) = T^{*}Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s';s,a) \max_{a'\in\mathcal{A}_{s'}} \underbrace{Q^{\pi}(s',a')}_{=\max_{a'\in\mathcal{A}_{s'}} Q^{\pi}(s',a')} = T^{*} \max_{a\in\mathcal{A}_s} Q^{\pi}(s,a)$$

Because T^{*} is unqiue and $Q^{\pi}(s,a) = T^{*}Q^{\pi}(s,a) = T^{*} \max_{a\in\mathcal{A}_s} Q^{\pi}(s,a)$ the equality follows.
Thus $\max_{a'\in\mathcal{A}_{s'}} Q^{\pi}(s,a)$ is the best current estimate, because it is the Q function of the greedy policy wrt. the current Q-function.

The factor $\alpha_n(s,a)$ is the "trust" we give a new estimate. This should intuitively go to zero and also due to Robins-Monro.
Drawbacks of Q-learning:

- First Note that stochastic fixed point iterations are in general not unbiased, i.e. $\mathbb{E}[x_n] \neq x^{*}$ for finite $n \in \mathbb{N}$. This is because $\epsilon(n)$ and $b(n)$ are assumed to be positive (bias tends to zero).

- Q-learning considers the function

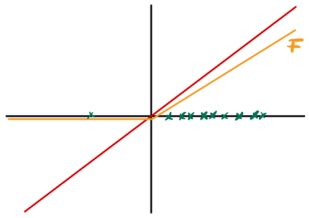
  $$F(Q) := \mathbb{E}[R(s,a) + \gamma \max_{a'} Q(S_1,a')].$$

  Lets consider two actions, i.e. $f(Q) = R(s,a) + \gamma \max\{Q(s',a), Q(s',a'\}$. Making this a little simpler we assume $f(Q) = \gamma \max\{0, Q(s,a)\}$. Then the fixed point iteration

  $$Q_{n+1}(s,a) := (1-\alpha_n)Q_n(s,a) + \alpha_n f(Q_n)$$

  can get stuck. I.e. we have the function $F(x) := \gamma \max\{0,x\}$, for $\gamma < 1$. Then we have two cases. If x_n is non-positive, it converges really quickly. If x_n is positive, then it gets stuck/converges really slowly:

  $$x_{n+1} := (1-\alpha_n)x_n + \alpha_n F(x_n) = \begin{cases} (1-\alpha_n)x_n + 0 & , x_n \leq 0 \\ (1-\alpha_n)x_n + \alpha_n \gamma x_n < x_n & , x_n > 0 \end{cases}$$



An overestimating approximate fixedpoint iteration

Figure 4.3

Thus if the Q-value is overestimated, then recovery takes a very long time. If it is underestimated, it reaches quickly Q^*. But due to

$$\hat{Q}_n(s,a) = Q^*(s,a) + \epsilon_n(s,a)$$

with an error term that does not vanish, the errors can kick us from a value close Q^* away into a positive region, where recovery takes a long time. Thus Q-learning by construction overestimates its Q-values due to non vanishing errors and the slow recovery from values above.

- Next, because we need to explore every state infinitely often, Q-learning is considered to be dangerous, because we also have to explore states that can be known to be dangerous for the agent. It is also not inclined to keep a distance from (s,a) that are deemed as dangerous.

Remark 3.7

A first workaround is adding a negative bias term that tends to zero, i.e. satsifies the Robins-Monro condition:

$$Q_n(s,a) = (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(n)(R(s,a) + \gamma \max_{a'} Q(s',a') + b_n(s,a))$$

where $s' \sim p(\cdot \times \{R(s,a)\}; s, a)$. If $b_n(s,a)$ converges fast enough to zero, then Q_n converges almoust surely to Q^*.

SARSA is another way to tackle the problems above (older than Q-learning). It can reduce playing dangouros action. Later we look at double Q-learning and see it tackles the problem of overestimation.

Definition 3.5

We call an update mechanism online, if the algorithm only uses currently observed state action pairs. It is called offline, if it uses state action pairs that are not currently observed. (Definition not as in Literature).

The update mechanism of Q-learning is offline, as it uses with the maximum hypothetical actions, that are not updated. SARSA will be a "online version" on policy (GLIE) version of biased Q-learning.

Remark 3.8

SARSA: (State action Reward (next) state action)
Here the idea is to avoid the maximum, but get it back in the limit. I.e. we construct the behavioral policy of the on-policy exploration mechanism, such that it is greedy in the limit (optimal policies are greedy, i.e. if it converges it will be automatically greedy).

$$Q_n(s,a) \leftarrow (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(s,a)(R(s,a) + \gamma Q(s',a'))$$

where $s' \sim p(\cdot \times \{R(s,a)\}; s, a)$ and $a' \sim \pi(\cdot; s')$.

So having a Behavioral policy that is GLIE will be shown to converge to Q^*. If we have any other behavioral policy π then the same iteration scheme converges to Q^π as in the Lemma above. Thus the SARSA trick entirely lies on this GLIE property! This will be seen in the following.

Algorithm 14: SARSA

Input : Behavior policy (GLIE) $\pi \in \Pi_S$, discount factor γ
Output: Approximations $Q \approx Q^*$, greedy$(Q) = \pi \approx \pi^*$
Initialize Q, e.g. $Q \equiv 0$;
while *not converged* **do**
  Initialise s, a, e.g. uniformly;
  while s *not terminal* **do**
    Determine stepsize α;
    Sample reward $R(s,a)$;
    Choose new (behavioral) policy π from Q (e.g. ε-greedy);
    Sample next state $s' \sim p(\cdot \mid s, a)$;
    Sample next action $a' \sim \pi(\cdot \mid s')$;
    Update

$$Q_{s,a} \leftarrow Q_{s,a} + \alpha_n(s,a)\big(R(s,a) + \gamma\, Q_{s',a'} - Q_{s,a}\big);$$

    Set $s \leftarrow s'$, $a \leftarrow a'$;
return Q

Definition 3.6: GLIE

We call an exploration mechanism GLIE, if it is

- in the limit almoust surely greedy wrt. to state action value function (Q-Value)
- It visists every state action pair infinitely often.

Theorem 3.7

Assume that $(Q_n)_{n\in\mathbb{N}}$ is a sequence of matrices obtained from the SARSA algorithm. Further, let the exploration mechanism be GLIE, the step sizes satisfy the Robins Monro condition and the reward distributions having bounded second moments.
Then $\lim_{n\to\infty} Q_n = Q^*$ almoust surely.

Proof. We can write SARSA as Q-learning with a bias term:

$$\begin{aligned}Q_n(s,a) &= (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(s,a)(R(s,a)+\gamma Q(s',a')) \\ &= (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(s,a)(R(s,a)+\gamma \max_{a'} Q(s',a') + \underbrace{R(s,a)+\gamma Q(s',a') - (R(s,a)+\gamma \max_{a'} Q(s',}_{=:b_n(s,a)}\end{aligned}$$

And we wrote Q-learning as

$$\begin{aligned}Q_n(s,a) &= (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(s,a)(R(s,a)+\gamma \max_{a'} Q(s',a') + b_n(s,a)) \\ &= (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(s,a)(T^*Q_n(s,a) + R(s,a)+\gamma \max_{a'} Q(s',a') - T^*Q_n(s,a) + b_n(s,a)) \\ &= (1-\alpha_n(s,a))Q_n(s,a) + \alpha_n(s,a)(T^*Q_n(s,a) + \epsilon_n(s,a) + b_n(s,a))\end{aligned}$$

It remains to show that

$$|b_n(s,a)| \leq \omega_n(\|Q_n(s,a)\|+1), \quad a.s.$$

for an almost surely zero sequence $(\omega_n)_{n\in\mathbb{N}}$.
This probably comes from bounded second moments and $b_n(s,a) \leq 0$???
With the GLIE property we get that in the limit the bias b_n vanishes, because we choose greedy. The conditions for infinite exploration are necassary such that the step sizes fullfil their properties. □

This theorem is quite trivial, as we assumed our bias that we added to vanish in the limit.

Example 3.1

One example of a GLIE exploration mechanism is if we do it $\epsilon_n := \frac{1}{T_{s,a}(n)}$-greedy.
Another way is Boltzman exploration based on $Q(n)$, i.e.

$$\pi_n(a;s) := \frac{e^{\log(T_{s,a}(n))\cdot Q_n(s,a)}}{\sum_{b\in\mathcal{A}_s} e^{\log(T_{s,b}(n))\cdot Q_n(s,b)}}$$

These behavioral polices are greedy in the limit, but also explore every state action pair infinitely often.

The non-positive Bias term ensures, that Q-values do not get overestimated. Further, neighboring states, i.e. the s' in the formula, that have high variability in their rewards get due to construction an even more negative bias. I.e. in finite time SARSA avoids theses states. But in infinite time, this does not matter. Because bad states can have high variety in their rewards, SARSA learns safer. But very good states could also be high in variety.

Remark 3.9

SARSA is a sample based version of Polciy Iteration. Suppose we have ϵ-greedy exploration. We evaluate the current policy by taking one step/we update the Q-value with one step and then with proability $1-\epsilon$ playing greedy, i.e. policy improvement. (But not quite??? Q-Values do not get deleted?)

Remark 3.10

Variance reduction applied to $Q-$learning. Before we can talk about this we need to show the following two Bellman-n-step equations

$$T_n^\pi V(s) = \mathbb{E}_s^\pi[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V(S_n)]$$

and

$$T_n^\pi Q(s, a) = \mathbb{E}_{s,a}^\pi[R(s, a) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n Q(S_n, A_n)].$$

A unique solution exists, if we show that they are contractions. We do it for V:

$$\begin{aligned}
\|T_n^\pi V_1 - T_n^\pi V_2\|_\infty &= \max_{s\in\mathcal{S}} |T_n^\pi V_1(s) - T_n^\pi V_2(s)| \\
&= \max_{s\in\mathcal{S}} |\mathbb{E}_s^\pi[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V_1(S_n)] \\
&\quad - \mathbb{E}_s^\pi[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V_2(S_n)]| \\
&= \max_{s\in\mathcal{S}} |\mathbb{E}_s^\pi[\gamma^n (V_1(S_n) - V_2(S_n))]| \\
&\leq \gamma^n \max_{s\in\mathcal{S}} |\mathbb{E}_s^\pi[\max_{s_n\in\mathcal{S}} (V_1(s_n) - V_2(s_n))]| \\
&= \gamma^n \max_{s_n\in\mathcal{S}} |V_1(s_n) - V_2(s_n)| \\
&= \gamma^n \|V_1 - V_2\|_\infty.
\end{aligned}$$

Analogously for Q. In order to show that the fixed point iteration from the Robins-Monro Theorem converges towards that unique fixed point, we need to show that the error term is bounded, i.e.

$$\epsilon_s(n) := \left(R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V(S_n) \right) - \mathbb{E}_s^\pi[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V(S_n)].$$

$\epsilon(n)$ is $\mathcal{F}_n$ measurable,

$$\mathbb{E}[\epsilon_s(n) \mid \mathcal{F}_n] = \mathbb{E}_s^\pi[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V(S_n) \mid \mathcal{F}_n] - \mathbb{E}_s^\pi[R(s, A_0) + \sum_{t=1}^{n-1} \gamma^t R(S_t, A_t) + \gamma^n V(S_n)] = \ldots = 0$$

and

$$\mathbb{E}[\epsilon_s(n)^2 \mid \mathcal{F}_n] \overset{hard}{\leq} A + b\|V\|_\infty.$$

Finally, the n-step Bellman expecation equation and the Bellman expectation equation have the same

fixed point, due to

$$\begin{aligned}
T^\pi Q^\pi(s,a) &= r(s,a) + \gamma \sum_{s',a'} p(s';s,a)\pi(a';s')Q^\pi(s',a') \left(= \mathbb{E}^\pi_{s,a}[\gamma R(S_0,A_0) + \gamma Q^\pi(S_1,A_1)]\right) \\
&= r(s,a) + \gamma \sum_{s',a'} p(s';s,a)\pi(a';s') \left(r(s',a') + \gamma \sum_{s'',a''} p(s'';s',a')\pi(a'';s'')Q^\pi(s'',a'') \right) \\
&= r(s,a) + \gamma \sum_{s',a'} p(s';s,a)\pi(a';s')r(s',a') + \gamma^2 \sum_{s',a'} p(s';s,a)\pi(a';s') \sum_{s'',a''} p(s'';s',a')\pi(a'';s'')Q^\pi(s'',a'') \\
&= \mathbb{E}^\pi_{s,a}[R(s,a) + \gamma R(S_1,A_1) + \gamma^2 Q^\pi(S_2,A_2)]
\end{aligned}$$

Via induction it holds that $T^\pi_n Q^\pi = T^\pi_m Q^\pi$ for all $m \neq n$. Thus the iteration

$$Q_{s,a}(n+1) = (1-\alpha_{s,a}(n))Q_{s,a}(n) + \alpha_{s,a}(n)(R(s,a) + \sum_{t=1}^{n-1} \gamma^t R(S_t,A_t) + \gamma^n Q(S_n,A_n)),$$

converges to the same fixed point almoust surely. If we have $\alpha_{s,a}(n) = \frac{1}{n}$ and $n \to \infty$ then we have just a Monte Carlo approximation (Memory trick). For $n = 1$ we have the iteration from theorem that did T^π. If $n = 1$ and we play with a GLIE behavioral policy, then we get SARSA. One can also show that the Bellman optimality operator can be written in the n-step fasion:

$$T^*_n Q = \mathbb{E}^\pi_{s,a}[R(s,a) + \sum_{t=1}^{n-1} \gamma^t R(S_t,A_t) + \gamma^n \max_{a' \in \mathcal{A}_{S_n}} Q(S_n,a')].$$

Then n-step Q-learning is

$$Q_{s,a}(n+1) = (1-\alpha_{s,a}(n))Q_{s,a}(n) + \alpha_{s,a}(n)(R(s,a) + \sum_{t=1}^{n-1} \gamma^t R(S_t,A_t) + \gamma^n \max_{a' \in \mathcal{A}_{S_n}} Q(S_n,a')),$$

Again for $n = 1$ we have Q-learning and $n \to \infty$ we have Monte Carlo.
The idea is to find a n such that we reduce the variance of the iteration.

Algorithm 15: n-Step SARSA

Input : Behavior policy (GLIE) $\pi \in \Pi_S$, discount factor γ
Output: Approximations $Q \approx Q^*$, $\text{greedy}(Q) = \pi \approx \pi^*$
Initialize Q, e.g. $Q \equiv 0$;
while *not converged* **do**
  Initialise s, a, e.g. uniformly;
  while s *not terminal* **do**
    Set $R = 0$;
    for $i = 0, ..., n-1$ **do**
      Sample reward $R(s,a)$;
      Set $R = R + \gamma^i R(s,a)$;
      Choose new (behavioral) policy π from Q (e.g. ε_N-greedy);
      Sample next state $s' \sim p(\cdot \mid s, a)$;
      Sample next action $a' \sim \pi(\cdot \mid s')$;
    Determine step size α_N;
    Update

$$Q_{s,a}(N+1) \leftarrow Q_{s,a}(N) + \alpha_{s,a}(N)\big(R(s,a) + \gamma\, Q_{s',a'}(N) - Q_{s,a}(N)\big);$$

    Set $s \leftarrow s'$, $a \leftarrow a'$;
return Q

A very important point in these methods is that it all comes down to the behavioral policy. Uniform exploration is really bad, because it takes a lot of time to find out, that the terminal state, is the best state and advantagous to reach. This is why Boltzmann and ϵ_n-greedy are beneficial. Further, if the behavioral policy is initialized in a way (for example by learning iterations before) then convergence will be also a lot faster.

3.3 Others: Deep Q and alternate interpretation

In a game like atarie the entire state space are all the pixels. But in this game particular it would also be playable, if we blurred the image, i.e. having less states. This is beneficial for computational costs. An approach would be to use a neural network to approximate the Q-value, i.e. $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{\mathcal{S}\cdot\mathcal{A}}$. This is a non-tabular methods, because if we change one parameter then all outputs are changed. The neural network is the some parameterized function Q_Θ. Problems here are overfitting. One can solve this by learning in batches (backpropagation only after a few steps) or using replay buffers (having batches contain old information, such that it does not forget the old information).
Finally, there is another way of interpreting Q-learning:
We have that

$$\begin{aligned}\mathbb{E}[X] &\in \arg\min_\theta \mathbb{E}[(X-\theta)^2] \\ &= \arg\min_\theta \frac{1}{2}\mathbb{E}[(X-\theta)^2] =: \arg\min_\theta F(\theta)\end{aligned}$$

Then under certain assumptions $\frac{d}{d\theta}F(\theta) = -\mathbb{E}[X-\theta]$. Thus the iteration

$$\theta_{n+1} := \theta_n + \alpha_n(X - \theta_n) = (1-\alpha_n)\theta_n + \alpha X$$

is a gradient descent method, that uses one sample. Because in Q-learning we also have an expectation, one can also interpret it as the same principle. Here then $X := R_n + \gamma \max_{a' \in \mathcal{A}_{s'_n}} Q_{S'_n,a'}(n)$. Using Extreme Value

theory we have that for an unbounded distribution with u_0 large enough, we have that for all $u \geq u_0$ holds that

$$(X - u \mid X > u) \sim GPD.$$

Thus it holds that

$$\mathbb{P}(X \leq t) = \mathbb{P}(X \leq t, X < u) + \mathbb{P}(X \leq t, X \geq u).$$

Now using $Y := (X - u \mid X > u) \sim GPD$ and for $u < t$ we have that the above is analogously to

$$\mathbb{P}(X \leq t) = \underbrace{\mathbb{P}(X \leq t, X < u)}_{=\mathbb{P}(X \leq u) =: \eta} + \underbrace{\mathbb{P}(u < X \leq t)}_{=\mathbb{P}(X - u \leq t, X > u)} = \eta + \mathbb{P}(X > u)\mathbb{P}(Y \leq t) = \eta + (1 - \eta)\mathbb{P}(Y \leq t)$$

Now we apply this to $T^*Q(s,a) = \mathbb{E}^{\pi}_{s,a}[R_0 + \max_{a' \in \mathcal{A}_{s'}} Q(S', a')]$, where $(R_0, S') \sim p(\cdot; s, a)$.
For every $i = 1, ..., N$ and (s, a) chosen by the exploration mechanism in Q-learning, sample reward and next state and calculate $X^i_{s,a} := R^i_0 + \max_{a' \in \mathcal{A}_{S',i}} Q(S'^{,i}, a')$. Then choose Batch size $n = kN$, i.e k many updates and do for every i=1,...,k:

- Step 1: Find smallest u such that

$$(X^i_{s,a} - u \mid X^i_{s,a} > u) \sim GPD$$

- Step 2: Define a Generalized linear model

$$\mathbb{E}[\mathbf{1}_{X^i_{s,a} > u}] = g^{-1}((w^i_{s,a})^T \beta)$$

  such that β^* is the best fit for the expected value for higher values than u and define $\hat{\eta} := g^{-1}(w_{s,a}\beta^*)$.

- Step 3: Find the MLE for the two probability distributions on the bulk $f_1^{\theta^1_{s,a}}$ and on the tail $f_2^{\theta^2_{s,a}}$:

$$l(\theta^1_{s,a}, \theta^2_{s,a}) := \sum_{X^i_{s,a} \leq u} \log(f_1^{\theta^1_{s,a}}(X^i_{s,a})) + \sum_{X^i_{s,a} > u} (\log(f_2^{\theta^2_{s,a}}(X^i_{s,a})) + \log(1 - \hat{\eta})),$$

  via stochastic gradient descent. Then define

$$E_1 := \int x f_1^{\theta^1_{s,a}}(x)dx, \quad E_2 := \int x f_2^{\theta^2_{s,a}}(x)dx$$

- Step 4: Q-learning Update (This is then actually a n-step Q-learning update)

$$Q_{s,a}(i+1) = (1 - \alpha_{s,a}(i))Q_{s,a}(i) + \alpha_{s,a}(i)(\eta E_1 + (1 - \eta)E_2).$$

CHAPTER 5

GRADIENT DESCENT METHODS

Our goal is to do policy gradient methods, i.e. we parametrize the policy $\{\pi^\theta \mid \theta \in \mathbb{R}^d\}$ and define the value function

$$J(\theta) := \mathbb{E}_\mu^{\pi^\theta}[\sum_{t=0}^{\infty} \gamma^t R_t]$$

for a fixed initial distribution μ. Finding here an optimal policy for this problem statement is different to our usual way. This is because we defined optimal policies for every starting state action pair (s, a)! Here μ is fixed. Therefore for

$$\theta^* \in \arg\max_{\theta \in \Theta} J(\theta)$$

π^{θ^*} is not an optimal policy!
In this section we focus on gradient descent methods for functions $f : \mathbb{R}^d \to \mathbb{R}$. Remember that the negative gradient is the direction of the steepest descent, i.e.

$$-\frac{1}{\|\nabla f(x)\|}\nabla f(x) \in \arg\min_{\|d\|=1} \underbrace{f'(x)d}_{\text{slope in direction d}} = \arg\min_{\|d\|=1} \langle f'(x), d\rangle = \arg\min_{\|d\|=1} \sum_{i=1}^{n} \frac{\partial}{\partial x_i} f(x) \cdot d$$

This motivates the minimization scheme that goes step by step into the direction of the gradient multiplied by a step size

$$\forall x_0 \in \mathbb{R}^d : \quad x_{n+1} := x_n - \alpha \nabla f(x_n)$$

Because only first derivatives are involved we call this numerical method first order scheme. The entire theory builds on how to choose the step size, as this is not trivial. A simple example is $f(x) = A\|x\|^2$. For $A = Id$ the iteration scheme can converge in one step if we choose the step size smart

$$\nabla f(x) = 2x, \text{ then } x_1 = x_0 - \alpha 2x_0 = 0 \quad \iff \quad \alpha = \frac{1}{2}.$$

If we choose A in such a way that it gets narrow valleys then the convergence is very slow, i.e. increase the values of the entries beside the diagonal. Here is the problem that the function is flat in one direction (in the valleys) and very steep in the other (down the ridge). This is reflected in the ratio of the largest and smallest Eigenvalues of the Hessian $\nabla^2 f(x)$. Thus function with smaller ratios of Eigenvalues are easier to do gradient descent on. We call such functions L-smooth. But we will not talk about Eigenvalues but use equivalent definitions that are more intuitive.
We will do some theory in the first section on L-smooth functions. This will not be very important for RL-theory. More important will be later so called PL-inequalities.

Note that methods containing the only the first derivative are called first order methods. Second order methods are not desirable, because the size of the Hessian can make so large, that its computation is not feasible.

Definition 0.1

We call a function $f : \mathbb{R}^d \to \mathbb{R}$ L-smooth if it is differentiable and the gradient ∇f is L-continuous, i.e.

$$\forall x, y \in \mathbb{R}^d : \quad \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

The property in this theorem is equivalent to the fact that the Hessian

$$\nabla^2 f(x) \in [-L, L].$$

Thus L-smooth functions do not have very strong curvature. Luckily we will later show that most functions for MDPs are L-smooth.

0.1 Gradient descent for L-smooth functions

In this section we will show under L-smooth functions that the gradient descent scheme if it converges, it will converge to a point that is stationary, i.e. with vanishing gradient. These are not necessarily local or global extreme points, but also saddle points.

Lemma 0.2: Descent Lemma

Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth with $L > 0$ then it holds that

$$\forall x, y \in \mathbb{R}^d : \quad f(x + y) \leq \underbrace{f(x) + y^T \nabla f(x) + \frac{L}{2}\|y\|^2}_{\text{tangent at x plus a quadratic term}}$$

Note that if L decreases then the upper bound decreases, thus making the function smoother. We will later show that gradient descent is faster on smoother functions

Proof. We define $\phi(t) := f(x + ty)$ and applying the chain rule yields

$$\phi'(t) = y^T \nabla f(x + ty), \quad t \in [0, 1].$$

Now with the fundamental theorem of calculus it follows that

$$\begin{aligned}
f(x+y) - f(x) = \phi(1) - \phi(0) &= \int_0^1 \phi'(t)dt \\
&= \int_0^1 y^T \nabla f(x+ty)dt \\
&= \int_0^1 y^T \nabla f(x+ty)dt + \int_0^1 y^T \nabla f(x)dt - \int_0^1 y^T \nabla f(x)dt \\
&= \int_0^1 y^T (\nabla f(x+ty) - \nabla f(x))dt + \int_0^1 y^T \nabla f(x)dt \\
&\leq \int_0^1 \|y\| \|\nabla f(x+ty) - \nabla f(x)\|dt + y^T \nabla f(x) \cdot 1 \\
&\overset{L\text{-smooth}}{\leq} \int_0^1 \|y\| L \|ty\|dt + y^T \nabla f(x) \\
&= L\|y\|^2 \int_0^1 tdt + y^T \nabla f(x) \\
&= \frac{L\|y\|^2}{2} + y^T \nabla f(x)
\end{aligned}$$

Befor L-smootheness we used cauchy schwarz, i.e. $\langle x, y\rangle \leq \|x\|\|y\|$. □

We can now show that gradient descent converges to a stationary point.

Theorem 0.3

Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and define the sequence $(x_k)_{k\in\mathbb{N}}$ as

$$x_{k+1} := x_k - \bar{\alpha}\nabla f(x_k), \quad \forall x_0 \in \mathbb{R}^d,$$

where the step size is chosen for $\epsilon < \frac{2}{L+1}$ as $\bar{\alpha} \in [\epsilon, \frac{2-\epsilon}{L}]$. It follows that every accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$ is a stationary point of f, i.e. $\nabla f(\bar{x}) = 0$.

The iteration scheme could also reach no accumulation point. I.e. it only converges to a stationary point, if the iteration reaches an accumulation point.

Proof. Since f is assumed to be L-smooth it follows that it is also smooth, i.e. for a convergent sequence $(x_n)_{n\in\mathbb{N}}$ with limit $\bar{x}$ it follows that every subsequence $(x_{n_k})_{k\in\mathbb{N}}$ converges to the same limit and also that $\lim_{k\to\infty} f(x_{n_k}) = f(\bar{x})$. Because it is a convergent sequence, it is also a cauchy sequence and satisfies by definition

$$\lim_{k\to\infty} (x_{n_k} - x_{n_{k-1}}) = 0$$

which implies that

$$\lim_{k\to\infty} (f(x_{n_k}) - f(x_{n_{k-1}})) \overset{MeanValueThm.}{=} \lim_{k\to\infty} \overbrace{\underbrace{f'(c_n)}_{c_n\in(x_{n_k}, x_{n_{k-1}})}}^{<\infty} (x_{n_k} - x_{n_{k-1}}) = 0$$

We now find a positive lower bound dependend on the norm of the gradient for the difference of one step, i.e. $f(x_k) - f(x_{k+1})$ and can then imply the assertion. We can apply the descent Lemma:

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= f(x_k) - f(\underbrace{x_k}_{x} \underbrace{-\bar{\alpha}\nabla f(x_k)}_{y}) \\ &\geq -y^T \nabla f(x) - \frac{L}{2}\|y\|^2 \\ &= (\bar{\alpha}\nabla f(x_k))^T \nabla f(x_k) - \frac{L}{2}\| - \bar{\alpha}\nabla f(x_k)\|^2 \\ &= \bar{\alpha}\|\nabla f(x_k))\|^2 - \frac{L\bar{\alpha}^2}{2}\|\nabla f(x_k)\|^2 \\ &= (\bar{\alpha} - \frac{L\bar{\alpha}^2}{2})\|\nabla f(x_k)\|^2 \end{aligned}$$

If we consider $\bar{\alpha} \in [\epsilon, \frac{2-\epsilon}{L}]$ and first plug in the upper bound and then the lower it follows

$$\begin{aligned} \bar{\alpha}(1 - \frac{L\bar{\alpha}}{2})\|\nabla f(x_k)\|^2 &\geq \bar{\alpha}(1 - \frac{L\frac{2-\epsilon}{L}}{2})\|\nabla f(x_k)\|^2 \\ &= \bar{\alpha}(1 - \frac{2-\epsilon}{2})\|\nabla f(x_k)\|^2 \\ &= \bar{\alpha}\frac{\epsilon}{2}\|\nabla f(x_k)\|^2 \\ &\geq \frac{\epsilon^2}{2}\|\nabla f(x_k)\|^2 \end{aligned}$$

Due to the fact that the lower bound of $f(x_k) - f(x_{k+1})$ is non-negative and the fact that our difference converge to zero, it follows that

$$\lim_{n\to\infty} \|\nabla f(x_k)\|^2 = 0 \quad \Rightarrow \quad \lim_{n\to\infty} \|\nabla f(x_k)\| = 0.$$

□

This theorem is very weak, as there is no statement weather this iteration scheme actually converges and there is no way of knowing if the accumulation point is a minimum or sattle point.

0.2 Gradient descent for L-smooth convex functions

Definition 0.4: Convex function

A function $f : \mathbb{R}^d \to \mathbb{R}$ is called convex if

$$\forall x, y \in \mathbb{R}^d : \quad f(y) \geq f(x) + (y - x)^T \nabla f(x)$$

Examples for convex functions are linear and quadratic functions. There is at every point a tangent plane that lies below the graph of f. Convex functions do not necessarily have to have a global minimum (linear functions), but we will assume this to be the case. We will always assume that a global minimum exists.

Remark 0.1

One can show that all local minima of convex functions must be global minima and all global minima have the same height.

We will denote by f_* the height of all global minima, i.e.

$$f_* := f(x^*), \quad x^* \in \arg\min_{x\in\mathbb{R}^d} f(x).$$

In the following analysis we will consider an error function $e : \mathbb{R}^d \to \mathbb{R}$ with the property $e(x) \geq 0$ for all $x \in \mathbb{R}^d$ and $e(x^*) = 0$. One typically chooses $e(x) = f(x) - f_*$.
Recall that we defined convergence to be "linear" (actually exponentially fast) if

$$\exists c \in (0,1) : \quad e(x_{k+1}) \leq ce(x_k).$$

We call it sublinear if it is slower. We will see that first order methods will be sublinear. This will be shown here

Theorem 0.5

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex $L-$smooth function and let $(x_k)_{k\in\mathbb{N}}$ be a seqence defined as

$$x_{k+1} := x_k - \bar{\alpha}\nabla f(x_k), \quad x_0 \in \mathbb{R}^d,$$

with $\bar{\alpha} \leq \frac{1}{L}$. Moreover assume that the set of global minimizers of f is non-empty. Then the sequence $(f(x_k))_{k\in\mathbb{N}}$ converges to f_* with speed

$$\exists c > 0 : \quad e(x_k) := f(x_k) - f_* \leq \frac{c}{k+1}, \quad k \in \mathbb{N}$$

and $f_* \in \min_{x\in\mathbb{R}^d} f(x)$.

Note as we assume there to be no unique global minimum, we only prove convergence to f_*.

Proof. Using the descent Lemma with $y = x_{k+1}$ and $x = x_k$ we get

$$\begin{aligned} f(x_{k+1}) = f(\underbrace{x_k}_{x} \underbrace{-\bar{\alpha}\nabla f(x_k)}_{y}) &\leq f(x_k) - (\bar{\alpha}\nabla f(x_k))^T\nabla f(x_k) + \frac{L}{2}\| - \bar{\alpha}\nabla f(x_k)\|^2 \\ &= f(x_k) - \bar{\alpha}\|\nabla f(x_k)\|^2 + \frac{L\bar{\alpha}^2}{2}\|\nabla f(x_k)\|^2 \\ &= f(x_k) + (\frac{L\bar{\alpha}^2}{2} - \bar{\alpha})\|\nabla f(x_k)\|^2 \end{aligned}$$

Since we chose $\bar{\alpha} < \frac{1}{L}$ leads to $\frac{L\bar{\alpha}^2}{2} - \bar{\alpha} = \bar{\alpha}(\frac{L\bar{\alpha}}{2} - 1) \leq \frac{1}{L}(\frac{1}{2} - 1) < 0$. Because the norm is non negative it follows that the sequence $(f(x_k))_{k\in\mathbb{N}}$ is decreasing. Let x_* be global minimum of f then using convexity it holds that

$$\forall k \in \mathbb{N} : \quad \underbrace{f(x_k) + (x_* - x_k)^T\nabla f(x_k)}_{\text{tangent at } x_*} \leq f(x_*)$$

This is saying that the tangent at the minimum of f is below the graph of f. If we plug in an equivalalent inequality as the above $f(x_k) \leq f(x_*) - (x_* - x_k)^T \nabla f(x_k)$ and then using the polarisation formula $-\langle a, b\rangle = \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2 - \frac{1}{2}\|a+b\|^2$, we get that

$$\begin{aligned}
f(x_{k+1}) &\leq f(x_k) + \bar{\alpha}(\frac{L\bar{\alpha}}{2} - 1)\|\nabla f(x_k)\|^2 \\
&\leq f(x_*) - \frac{\bar{\alpha}}{\bar{\alpha}}(x_* - x_k)^T \nabla f(x_k) + \bar{\alpha}(\frac{L\bar{\alpha}}{2} - 1)\|\nabla f(x_k)\|^2 \\
&= f(x_*) - \frac{1}{\bar{\alpha}}\bar{\alpha}\langle (x_* - x_k), \nabla f(x_k)\rangle + \bar{\alpha}(\frac{L\bar{\alpha}}{2} - 1)\|\nabla f(x_k)\|^2 \\
&= f(x_*) - \frac{1}{\bar{\alpha}}(\frac{1}{2}\|x_* - x_k\|^2 + \frac{1}{2}\|\bar{\alpha}\nabla f(x_k)\|^2 - \frac{1}{2}\|x_* - x_k + \bar{\alpha}\nabla f(x_k)\|^2) + \bar{\alpha}(\frac{L\bar{\alpha}}{2} - 1)\|\nabla f(x_k)\|^2 \\
&= f(x_*) - \frac{1}{\bar{\alpha}}(\frac{1}{2}\|x_* - x_k\|^2 + \frac{1}{2}\|\bar{\alpha}\nabla f(x_k)\|^2 - \frac{1}{2}\|x_* - x_{k+1}\|^2) + \bar{\alpha}(\frac{L\bar{\alpha}}{2} - 1)\|\nabla f(x_k)\|^2 \\
&= f(x_*) - \frac{1}{\bar{\alpha}}(\frac{1}{2}\|x_* - x_k\|^2 - \frac{1}{2}\|x_* - x_{k+1}\|^2) + \bar{\alpha}\underbrace{(\frac{L\bar{\alpha}}{2} - 1 - \frac{1}{2})}_{\leq 0; \bar{\alpha} \leq \frac{1}{L}}\|\nabla f(x_k)\|^2 \\
&\leq f(x_*) - \frac{1}{2\bar{\alpha}}(\|x_* - x_k\|^2 - \|x_* - x_{k+1}\|^2)
\end{aligned}$$

The above is equivalent to

$$f(x_{k+1}) - f(x_*) \leq -\frac{1}{2\bar{\alpha}}(\|x_* - x_k\|^2 - \|x_* - x_{k+1}\|^2) \qquad (*)$$

Now using that $(f(x_k))_{k\in\mathbb{N}}$ is a decreasing sequence it follows that

$$\sum_{k=0}^{N} f(x_{k+1}) \geq \sum_{k=0}^{N} f(x_{N+1}) = (N+1) f(x_{N+1}),$$

which implies that

$$\begin{aligned}
f(x_{N+1}) - f(x_*) &\overset{above}{\leq} \frac{1}{N+1}\sum_{k=0}^{N}(f(x_{k+1}) - f(x_*)) \\
&\overset{(*)}{\leq} \frac{1}{N+1}\sum_{k=0}^{N}(\frac{1}{2\bar{\alpha}}(\|x_* - x_k\|^2 - \|x_* - x_{k+1}\|^2)) \\
&= \frac{1}{2\bar{\alpha}(N+1)}\sum_{k=0}^{N}(\|x_* - x_k\|^2 - \|x_* - x_{k+1}\|^2) \\
&\overset{TelescopingSum}{=} \frac{1}{2\bar{\alpha}(N+1)}(\|x_* - x_0\|^2 - \|x_* - x_{N+1}\|^2) \\
&\leq \frac{1}{2\bar{\alpha}(N+1)}\|x_* - x_0\|^2 =: \frac{c}{N+1}
\end{aligned}$$

□

This is sublinear convergence, because "linear" convergence satisfies

$$e(x_k) \leq c e(x_{k-1}) \leq ... \leq c^k e(x_0)$$

Thus $e(x_k) \in \mathcal{O}(c^k) = \mathcal{O}(e^{k\log(c)})$, but in the case of the theorem we only have $(e_k) \in \mathcal{O}(\frac{1}{k+1})$.
In order to unsure that our convex functions have a unique minimum we define a class of functions that have to "bend" in all coordinates, i.e. being stronger convex than linear.

Definition 0.6: μ-Convexity

We call a function $f : \mathbb{R}^d \to \mathbb{R}$ μ-strongly convex if

$$f(y) \geq \underbrace{f(x) + (y-x)^T \nabla f(x)}_{\text{tangent}} + \frac{\mu}{2}\|y-x\|^2$$

for all $x, y \in \mathbb{R}^d$.

Remark 0.2

To show: Every strongly convex function has a unique global minimum.

Now that we tightend the assumptions on the function f, we can show in the following that we can improve the convergence from sub linear to linear!

Theorem 0.7

Let $f : \mathbb{R}^d \to \mathbb{R}$ be $\mu-$strongly convex for some $\mu > 0$ and let it be L-smooth. Further, let the sequence $(x_k)_{k\in\mathbb{N}}$ be generated by

$$x_{k+1} := x_k - \bar{\alpha}\nabla f(x_k), \quad x_0 \in \mathbb{R}^d$$

with $\bar{\alpha} < \frac{1}{L}$. Then the sequence $(x_k)_{k\in\mathbb{N}}$ convergences linearly, i.e.

$$\exists c \in (0,1) : \quad e(x_k) := \|x_k - x_*\| \leq c^k \|x_0 - x_*\|, \quad k \in \mathbb{N},$$

where $x_* \in \mathbb{R}^d$ is the unique global minimum of f and the constant can be chosen as $c = \sqrt{\frac{1}{1+\mu\bar{\alpha}}}$.

Proof. Let $x_* \in \mathbb{R}^d$ be the unique global minimum of f, i.e. among other things it holds that $\nabla f(x_*) = 0$. Because f is $\mu-$strongly convex it is also convex and thus get the result from the last theorem that

$$f(x_{k+1}) - f(x_*) \leq \frac{1}{2\bar{\alpha}}(\|x_k - x_*\|^2 - \|x_{k+1} - x_*\|^2). \tag{5.1}$$

Now using the μ-strong convexity we also get that

$$\frac{\mu}{2}\|x_{k+1} - x_*\|^2 = \underbrace{\nabla f(x_*)^T}_{=0}(x_{k+1} - x_*) + \frac{\mu}{2}\|x_{k+1} - x_*\|^2 \leq f(x_{k+1}) - f(x_*). \tag{5.2}$$

Together this leads to

$$\begin{aligned}
(\frac{\mu}{2} + \frac{1}{2\bar{\alpha}})\|x_{k+1} - x_*\|^2 &\overset{5.2}{\leq} f(x_{k+1}) - f(x_*) + \frac{1}{2\bar{\alpha}}\|x_{k+1} - x_*\|^2 \\
&\overset{5.1}{\leq} \frac{1}{2\bar{\alpha}}(\|x_k - x_*\|^2 - \|x_{k+1} - x_*\|^2) + \frac{1}{2\bar{\alpha}}\|x_{k+1} - x_*\|^2 \\
&= \frac{1}{2\bar{\alpha}}\|x_k - x_*\|^2.
\end{aligned}$$

Thus the assertion follows with

$$c = \sqrt{\frac{1}{2\bar{\alpha}(\frac{\mu}{2} + \frac{1}{2\bar{\alpha}})}} = \sqrt{\frac{1}{\bar{\alpha}\mu + 1}}$$

and if and only if $c \in (0, 1)$. □

Due to

$$\|x_k - x_*\| \leq (\sqrt{\frac{1}{\frac{\mu}{L} + 1}})^k \|x_0 - x_*\| \leq (\sqrt{\frac{1}{\bar{\alpha}\mu + 1}})^k \|x_0 - x_*\| = c^k \|x_0 - x_*\| =$$

Thus if we choose $\bar{\alpha}$ close to $\frac{1}{L}$ we get faster convergence. Further, the speed of convergence is even better when L is small, which in turn makes the function f smoother. Finally, the factor μ also determines the speed of convergence. The larger μ is, the more convex the function is. This also increases the speed. Thus the factor

$$\sqrt{\frac{1}{\frac{\mu}{L} + 1}}$$

completely determines the theoretically possible speed of convergence.

Remark 0.3

Show that for the quadratic function $f(x) = x^T Ax$ the specturm of A describes L and μ. L is two times the largest singular value und μ is the two times smallest singular value. Thus the convergence is fast, if all Eigenvalues are similar. If the singular values differ a lot, we have the situation of narrow valleys, where the curvature is strong in one direction but not so strong in another.

0.3 Gradient descent for L-smooth functions with PL inequality

In most application for policy gradient method, the assumption of convexity is not fulfilled. The following definition finds a way to relax this assumption.
The estimates of the norm of the gradient as in the following definition are called gradient domination inequalities.

Definition 0.8: PL-Inequality

A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the strong PL inequality if

$$\exists C > 0 \forall x \in \mathbb{R}^d : \quad \|\nabla f(x)\|^2 \geq C(f(x) - f_*),$$

and $f_* := \min_{x \in \mathbb{R}^d} f(x) > -\infty$.

In fact, the PL inequality was constructed in such a way, that we get the convergence of gradient descent really easily.

Theorem 0.9

Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and satisfies the PL inequality for $C = 2r$, where $r \in (0, L)$. Then the sequence $(x_k)_{k \in \mathbb{N}}$ generated by

$$x_{k+1} = x_k - \bar{\alpha}\nabla f(x_k), \quad x_0 \in \mathbb{R}^d,$$

with $\bar{\alpha} = \frac{1}{L}$, then the sequence $(f(x_k))_{k\in\mathbb{N}}$ converges linearly to f_*, i.e.

$$e(x_k) := f(x_k) - f_* \leq c^k(f(x_0) - f_*), \quad c = 1 - \frac{r}{L} \in (0,1).$$

Proof. Using the descent Lemma we get that

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) - \bar{\alpha}(1 - \frac{L\bar{\alpha}}{2})\|\nabla f(x_k)\|^2 \overset{\bar{\alpha}=\frac{1}{L}}{=} f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2 \\ &\overset{PL-Inequality}{\leq} f(x_k) - \frac{1}{2L}2r(f(x_k) - f_*) \\ &= f(x_k) - \frac{r}{L}(f(x_k) - f_*). \end{aligned}$$

Now subtracting f_* from both sides yields

$$f(x_{k+1}) - f_* \leq f(x_k) - \frac{r}{L}(f(x_k) - f_*) - f_* = (1 - \frac{r}{L})(f(x_k) - f_*).$$

The claim follows by induction. □

The idea behind the PL inequality is that it ensures that the gradient does not vanish as long as the gradient descent algotithm has not reached f_*. It further implies that that a small gradient norm at x implies that there is a small difference of $f(x)$ and f_*.

Remark 0.4

Prove that $\mu-$strong covnexity and L-smoothness imply the PL-inequality. What is C?

If we relax the condition of the PL inequlity even further, we get sublinear convergence:

Definition 0.10: Weak PL inequality

A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the weak PL inequality if

$$\exists C > 0 \forall x \in \mathbb{R}^d : \quad \|\nabla f(x)\| \geq C(f(x) - f_*),$$

and $f_* := \min_{x\in\mathbb{R}^d} f(x) > -\infty$.

Remark 0.5

We will later see that for a specific parametrisation of policies the weak PL inequality holds for the parametrised value function!

Theorem 0.11

Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and satisfies the PL weak inequality for $C = 2r$, where $f \in (0, L)$. Then the sequence $(x_k)_{k\in\mathbb{N}}$ generated by

$$x_{k+1} = x_k - \bar{\alpha}\nabla f(x_k), \quad x_0 \in \mathbb{R}^d,$$

with $\bar{\alpha} = \frac{1}{L}$, then the sequence $(f(x_k))_{k\in\mathbb{N}}$ converges sub-linearly to f_*, i.e.

$$e(x_k) := f(x_k) - f_* \leq \frac{L}{2r^2(k+1)}.$$

Proof. As in the previous proof using the descent Lemma with L-smoothness we get that

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2.$$

Now applying the weak PL-inequality for the gradient yields

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}4r^2(f(x_k) - f_*)^2. \tag{5.3}$$

Subtracting on both sides with f_* leads to

$$e(x_{k+1}) = f(x_{k+1}) - f_* \overset{5.3}{\leq} f(x_k) - \frac{1}{L}2r^2(f(x_k) - f_*)^2 - f_* = e(x_k) - \frac{1}{L}2r^2 e(x_k)^2 = \left(1 - \underbrace{\frac{1}{L}2r^2}_{=:q} e(x_k)\right) e(x_k)$$

To show: Any positive sequence $(a_n)_{n\in\mathbb{N}}$ with $a_n \in [0, \frac{1}{q}]$ for some $q > 0$ that satisfies the diminishing contraction:

$$\forall n \geq 0: \quad 0 \leq a_{n+1} \leq (1 - qa_n)a_n$$

converges to zero with convergence rate

$$a_n \leq \frac{1}{nq + \frac{1}{a_0}} \overset{\text{Def. of } a_n}{\leq} \frac{1}{(n+1)q}.$$

We only need to show the first inequality.
We reformulate the definied contraction

$$a_{n+1} \leq (1 - qa_n)a_n \quad \iff \quad a_{n+1} + qa_n^2 \leq a_n$$

and divide by $a_n a_{n+1}$ and obtaining

$$\frac{1}{a_{n+1}} \leq \frac{1}{a_n} + q \qquad \underbrace{\frac{a_n}{a_{n+1}}}_{\geq 1, \text{because } \frac{a_{n+1}}{a_n} \leq (1-qa_n) \leq 1-0} \leq \frac{1}{a_n} + q.$$

Now using the telescoping sum leads to

$$\frac{1}{a_n} - \frac{1}{a_0} = \sum_{k=0}^{n-1}\left(\frac{1}{a_{k+1}} - \frac{1}{a_k}\right) \geq \sum_{k=0}^{n-1} q = nq.$$

We can now rewrite this as

$$a_n \leq \frac{1}{nq + \frac{1}{a_0}}.$$

All we need to do now is to show that our sequence $e(x_k) \in [0, \frac{1}{2r^2/L}]$. Positivity is trivial. It remains to show that $e(x_k) \leq \frac{L}{2r^2}$ for all $k \in \mathbb{N}$.
In the first part of this proof we got the result that

$$f_* \leq f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2$$

which implies that

$$f_* - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2 \quad \iff \quad e(x_k) \geq \frac{1}{2L}\|\nabla f(x_k)\|^2 \geq \frac{1}{2L}4r^2 e(x_k)^2$$

which is equivalent to

$$\frac{1}{e(x_k)} \geq \frac{2r^2}{L} \quad \iff \quad e(x_k) \leq \frac{L}{2r^2}.$$

□

There are L-smooth functions that are $L-$smooth but are not convex.

Remark 0.6

Show that the function $f(x) = x^2 + 3\sin(x)^2$ satisfies the weak PL inequality.

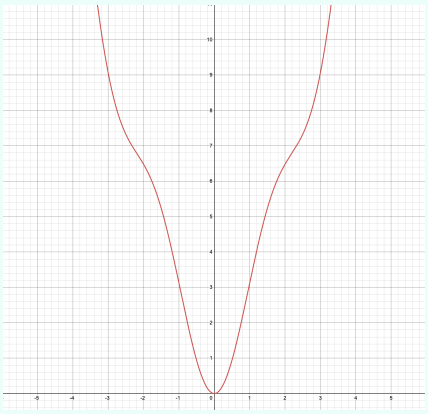


Figure 5.1: Non Convex function that satisfies the PL-inequality

This plot can help to find the parameter r of the PL condition. This is due to the fact that the condition

$$\|\nabla f(x)\| \geq 2r(f(x) - f_*)$$

gives a lower bound of the slope in dependence of the L-smoothness point wise, i.e. f can fall/increase faster than that lower bound. If $f(x_k)$ is closer to f_* then this lower bound decreases.

0.4 Gradient descent with diminishing step-sizes

Typically the L-smoothness parameter is unknow. Thus we cannot choose the step size as $\bar{\alpha} = \frac{1}{L}$ or smaller, because we do not know it. Thus we want to choose a step size, that diminishes, such that it will be eventually smaller than $\frac{1}{L}$, but not too fast such that the sequence will stop to early.

Theorem 0.12

Let $f : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and the sequence $(x_k)_{k\in\mathbb{N}}$ defined by

$$x_{k+1} := x_k - \alpha_k \nabla f(x_k), \quad x_0 \in \mathbb{R}^d,$$

with non-negative step-size sequence $(\alpha_k)_{k\in\mathbb{N}}$ that satisfies

$$\lim_{k\to\infty} \alpha_k = 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \alpha_k = \infty.$$

Then the sequence $(f(x_k))_{k\in\mathbb{N}}$ satisfies

$$\lim_{k\to\infty} f(x_k) = -\infty \quad \text{or} \quad \lim_{k\to\infty} \nabla f(x_k) = 0.$$

Moreover it holds that every accumulation point $\lim_{k\to\infty} x_k = \bar{x}$ is a stationary point of f, i.e. $\nabla f(\bar{x}) = 0$.

Proof. Again using the descent Lemma we obtain

$$f(x_{k+1}) \leq f(x_k) + (\frac{L\alpha_k}{2} - 1)\|\nabla f(x_k)\|^2.$$

Since the sequence $\lim_{k\to\infty} a_k = 0$ there exsists a $k_0 \in \mathbb{N}$ such that

$$\forall k \geq k_0 : \quad f(x_{k+1}) \leq f(x_k) + \underbrace{(\frac{L\alpha_k}{2} - 1)}_{\leq 0}\|\nabla f(x_k)\|^2.$$

i.e. the sequence $(f(x_k))_{k\geq k_0}$ is decreasing. Therefore we have two cases:

$$\lim_{k\to\infty} f(x_k) = \begin{cases} M \in \mathbb{R} \\ -\infty \end{cases}$$

Case 1: $\lim_{k\to\infty} f(x_k) = M$
It holds that

$$\forall K > k_0 : \quad \sum_{k=k_0}^{K} \frac{\alpha_k}{2}\|\nabla f(x_k)\|^2 \overset{above}{\leq} \sum_{k=k_0}^{K} (f(x_k) - f(x_{k+1})) \overset{\text{telescoping sum}}{=} f(x_{k_0}) - f(x_K)$$

If we let $K \to \infty$ we get that

$$\sum_{k=k_0}^{\infty} \frac{\alpha_k}{2}\|\nabla f(x_k)\|^2 \leq f(x_{k_0}) - M < \infty.$$

Since we required the step size to satisfy $\sum_{k=0}^{\infty} \alpha_k = \infty$ it follows that

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$$

In order to prove that the limit goes to zero of the above we only need to show that the limsup has the same limit, i.e.

$$\limsup_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Suppose that $\exists \epsilon > 0$ such that $\limsup_{k\to\infty} \|\nabla f(x_k)\| \geq \epsilon$. Further, consider two sub-sequences $(m_k)_{j\in\mathbb{N}}$ and $(n_k)_{j\in\mathbb{N}}$, that satisfy

$$\forall j \in \mathbb{N}: \quad m_j < n_j < m_{j+1}$$

and when the sequence of $(\|\nabla f(x_k)\|)_{k\in\mathbb{N}}$ is for $m_j \leq k < n_j$, i.e. in the index between the two subsequnces the j^{th} time, then it satisfies

$$\frac{\epsilon}{3} < \|\nabla f(x_k)\|, \quad m_j \leq k < n_j$$

and if this sequence satisfies in the indices that $m_{j+1} \leq k < n_1$,i.e. the index being between the two subsequences for the $j + 1^{th}$ time then is satsifies

$$\frac{\epsilon}{3} \geq \|\nabla f(x_k)\|, \quad n_j \leq k < m_{j+1}.$$

These sequences exists, due to the fact that only $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$ and the limsup has to be larger. Now choose $\bar{j} \in \mathbb{N}$ sufficiently large such that

$$\sum_{k=m_{\bar{j}}}^{\infty} \alpha \|\nabla f(x_k)\|^2 \leq \frac{\epsilon^2}{9L}.$$

Now we can use L-smoothness for all $j \geq \bar{j}$ and $m_j \leq m \leq n_j - 1$ such that we are in the area where $m_j \leq k < n_j$:

$$\begin{aligned}
\|\nabla f(x_{n_j}) - \nabla f(x_m)\| &\overset{\text{telescoping Sum}}{=} \|\nabla \sum_{k=m}^{n_j-1} f(x_{k+1}) - \nabla f(x_k)\| \\
&\leq \sum_{k=m}^{n_j-1} \|\nabla f(x_{k+1}) - \nabla f(x_k)\| \\
&\leq L \sum_{k=m}^{n_j-1} \|x_{k+1} - x_k\| \\
&\leq \frac{3\epsilon}{3\epsilon} L \sum_{k=m}^{n_j-1} \|x_k + \alpha_k \nabla f(x_k) - x_k\| \\
&\leq \frac{3}{\epsilon} L \sum_{k=m}^{n_j-1} \frac{\epsilon}{3} \alpha_k \|\nabla f(x_k)\| \\
&\leq \frac{3}{\epsilon} L \sum_{k=m}^{n_j-1} \alpha_k \|\nabla f(x_k)\|^2 \\
&= L \frac{3}{\epsilon} \frac{\epsilon^2}{9L} = \epsilon/3
\end{aligned}$$

Still using this m we het that

$$\|\nabla f(x_m)\| \leq \|\nabla f(x_{n_j})\| + \|\nabla f(x_{n_j}) - \nabla f(x_m)\| \overset{above}{\leq} |\nabla f(x_{n_j})\| + \frac{\epsilon}{3} \leq \frac{2\epsilon}{3}.$$

This implies that $\forall m \geq m_{\bar{j}}$ it holds that $\|\nabla f(x_m)\| \leq \frac{2\epsilon}{3}$???!!! This is a contradiction to $\limsup_{k\to\infty} \|\nabla f(x_k)\| \geq \epsilon$. Therefore

$$\limsup_{k\to\infty} \|\nabla f(x_k)\| = \liminf_{k\to\infty} \|\nabla f(x_k)\| = \lim_{k\to\infty} \|\nabla f(x_k)\| = 0.$$

Finally, for a accumulation point $\bar{x}$ of $(x_k)_{k\in\mathbb{N}}$ it holds that

$$\nabla f(\bar{x}) = \lim_{k\to\infty} \nabla f(x_k) = 0$$

because $(f(x_k))_{k\geq k_0}$ is decreasing. □

Choosing the step size right is the essential thing in this approach! Consider $f(x) = \|x\|^2$, then $\nabla f(x,y) = (2x, 2y)$, i.e. if we choose the step size to be $\alpha = \frac{1}{L} = \frac{1}{2}$, then we converge in just one step. If we choose it smaller, we need more steps. This can get arbitrarly bad, if the step size is wrong!

BILD???

0.5 Stochastic gradient descent

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be the underlying probability space and $Z : \Omega \times \mathbb{R}^d \to \mathbb{R}^p$ be a random variable for every $x \in \mathbb{R}$ with distribution μ_x on this probability space. We define the cost function

$$F : \mathbb{R}^d \to \mathbb{R}, \quad x \mapsto F(x) := \mathbb{E}_{Z\sim\mu_x}[f(x, Z)] = \int_{\mathbb{R}^p} f(x, z)\mu_x(dz)$$

where $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$. We now want to optimize the

$$\min_{x\in\mathbb{R}^d} F(x).$$

Them method of getting a sample $Z \sim \mu_x$ is often called oracle. We assume that we can repeatedly ask the oracle for samples. Here are standard assumptions that one upholds

- The function $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$ is $\mathcal{B}(\mathbb{R}^d) \otimes \mathcal{B}(\mathbb{R}^p) - \mathcal{B}(\mathbb{R})$ measurable.
- For every $z \in \mathbb{R}^p$ the function $x \mapsto f(x, z)$ is vontinuously differentiable
- For every $x \in \mathbb{R}^d$ it holds that

$$\mathbb{E}_{Z\sim\mu_x}[|f(x, Z)| + \|\nabla_x f(x, Z)\|] < \infty.$$

Algorithm 16: Plain vanilla stochastic gradient descent method (SGD)

Data : Initial random variable $X_0 : \Omega \to \mathbb{R}^d$
Result: Approximation of a stationary point
Set $k = 0$;
while *not converged* **do**
  Determine α_k;
  Sample Z_{k+1} from μ_{X_k};
  Set $X_{k+1} = X_k - \alpha_k \nabla_x f(X_k, Z_{k+1})$;
  $k \leftarrow k + 1$;

We can now rewrite the iteration scheme as

$$X_{k+1} = X_k - \alpha_k \nabla_x f(X_k, Z_{k+1}) = X_k - \alpha_k(\nabla F(x) + \underbrace{\nabla_x f(X_k, Z_{k+1}) - \nabla F(x)}_{=:\epsilon_k}) = X_k - \alpha_k(\nabla F(x) + \epsilon_k)$$

which looks exactly as our stochastic approximation scheme with unbiased error ϵ_k. Thus this iteration scheme is ought to converge to zero if the error term of $\nabla F(x)$ is sufficiently well behaved and we have some

assumptions on the step size.
We already had a theorem directly after the descent Lemma where we had a similiar iteration scheme but only for a deterministic sequence. The following will expand this.
Further, we do not have strong assumptions of F in the following, thus the convergence is weak. There are also theorems, that do SGD for stronger assumptions on F and thus get better convergence results.
Important point: Becuase the errors are only bounded, but to not tend to zero, the step size has to go to zero to ensure stability when the fixed point is reached.

Theorem 0.13: SGD almoust sure convergence for L-smooth functions

Let $F : \mathbb{R}^d \to \mathbb{R}$ be L-smooth and $F_* := \inf_{x\in\mathbb{R}^d} F(x) > -\infty$ and satisfying

$$\mathbb{E}_{Z\sim\mu_x}[\|\nabla_x f(x,Z) - \mathbb{E}[\nabla_x f(x,Z)]\|^2] \leq c(1 + F(x) - F_*).$$

Suppose that the step sizes $(\alpha_k)_{k\in\mathbb{N}}$ are non negative and either deterministic or adapted to the SGD scheme and satisfying the Robbins-Monro conditions

$$\sum_{k=0}^{\infty} \alpha_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty \quad a.s.$$

Moreover, let X_0 be a randum variable that satsifies $\mathbb{E}[F(X_0)] < \infty$ and

$$X_{k+1} := X_k - \alpha_k \nabla_x f(X_k, Z_{k+1}), \quad Z_{k+1} \sim \mu_{X_k}.$$

Then

$$\lim_{k\to\infty} F(X_k) = F_\infty < \infty \quad a.s.$$

and

$$\lim_{k\to\infty} \|F(X_k)\|^2 = 0 \quad a.s.$$

Proof. We assume that we can interchange gradient and the expectation (DCT):

$$\mathbb{E}_{Z\sim\mu_x}[\nabla_x f(x,Z)] = \nabla_x \mathbb{E}_{Z\sim\mu_x}[f(x,Z)] = \nabla_x F(x).$$

Define the natural filtration $\mathcal{F} := (\mathcal{F}_k)_{k\in\mathbb{N}}$ trough $\mathcal{F}_k = \sigma(X_m, m \leq k) = \sigma(X_0, Z_m, m \leq k)$ and note that $(\alpha_k)_{k\in\mathbb{N}}$ is $\mathcal{F}$-adapted by construction??? Now using the descent Lemma and the polatisation formula

$\langle x, y\rangle = \frac{1}{2}(\|x\|^2 + \|y\|^2 + \|x-y\|^2)$ we get pathwise that

$$
\begin{aligned}
F(X_{k+1}) &= F(X_k - \alpha_k \nabla_x f(X_k, Z_{k+1})) \\
&\overset{\text{Descent Lemma}}{\leq} F(X_k) - \alpha_k \langle \nabla_x f(X_k, Z_{k+1}), \nabla_x F(X_k)\rangle + \alpha_k^2 \frac{L}{2}\|\nabla_x f(X_k, Z_{k+1})\|^2 \\
&= F(X_k) - \frac{\alpha_k}{2}\left(\|\nabla_x F(X_k)\|^2 + \|\nabla_x f(X_k, Z_{k+1})\|^2 + \|\nabla_x F(X_k) - \nabla_x f(X_k, Z_{k+1})\|^2\right) + \alpha_k^2 \frac{L}{2}\|\nabla_x f(X_k, Z_{k+} \\
&= F(X_k) - \frac{\alpha_k}{2}\left(\|\nabla_x F(X_k)\|^2 + \|\epsilon_k\|^2\right) + (\alpha_k^2 \frac{L}{2} - \frac{\alpha_k}{2})\|\nabla_x f(X_k, Z_{k+1})\|^2 ??? \\
&= F(X_k) - \frac{\alpha_k}{2}\left(\|\nabla_x F(X_k)\|^2 + \|\epsilon_k\|^2\right) + \frac{\alpha_k}{2}(\alpha_k L - 1)\|\nabla_x f(X_k, Z_{k+1})\|^2 \\
&= ??? \\
&= F(X_k) + (\frac{\alpha_k^2 L}{2} - \alpha_k)(\langle \nabla F(X_k), \nabla_x f(X_k, Z_{k+1})\rangle - \|\nabla_x f(X_k, Z_{k+1})\|^2 - \|\nabla F(X_k)\|^2) + (\alpha_k - \frac{\alpha_k^2 L}{2})\langle \nabla_x \\
&= F(X_k) + (\frac{\alpha_k^2 L}{2} - \alpha_k)\|\nabla F(X_k) - \nabla_x f(X_k, Z_{k+1})\|^2 + (\alpha_k - \frac{\alpha_k^2 L}{2})\langle \nabla_x F(X_k), -\nabla_x f(X_k, Z_{k+1})\rangle \\
&= F(X_k) + (\frac{\alpha_k^2 L}{2} - \alpha_k)\|\epsilon_k\|^2 + (\alpha_k - \frac{\alpha_k^2 L}{2})\langle \nabla_x F(X_k), -\nabla_x f(X_k, Z_{k+1})\rangle \\
&= F(X_k) + (\frac{\alpha_k^2 L}{2} - \alpha_k)(\|\nabla F(X_k)\|^2 + \|\epsilon_k\|^2) + (\alpha_k - \frac{\alpha_k^2 L}{2})(\|\nabla_x F(X_k)\|^2 + \langle \nabla_x F(X_k), -\nabla_x f(X_k, Z_{k+1})\rangle \\
&= F(X_k) + (\frac{\alpha_k^2 L}{2} - \alpha_k)(\|\nabla F(X_k)\|^2 + \|\epsilon_k\|^2) + (\alpha_k - \frac{\alpha_k^2 L}{2})\langle \nabla_x F(X_k), \nabla F(X_k) - \nabla_x f(X_k, Z_{k+1})\rangle \\
&= F(X_k) + (\frac{\alpha_k^2 L}{2} - \alpha_k)(\|\nabla F(X_k)\|^2 + \|\epsilon_k\|^2) + (\alpha_k - \frac{\alpha_k^2 L}{2})\langle \nabla_x F(X_k), \epsilon_k\rangle \\
&= F(X_k) - \alpha_k \|\nabla F(X_k)\|^2 + \alpha_k \langle \nabla_x F(X_k), \epsilon_k\rangle + \frac{\alpha_k^2 L}{2}\left(\|\nabla_x F(X_k)\|^2 - \langle \nabla F(X_k), \epsilon_k\rangle + \|\epsilon_k\|^2\right)
\end{aligned}
$$

where $\epsilon_k := \nabla F(X_k) - \nabla_x f(X_k, Z_{k+1})$. Now using the fact that we can pull out the gradient we get that

$$\mathbb{E}[\epsilon_k \mid \mathcal{F}_k] = \nabla F(X_k) - \nabla \mathbb{E}_{Z\sim\mu_{X_k}}[f(X_k, Z_{k+1}] = 0$$

and using the assumptions we get that

$$\mathbb{E}[\|\epsilon_k\|^2 \mid \mathcal{F}_k] \leq c(1 + F(X_k) - F_*).$$

Together this fives us the assumptions the Robins Monro theorem

$$
\begin{aligned}
\mathbb{E}[\underbrace{F(X_{k+1}) - F_*}_{=:\tilde{Z}_{k+1}} \mid \mathcal{F}_k] &= \mathbb{E}[F(X_{k+1}) \mid \mathcal{F}_k] - F_* \\
&\leq \mathbb{E}[F(X_k) - \alpha_k \|\nabla F(X_k)\|^2 + \alpha_k \langle \nabla_x F(X_k), \epsilon_k\rangle + \frac{\alpha_k^2 L}{2}\left(\|\nabla_x F(X_k)\|^2 - \langle \nabla F(X_k), \epsilon_k\rangle + \|\epsilon_k\| \\
&= (F(X_k) - F_*) - \alpha_k \|\nabla F(X_k)\|^2 + (\alpha_k - \frac{\alpha_k^2 L}{2})\underbrace{\mathbb{E}[\langle \nabla_x F(X_k), \epsilon_k\rangle \mid \mathcal{F}_k]}_{=0} + \frac{\alpha_k^2 L}{2}\left(\|\nabla_x F(X_k)\|^2 + \right. \\
&\leq (F(X_k) - F_*) - \alpha_k \|\nabla F(X_k)\|^2 + \frac{\alpha_k^2 L}{2}\left(\|\nabla_x F(X_k)\|^2 + c(1 + F(X_k) - F_*)\right) \\
&= (1 + c\frac{\alpha_k^2 L}{2})\underbrace{(F(X_k) - F_*)}_{=:\tilde{Z}_k} + \underbrace{c\frac{\alpha_k^2 L}{2}}_{=:C_k} + \underbrace{(\frac{\alpha_k^2 L}{2} - \alpha_k)\|\nabla_x F(X_k)\|^2}_{=:B_k}
\end{aligned}
$$

Now we need to show that these sequences satisfy the conditions from Robins Siegmund.
Here we used that the sequence $(\tilde{Z}_k)_{k\in\mathbb{N}}$ is non-negative, because F is bounded by below. CHOICE Of α_k???
□

We now take a completely different view on optimal control problems. The theory in the following is more complex and less understood. Thus the following will be less complete. We start with a policy gradient method, where we solve the optimal control problem numerically for the value function. But the difference is that we do not do this for the entire vector, but only for specific $V(s)$. This is because, not every state has the same importance in practice!

Remark 0.7

In Policy Gradient, we do not parametrize the Value function! Instead we only parametrize the policy. Fix a subclass $\Pi^\Theta \subseteq \Pi_S$ of parametrized stationary policies π^θ that hopefully contains the optimal policy. The goal is then to maximize for an initial distribution μ on $\mathcal{S}$ the mapping

$$\theta \mapsto J_\mu(\theta) := V^{\pi^\theta}(\mu) := \sum_{s\in\mathcal{S}} \mu(s) V^{\pi^\theta}(s),$$

where $V^{\pi^\theta}(s) = \mathbb{E}_s^{\pi^\theta}[\sum_{t=1}^T R_t]$, for $T \sim Geo(1-\gamma)$ or fixed (we look at both infinite and finite MDPs), using numerical methods. We will denote the best policy $\pi^{\theta^*} \approx \pi^*$. Due to the fact that

$$\nabla J_\mu = \sum_{s\in\mathcal{S}} \mu(s) \nabla J_s$$

we will only look at the formulas for $J(\theta) := V^{\pi^\theta}(s)$.

We now want to answer the following questions

- How to set up an algorithm and how to compute the gradient?

  -> Gradient estimates can be obtained in a model free way (Policy gradient theorems)

- How to choose Π^Θ such that $\pi^* \in \Pi^\Theta$?

  -> Trade off of choosing large and small families. Not much is known in practical applications, i.e. Neural Networks and hope.

- How to initialize?

Definition 0.14

Let $\Theta \subset \mathbb{R}^d$ and $\{\pi^\theta \mid \theta \in \Theta\}$ a set of stationary policies such that $\theta \mapsto \pi^\theta(a;s)$ is differentiable in θ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Then we call this mapping a differentiable parametrised family of stationary policies.

Later we will discuss policies that are neural networks which have to be large enough to hopefully contain the optimal policiy, but at the same time small enough to be numerically feasible.

Remark 0.8

If $\Theta = \mathbb{R}^d$ then we have to choose $d < |\mathcal{A}||\mathcal{S}|$ in order to keep Θ from exploding in size.

Example 0.15

Suppose $d = |\mathcal{S}||\mathcal{A}|$ such that every state action pair has its own coordinate. Denote by $\theta = (\theta_{s,a})_{s\in\mathcal{S},a\in\mathcal{A}}$ the parameters for the parametrized family. Direct parametrization is given by

$$\pi^\theta(a;s) := \theta_{s,a}, \quad \sum_a \theta_{s,a} = 1, \theta_{s,a} \geq 0.$$

or define the softmax parametrization

$$\pi^\theta(a;s) = \frac{e^{\theta_{s,a}}}{\sum_{a'\in\mathcal{A}_s} e^{\theta_{s,a'}}}.$$

The problem with these two parametrization is that they are to large and contain to much policies. Instead of the tabular parameterization as above, we can also define the generalized softmax policy by any differentiable function $\theta \mapsto h_\theta(s,a)$ such that

$$\pi^\theta(a;s) = \frac{e^{h_\theta(s,a)}}{\sum_{a'\in\mathcal{A}_s} e^{h_\theta(s,a')}}.$$

If for instance $h_\theta(s,a) := \theta^T\Phi(s,a)$ then we call it linear softmax with features $\Phi(s,a)$. If $\Phi(s,a) = \theta_{s,a}\mathbf{1}_{s,a}$, i.e. the unit vector, we have the tabular softmax. If $\Phi(s,a)$ builds a different kind of basis for $\mathbb{R}^d$ it is no longer a tabular policy. The feature vector is supposed to reduce complexity, thus $\Phi(s,a)$ should build a basis of a space that is small enough to be computationally feasible but at the same time large enough to separate state action pairs sufficiently.
An example of a generalized softmax policy is, when we take a neural network where in the last layer we have a softmax activation function. Usually the number of weights is lower than $d = |\mathcal{A}| \cdot |\mathcal{S}|$.???

In the following we make sense of the following gradient ascent algorithm assuming we can calculate $V^{\pi^\theta}(s)$ explicitly.

Algorithm 17: Plain vanilla policy gradient algorithm (exact gradients)

Data : Initial parameter θ_0
Result: Approximate policy $\pi^\theta \approx \pi^{\theta^*}$
Set $n = 0$;
while *not converged* **do**
  Choose step-size α;
  Compute $K = \nabla J(\theta)\big|_{\theta=\theta_n}$;
  Update $\theta_{n+1} = \theta_n - \alpha K$;
  $n \leftarrow n + 1$;
return π^{θ_n}

There are a lot of problems:

- There is no reason to assume that $\theta \mapsto J(\theta)$ satisfies typical concavity conditions. Therefore it is very likely that we get stuck in a local stationary point.

- The gradient $\nabla J(\theta)$ is typically unknown. How could one impliment this method without access to the gradient?
- This method is really slow! How can we speed up the algorithm?

The first two problems can be partially resolved.
We will first focus on classical results for gradient ascent. We will show that the right set up for this method is the PL inequality and L-smoothness.
Next we we discuss how we can combine the policy gradient theorems with neural netweork parametrisations.
Finally, we will consider some modifiactions that are used in practice.

1 Policy gradient theorems

Our first result will be that we can express the gradient in a rather nice way such that we can approximate it in a model free way as an expectation, that does not involve the transition probabilities and thus can be approximated by samples only!
For didactic reasons we first only deal with finite MDPs. Here the situation is not extremely realistic, because most finite-time MDPs do not have optimal strategies that are stationary (rember the last time step in the ice vendor example). But this is good starting point for the later discussed infinite time MDPs.

1.1 Finite-time MDPs

Here we assume that there is no reward in the last step, i.e.

$$J(\theta) = \mathbb{E}_{\mu}^{\pi^\theta}[\sum_{t=0}^{T-1} R_t].$$

In this didactic section we will derive three firrerent expressions for the gradient. In the following we write

$$R_t^T := \sum_{k=t}^{T-1} R_{k+1}$$

for the reward after time t up to time T which we call the reward to go. We call R_0^T the total non-discounted reward.
Remember that in bandits the gradient was of the form

$$\mathbb{E}[\nabla_\theta \log(\pi^\theta(A)) X_A].$$

Theorem 1.1

Assume that (S, A, R) is a T-step MDP with finite state action spaces and consider a stationary differentiable parameterized family of policies $\{\pi^\theta \mid \theta \in \Theta\}$. Then it holds that the gradient of the value function exists wrt. θ and is given by

$$\nabla_\theta J_s(\theta) = \mathbb{E}_s^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t)) R_0^T)].$$

Note that in the proof we used that

$$\mathbb{E}[Y] = \begin{pmatrix} \mathbb{E}[Y_1] \\ \cdots \\ \mathbb{E}[Y_n] \end{pmatrix}, \quad Y \in \mathbb{R}^n.$$

Proof. Consider a trajectory $\tau = (s_0, a_0, r_0, ..., r_{T-1}, s_{T-1}, r_{T-1}, s_T, r_T)$ of the $T-$step MDP and its path probability without a_T as

$$\begin{aligned}\mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau) = \delta_s(s_0)\pi^\theta(a_0; s_0) \prod_{t=0}^{T-2} p(s_{i+1}, r_i; s_i, a_i)\pi^\theta(a_{i+1}; s_{i+1}) \\ \cdot p(s_T, r_{T-1}; s_{T-1}, a_{T-1})\pi^\theta(\mathcal{A}_{S_T}; s_T) \cdot p(\mathcal{S} \times \{r_T\}; s_T, \mathcal{A}_{s_T})\end{aligned}$$

and define the set of all trajectories for T steps as

$$\mathcal{T} := \{\tau = (s_0, a_0, r_0, ..., s_{T-1}, a_{T-1}, r_{T-1}, s_T, r_T) \mid \forall t \leq T : r_t \in \mathcal{R}, s_t \in \mathcal{S} \text{ and } \forall t < T : a_t \in \mathcal{A}_{s_t}\}.$$

Thus we can write

$$J_s(\theta) = V^{\pi^\theta}(s) = \mathbb{E}_s^{\pi^\theta}[\sum_{t=0}^{T-1} R_{t+1}] = \sum_{\tau \in \mathcal{T}} \mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau) \sum_{t=0}^{T-1} r_{t+1}$$

Because the probability over every trajectory is differentiable and we have a finite sum, this proves the differentiability.
Next using the log trick, we have that

$$\begin{aligned}&\nabla_\theta \mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau) \\ &= \nabla_\theta \mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau) \frac{\mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau)}{\mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau)} \\ &= \nabla_\theta \log(\mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau)) \mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau) \\ &= \nabla_\theta \Bigg(\log(\delta_s(s_0) + \log(\pi(a_0; s_0) + \sum_{t=0}^{T-1} (\log(p(s_{i+1}, r_i; s_i, a_i)) + \log(\pi^\theta(a_{i+1}; s_{i+1}))) + \log(p(s_T, r_{T-1}; s_{T-1}, a_{T-1})) + \text{l} \\ &\cdot \mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau) \\ &= \sum_{t=0}^{T-1} \nabla_\theta (\log(\pi^\theta(a_t, s_t)) \cdot \mathbb{P}_s^{\pi^\theta}((S,A,R) = \tau)\end{aligned}$$

Due to the log, the only summand that depemds on θ is the policy π^θ, thus all other summands dissapear. Due to the fact that the state, action and reward space are finite we can interchange the derivative with the

sum:

$$\begin{aligned}
\nabla_\theta J_s(\theta) &= \nabla_\theta V^{\pi^\theta}(s) \\
&= \sum_{\tau \in \mathcal{T}} \nabla_\theta \mathbb{P}_s^{\pi^\theta}((S,A,R)=\tau) \sum_{t=0}^{T-1} r_{t+1} \\
&= \sum_{\tau \in \mathcal{T}} \left(\sum_{t=0}^{T-1} \nabla_\theta (\log(\pi^\theta(a_t,s_t)) \cdot \mathbb{P}_s^{\pi^\theta}((S,A,R)=\tau) \right) \sum_{t=0}^{T-1} r_{t+1} \\
&= \mathbb{E}_s^{\pi^\theta}[(\sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(A_t,S_t)) \sum_{t=0}^{T-1} R_{t+1}] \\
&= \mathbb{E}_s^{\pi^\theta}[(\sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(A_t,S_t)) R_0^T]
\end{aligned}$$

□

Note that in this proof, the log trick made sure that we got an expectation again.
Problem: This estimator has a high variance (not only problem with length of the gradient but also the direction). Note that for every $s \in \mathcal{S}$ that $\nabla_\theta J_s(\theta)$ is a vector.

Definition 1.2: Score function

If π is a policy, then we define the vector $\nabla_\theta(\log(\pi^\theta(a;s)))$ as the score function of π^θ.

Similar as in the bandit chapter we can show the following

Example 1.3

Show that for the tabular softmax parametrization that

$$\frac{\partial \log(\pi^\theta(a;s))}{\partial \theta_{s',a'}} = \mathbf{1}_{s=s'} \cdot (\mathbf{1}_{a=a'} - \pi^\theta(a';s'))$$

and for the linear softmax parametrization with features $\Phi(s,a)$ it holds that

$$\nabla_\theta \log(\pi^\theta(a;s)) = \Phi(s,a) - \sum_{a' \in \mathcal{A}_s} \pi^\theta(a';s)\Phi(s,a').$$

In our theorem above, we have that the we multiply by the constant $R_0^T := \sum_{t=0}^{T-1} R_t$. In the next theorem we can show that we can we can rewirte the constant in a time depended manner.

Theorem 1.4

Assume that (S,A,R) is a T-step MDP with finite state action spaces and consider a stationary differentiable parameterized family of policies $\{\pi^\theta \mid \theta \in \Theta\}$. Then it holds that the gradient of the value

function exists wrt. θ and is given by

$$\nabla_\theta J_s(\theta) = \mathbb{E}_s^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t))R_t^T)].$$

Proof. From the Last theorem we have that

$$\nabla_\theta J_s(\theta) = \mathbb{E}_s^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t))R_0^T)] = \sum_{t'=0}^{T-1}\sum_{t^*=0}^{T-1} \mathbb{E}_s^{\pi\theta}[\nabla_\theta(\log(\pi^\theta(A_{t'}; S_{t'}))R_{t^*})]$$

We can now show that all summands with $t^* < t'$ fall away, i.e.

$$\begin{aligned}
&\mathbb{E}_s^{\pi\theta}[\nabla_\theta(\log(\pi^\theta(A_{t'}; S_{t'}))R_{t^*})] \\
&= \sum_{\tau\in\mathcal{T}} \mathbb{P}_s^\pi((S,A,R)=\tau)\nabla_\theta(\log(\pi^\theta(a_{t'}; s_{t'}))r_{t^*} \\
&= \sum_{a_0}\sum_{s_1}\sum_{r_1}\cdots\sum_{s_T}\sum_{a_T}\sum_{r_T} \delta_s(s_0)\pi^\theta(a_0; s_0)\prod_{t=0}^{T-2} p(s_{i+1}, r_i; s_i, a_i)\pi^\theta(a_{i+1}; s_{i+1}) \\
&\cdot p(s_T, r_{T-1}; s_{T-1}, a_{T-1})\pi^\theta(\mathcal{A}_{S_T}; s_T)\cdot p(\mathcal{S}\times\{r_T\}; s_T, \mathcal{A}_{s_T})\nabla_\theta(\log(\pi^\theta(a_{t'}; s_{t'}))r_{t^*} \\
&= \sum_{a_0}\sum_{s_1}\sum_{r_1}\cdots\sum_{s_{T-1}}\sum_{a_{T-1}}\sum_{r_{T-1}} \delta_s(s_0)\pi^\theta(a_0; s_0)\prod_{t=0}^{T-2} p(s_{i+1}, r_i; s_i, a_i)\pi^\theta(a_{i+1}; s_{i+1}) \\
&\cdot \nabla_\theta(\log(\pi^\theta(a_{t'}; s_{t'}))r_{t^*} \\
&\cdots \\
&= \sum_{a_0}\sum_{s_1}\sum_{r_1}\cdots\sum_{s_{t'-1}}\sum_{a_{t'-1}}\sum_{r_{t'-1}} \delta_s(s_0)\pi^\theta(a_0; s_0)\prod_{t=0}^{t'-2} p(s_{i+1}, r_i; s_i, a_i)\pi^\theta(a_{i+1}; s_{i+1}) \\
&\cdot \underbrace{\sum_{a_{t'}}\sum_{s_{t'}}\sum_{r_{t'}} p(s_{t'}, r_{t'}; s_{t'-1}, a_{t'-1})}_{=1} \underbrace{\pi^\theta(a_{t'}; s_{t'})\nabla_\theta(\log(\pi^\theta(a_{t'}; s_{t'}))}_{=\nabla_\theta\pi^\theta(a_{t'}; s_{t'})} r_{t^*} \\
&= \sum_{a_0}\sum_{s_1}\sum_{r_1}\cdots\sum_{s_{t'-1}}\sum_{a_{t'-1}}\sum_{r_{t'-1}} \delta_s(s_0)\pi^\theta(a_0; s_0)\prod_{t=0}^{t'-2} p(s_{i+1}, r_i; s_i, a_i)\pi^\theta(a_{i+1}; s_{i+1}) \\
&\cdot \underbrace{\nabla_\theta \sum_{a_{t'}} \pi^\theta(a_{t'}; s_{t'})}_{=\nabla_\theta 1 = 0} \\
&= 0
\end{aligned}$$

Thus the assertion follows. □

This has a lower variance as the one with R_0^T, because a lot of terms that are in expectation zero are gone. Finally, we can also show that we can replace the reward to go by the Q-function to go, i.e.

$$Q_t^{\pi^\theta}(s,a) := \mathbb{E}^{\pi^\theta}[\sum_{i=t}^{T} R_i \mid S_t = s, A_t = a] = \mathbb{E}^{\pi^\theta}[R_t^T \mid S_t = s, A_t = a] \overset{Markov}{=} \mathbb{E}_{s'}^{\pi^\theta}[R_t^T \mid S_t = s, A_t = a]$$

Theorem 1.5

Assume that (S, A, R) is a T-step MDP with finite state action spaces and consider a stationary differentiable parameterized family of policies $\{\pi^\theta \mid \theta \in \Theta\}$. Then it holds that the gradient of the value function exists wrt. θ and is given by

$$\nabla_\theta J_s(\theta) = \mathbb{E}_s^{\pi^\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t))Q_t^{\pi^\theta}(S_t, A_t)].$$

Proof. We have that

$$\begin{aligned}
\nabla_\theta J_{s'}(\theta) &= \mathbb{E}_{s'}^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t))R_t^T] \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \mathbb{E}_{s'}^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t))R_t^T \mid S_t = s, A_t = a]\mathbb{P}_{s'}^{\pi^\theta}(S_t = s, A_t = a) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \mathbb{E}_{s'}^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(a; s))R_t^T \mid S_t = s, A_t = a]\mathbb{P}_{s'}^{\pi^\theta}(S_t = s, A_t = a) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(a; s))\mathbb{E}_{s'}^{\pi\theta}[R_t^T \mid S_t = s, A_t = a]\mathbb{P}_{s'}^{\pi^\theta}(S_t = s, A_t = a) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(a; s))Q_t^{\pi^\theta}(s, a)\mathbb{P}_{s'}^{\pi^\theta}(S_t = s, A_t = a) \\
&= \mathbb{E}_s^{\pi\theta}[\sum_{t=0}^{T-1} \nabla_\theta(\log(\pi^\theta(A_t; S_t))Q_t^{\pi^\theta}(S_t, A_t)].
\end{aligned}$$

□

This has the lowest variance of all.
Note that here we parametrize not only the policy, but also $Q_t^{\pi^\theta}$. One can interpret Policy gradient as a continuous version of Policy iteration. We first evaluate the policy, by calculating $Q_t^{\pi^\theta}$ and then the gradient step is the improvement. In Policy Iteration we only considered deterministic stationary policies and here we only consider arbitrary stationary polices.
The Q-value to go is the expected reward to go, i.e. we have an inner and an outer expectation. The rewards to go depend on the outer expectation, but for the Q-Value to go, it only depends its own inner expectation. For now it seems mysterious why the representation of the Q-Value might be beneficial. This will be answered later in ???
Finally, note that R_t^T is one sample from $Q_t^{\pi^{\theta_k}}(A_t, S_t)$.

Definition 1.6

In general, methods where we try to maximize f numerically and where the derivative of f takes the form of an expectation that can be sampled are called stochastic gradient methods, where the gradient is usually of the form

$$\nabla f(\theta) = \mathbb{E}[g(X, \theta)].$$

If unbiased samples of the gradient $\tilde{\nabla} f$ are available, then the scheme

$$\theta_{n+1} \leftarrow \theta_n + \alpha \tilde{\nabla} f(\theta_n)$$

is called stochastic gradient algorithm.

If multiple samples are used for one estimate of the gradient, we call this batch stochastic gradient ascent algorithm. In Policy gradient methods the situation is more deleicate, because

$$\nabla f(\theta) = \mathbb{E}^{\theta}[g(X, \theta)].$$

Thus we get the following two stochastic gradient algorithms:

$$\theta_{n+1} = \theta_n + \alpha \mathbb{E}_s^{\pi_\theta}[\sum_{t=0}^{T-1} \nabla_\theta (\log(\pi^\theta(A_t; S_t)) Q_t^{\pi^\theta}(S_t, A_t)]$$

and

$$\theta_{n+1} = \theta_n + \alpha \mathbb{E}_s^{\pi_\theta}[\sum_{t=0}^{T-1} \nabla_\theta (\log(\pi^\theta(A_t; S_t)) \sum_{i=t}^{T-1} R_i)].$$

Again, the first method is for now unfeasable. The second one could be obtained by taking Monte-Carlo Samples. An estimator for the gradient is the given by

$$\tilde{\nabla} J_\mu(\theta) = \frac{1}{K} \sum_{i=1}^{K} \left(\sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(a_t^i; s_t^i)) \sum_{t'=t}^{T-1} r_{t'}^i \right),$$

where we sample the trajectories $(s_0^i, a_0^i, r_0^i, ..., s_T^i, a_T^i, r_T^i)$ K times according to the current policy π^{θ_n} and the initial distribution μ.
In Mathematics this does not sound bad, to sample a trajectory. This is because, if we play a policy K times then this could be harmful, to the agent or its environment, if the underlying policy is bad. Thus we want to take as few samples as possible and play bad policies not for long!

Algorithm 18: REINFORCE: (Batch-)Stochastic policy–gradient algorithm

Data : Initial parameter θ_0, batch size $K \geq 1$, initial state distribution μ
Result: Approximate optimal policy $\pi^{\theta_L} \approx \pi^{\theta^*}$
$l \leftarrow 1$;
while *not converged* **do**
  for $i = 1, \ldots, K$ **do**
    Sample initial state $s_0^i \sim \mu$;
    Sample trajectory $\left(s_0^i, a_0^i, s_1^i, r_1^i, \ldots, a_{T-1}^i, s_T^i, r_T^i\right)$ using $\pi^{\theta_{l-1}}$;
  Choose step size α;
  $\tilde{\nabla} J(\theta_{l-1}) = \frac{1}{K} \sum_{i=1}^{K} \Big[\sum_{t=0}^{T-1} \nabla_\theta \log \pi^{\theta_{l-1}}\big(a_t^i \mid s_t^i\big) \sum_{t'=t}^{T-1} r_{t'+1}^i \Big]$;
  $\theta_l \leftarrow \theta_{l-1} - \alpha \tilde{\nabla} J(\theta_{l-1})$;
  $l \leftarrow l + 1$;
return π^{θ_l}

Note that we need to sample from π^{θ_k} because the expectation depends on this policy, i.e. $\mathbb{E}^{\pi^\theta}$. Later in importance sampling methods we will use a different policy π', but in order to use a different policy we need the expectation also to be depended on that policy, i.e. $\mathbb{E}^{\pi'}$.

Why is this Reinforcement Learning?

Get rollouts by playing the current policy π^{θ_k}. This gives rewards. Then update according to these rewards the θ_k in a gradient ascent scheme. As long as the gradient of the value function is sufficiently approximated, we get an improvement. Here lies the dilemma: We want to do as many gradient steps as computationally possible, but also need to create as many rollouts as possible in order to sufficiently approximate the gradient. If we approximate the gradient with a lot of samples, this might be so costly that we cannot do as many gradient steps! Thus the number of Rollouts for the approximation of the gradient should be chosen according to the noisyness of the environment/Value function (Not done here).

Note that the standard REINFORCE algorithm as above performance really bad!

1.2 Infinite-Time MDPs with discounted rewards

For infinite time MDPs thast are discounted the stationarity is justified. But this infinite sum is going to make everything a lot more difficult! Now the value function is

$$J_s(\theta) := \mathbb{E}_s^{\pi^\theta}[\sum_{t=0}^{\infty} \gamma^t R_t] = \sum_{a \in \mathcal{A}_s} \pi^\theta(a; s) Q^{\pi^\theta}(a, s).$$

Our first result is the following

Theorem 1.7

For a value function J_s that is differentiable in every state $s \in \mathcal{S}$ we have

$$\nabla J_s(\theta) = \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} \rho_s^\pi(s') \nabla \pi^\theta(a; s') Q^{\pi^\theta}(s', a),$$

where $\rho_\mu^\pi(s) := \sum_{t=0}^{\infty} \gamma^t \mathbb{P}_\mu^\pi(S_t = s) = \mathbb{E}_\mu^\pi[\sum_{t=0}^{\infty} \gamma^t \mathbf{1}_{S_t = s}]$.

Proof. The idea of the proof is: Chain Rule and Dynamic programming Bellman equation. Remeber that in dynamic programming we had that the value of the policy equals: play one step + $\gamma \cdot$ restart the problem.
We first define

$$p(s \to s'; n) := \mathbb{P}_s^\pi(S_t = s')$$

which denotes the probability of transitioning from s to s' in n steps under the policy π. We want to show via induction that for all $n \in \mathbb{N}$

$$\begin{aligned} \nabla J_s(\theta) = &\sum_{t=0}^{n} \sum_{s' \in \mathcal{S}} \gamma^t p(s \to s'; t) \sum_{a \in \mathcal{A}_{s'}} \nabla(\pi^\theta(a; s')) Q^{\pi^\theta}(s', a) \\ &+ \sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \nabla J_{s'}(\theta). \end{aligned}$$

IA:

$$
\begin{aligned}
\nabla J_s(\theta) &= \nabla \sum_{a\in\mathcal{A}} \pi^\theta(a;s) Q^{\pi^\theta}(s,a) \\
&= \sum_{a\in\mathcal{A}} \Big(\nabla\pi^\theta(a;s) Q^{\pi^\theta}(s,a) + \pi^\theta(a;s)\nabla Q^{\pi^\theta}(s,a) \Big) \\
&= \sum_{a\in\mathcal{A}} \left(\nabla\pi^\theta(a;s) Q^{\pi^\theta}(s,a) + \pi^\theta(a;s)\nabla \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}_{s'}} p(s';s,a)\pi^\theta(a';s') Q^{\pi^\theta}(s',a') \right) \right) \\
&= \sum_{a\in\mathcal{A}} \left(\nabla\pi^\theta(a;s) Q^{\pi^\theta}(s,a) + \pi^\theta(a;s)\nabla\gamma \sum_{s'\in\mathcal{S}} p(s';s,a) V^{\pi^\theta}(s') \right) \\
&= \sum_{a\in\mathcal{A}} \left(\nabla\pi^\theta(a;s) Q^{\pi^\theta}(s,a) + \pi^\theta(a;s)\gamma \sum_{s'\in\mathcal{S}} p(s';s,a)\nabla \left(\sum_{a'\in\mathcal{A}_{s'}} \pi^\theta(a',s') \left(r(s',a') + \gamma \sum_{s''\in\mathcal{S}} p(s'';s',a') V^{\pi^\theta}(s'') \right) \right) \right)
\end{aligned}
$$

We can rewrite this one term inside the above as

$$
\begin{aligned}
&\nabla \left(\sum_{a'\in\mathcal{A}_{s'}} \pi^\theta(a',s') \left(r(s',a') + \gamma \sum_{s''\in\mathcal{S}} p(s'';s',a') V^{\pi^\theta}(s'') \right) \right) \\
&\overset{prodRule}{=} \sum_{a'\in\mathcal{A}_{s'}} \nabla\pi^\theta(a',s') \left(r(s',a') + \gamma \sum_{s''\in\mathcal{S}} \underbrace{p(s'';s',a') V^{\pi^\theta}(s'')}_{= p(s'';s',a') \sum_{a''\in\mathcal{A}_{s''}} \pi^\theta(a'';s'') Q^{\pi^\theta}(s'',a'')} \right) \\
&+ \sum_{a'\in\mathcal{A}_{s'}} \pi^\theta(a',s') \nabla \left(r(s',a') + \gamma \sum_{s''\in\mathcal{S}} p(s'';s',a') V^{\pi^\theta}(s'') \right) \\
&= \sum_{a'\in\mathcal{A}_{s'}} \left(\nabla\pi^\theta(a',s') Q^{\pi^\theta}(s',a') + \pi^\theta(a',s')\gamma \sum_{s''\in\mathcal{S}} p(s'';s',a') \nabla V^{\pi^\theta}(s'') \right)
\end{aligned}
$$

Plugging this in into the main calulation yields

$$\sum_{a\in\mathcal{A}}\left(\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\nabla\left(\sum_{a'\in\mathcal{A}_{s'}}\pi^\theta(a',s')\left(r(s',a')+\gamma\sum_{s''\in\mathcal{S}}p(s'';s',a')V^{\pi^\theta}(s'')\right)\right)\right)$$

$$=\sum_{a\in\mathcal{A}}\left(\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\left(\sum_{a'\in\mathcal{A}_{s'}}\left(\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\pi^\theta(a',s')\gamma\sum_{s''\in\mathcal{S}}p(s'';s',a')\nabla V^{\pi}\right.\right.\right.$$

$$=\sum_{a\in\mathcal{A}}\left(\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\sum_{a'\in\mathcal{A}_{s'}}\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\sum_{a'\in\mathcal{A}_{s'}}\pi^\theta(a',\right.$$

$$=\sum_{a\in\mathcal{A}}\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\sum_{a\in\mathcal{A}}\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\sum_{a'\in\mathcal{A}_{s'}}\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\sum_{s''\in\mathcal{S}}\gamma^2\sum_{a\in\mathcal{A}}\pi^\theta(a;s)\sum_{s'\in\mathcal{S}}p(s';s,a)$$

$$=\sum_{a\in\mathcal{A}}\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\sum_{a\in\mathcal{A}}\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\sum_{a'\in\mathcal{A}_{s'}}\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\sum_{s''\in\mathcal{S}}\gamma^2\sum_{a\in\mathcal{A}}\sum_{s'\in\mathcal{S}}\sum_{a'\in\mathcal{A}_{s'}}\underbrace{\pi^\theta(a;s)p(s';}_{\mathbb{P}(A_0=}$$

$$=\sum_{a\in\mathcal{A}}\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\sum_{a\in\mathcal{A}}\pi^\theta(a;s)\gamma\sum_{s'\in\mathcal{S}}p(s';s,a)\sum_{a'\in\mathcal{A}_{s'}}\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\sum_{s''\in\mathcal{S}}\gamma^2p(s\to s'',2)\nabla V^{\pi^\theta}(s'')$$

$$=\sum_{a\in\mathcal{A}}\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\sum_{s'\in\mathcal{S}}\gamma\underbrace{\sum_{a\in\mathcal{A}}\pi^\theta(a;s)p(s';s,a)}_{=p(s\to s',1)}\sum_{a'\in\mathcal{A}_{s'}}\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\sum_{s''\in\mathcal{S}}\gamma^2p(s\to s'',2)\nabla V^{\pi^\theta}(s'')$$

$$=\sum_{a\in\mathcal{A}}\nabla\pi^\theta(a;s)Q^{\pi^\theta}(s,a)+\sum_{s'\in\mathcal{S}}\gamma p(s\to s',1)\sum_{a'\in\mathcal{A}_{s'}}\nabla\pi^\theta(a',s')Q^{\pi^\theta}(s',a')+\sum_{s''\in\mathcal{S}}\gamma^2p(s\to s'',2)\nabla V^{\pi^\theta}(s'')$$

$$=\left(\underbrace{1}_{=\sum_{s'\in\mathcal{S}}\gamma^0\mathbb{P}(S_0=s,S_0=s')}+\sum_{s'\in\mathcal{S}}\gamma p(s\to s',1)\right)\sum_{a\in\mathcal{A}_{s'}}\nabla\pi^\theta(a;s')Q^{\pi^\theta}(s',a)+\sum_{s''\in\mathcal{S}}\gamma^2p(s\to s'',2)\nabla V^{\pi^\theta}(s'')$$

$$=\sum_{t=0}^{1}\sum_{s'\in\mathcal{S}}\gamma^tp(s\to s',t)\sum_{a\in\mathcal{A}_{s'}}\nabla\pi^\theta(a;s')Q^{\pi^\theta}(s',a)+\sum_{s''\in\mathcal{S}}\gamma^2p(s\to s'',2)\nabla V^{\pi^\theta}(s'')$$

IS: Assume that the statement holds for every $n \in \mathbb{N}$

$$\begin{aligned}
\nabla J_s(\theta) &\overset{IV}{=} \sum_{t=0}^{n} \sum_{s' \in \mathcal{S}} \gamma^t p(s \to s'; t) \sum_{a \in \mathcal{A}_{s'}} \nabla(\pi^\theta(a; s')) Q^{\pi^\theta} \\
&+ \sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \nabla J_{s'}(\theta) \\
&= \sum_{t=0}^{n} \sum_{s' \in \mathcal{S}} \gamma^t p(s \to s'; t) \sum_{a \in \mathcal{A}_{s'}} \nabla(\pi^\theta(a; s')) Q^{\pi^\theta} \\
&+ \sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \\
&\cdot \nabla \left(\sum_{a' \in \mathcal{A}_{s'}} \pi^\theta(a'; s') r(s', a') + \sum_{a' \in \mathcal{A}_{s'}} \gamma \sum_{s'' \in \mathcal{S}} p(s''; a', s') J_{s''}(\theta) \right) \\
&\overset{above}{=} \sum_{t=0}^{n} \sum_{s' \in \mathcal{S}} \gamma^t p(s \to s'; t) \sum_{a \in \mathcal{A}_{s'}} \nabla(\pi^\theta(a; s')) Q^{\pi^\theta} \\
&+ \sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \\
&\cdot \sum_{a' \in \mathcal{A}_{s'}} \left(\nabla \pi^\theta(a', s') Q^{\pi^\theta}(s', a') + \pi^\theta(a', s') \gamma \sum_{s'' \in \mathcal{S}} p(s''; s', a') \nabla V^{\pi^\theta}(s'') \right)
\end{aligned}$$

Now looking only at the second summand yields

$$\begin{aligned}
&\sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \\
&\cdot \sum_{a' \in \mathcal{A}_{s'}} \left(\nabla \pi^\theta(a', s') Q^{\pi^\theta}(s', a') + \pi^\theta(a', s') \gamma \sum_{s'' \in \mathcal{S}} p(s''; s', a') \nabla V^{\pi^\theta}(s'') \right) \\
&= \sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \sum_{a' \in \mathcal{A}_{s'}} \nabla \pi^\theta(a', s') Q^{\pi^\theta}(s', a') \\
&+ \sum_{s' \in \mathcal{S}} \underbrace{\gamma^{n+1} p(s \to s'; n+1) \sum_{a' \in \mathcal{A}_{s'}} \pi^\theta(a', s') \gamma \sum_{s'' \in \mathcal{S}} p(s''; s', a') \nabla V^{\pi^\theta}(s'')}_{\substack{= \gamma^{n+2} \sum_{a' \in \mathcal{A}_{s'}} \sum_{s'' \in \mathcal{S}} p(s \to s'; n+1) \pi^\theta(a', s') p(s''; s', a') \nabla V^{\pi^\theta}(s'') \\ = \gamma^{n+2} \sum_{s'' \in \mathcal{S}} p(s \to s''; n+2) \nabla V^{\pi^\theta}(s'')}}
\end{aligned}$$

and thus the assertion follows. Now note that the second summand of the formula we showed via induction goes to zero as we let $n \to \infty$, i.e.

$$\begin{aligned}
\sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \nabla J_{s'}(\theta) &\leq \sum_{s' \in \mathcal{S}} \gamma^{n+1} p(s \to s'; n+1) \max_{s'} \nabla J_{s'}(\theta) \\
&= \gamma^{n+1} \max_{s'} \nabla J_{s'}(\theta) \underbrace{\sum_{s' \in \mathcal{S}} p(s \to s'; n+1)}_{=1} \\
&\to 0, \quad n \to \infty
\end{aligned}$$

Henceforth we have shown

$$\nabla J_s(\theta) = \sum_{t=0}^{n} \sum_{s' \in \mathcal{S}} \gamma^t p(s \to s'; t) \sum_{a \in \mathcal{A}_{s'}} \nabla(\pi^\theta(a; s')) Q^{\pi^\theta}$$
$$\overset{DCT}{=} \sum_{s' \in \mathcal{S}} \underbrace{\sum_{t=0}^{n} \gamma^t p(s \to s'; t)}_{=\rho_s^{\pi^\theta}(s')} \sum_{a \in \mathcal{A}_{s'}} \nabla(\pi^\theta(a; s')) Q^{\pi^\theta}$$

□

In order to make the gradient an expectation, we will define the following

Definition 1.8

The discounted visitation measure under the policy π when starting with measure μ is defined for all s as

$$d_\mu^\pi(s) := \frac{\rho_\mu^\pi(s)}{\sum_{s' \in \mathcal{S}} \rho_\mu^\pi(s')}$$

which denotes the relative visitation of state s compared to all other states.

This large sum in the fraction can be rewritten as follows

$$\sum_{s \in \mathcal{S}} \rho_\mu^\pi(s) = \sum_{s \in \mathcal{S}} \mathbb{E}_\mu[\sum_{t=0}^{\infty} \gamma^t \mathbf{1}_{S_t = s}] = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\mu[\sum_{s \in \mathcal{S}} \mathbf{1}_{S_t = s}] = \frac{1}{1-\gamma}.$$

Thus we can write it also as

$$d_\mu^\pi(s) = (1-\gamma)\rho_\mu^\pi(s).$$

Using this we can now write the gradient as an expectation

Theorem 1.9

Under the assumption that $J_s(\theta)$ is differentiable we can write the gradient for every starting state $s \in \mathcal{S}$ as

$$\nabla J_s(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{S \sim d_s^\pi, A \sim \pi^\theta(\cdot; S)}[\nabla \log(\pi^\theta(A; S)) Q^{\pi^\theta}(S, A)].$$

Proof. This follows immediately with the lemma above

$$\begin{aligned}
\nabla J_s(\theta) &= \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} \rho_s^\pi(s') \nabla \pi^\theta(a; s') Q^{\pi^\theta}(s', a) \\
&= \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} d_s^\pi(s') \frac{1}{1-\gamma} \nabla \pi^\theta(a; s') Q^{\pi^\theta}(s', a) \\
&= \frac{1}{1-\gamma} \sum_{s' \in \mathcal{S}} \sum_{a \in \mathcal{A}_{s'}} \underbrace{d_s^\pi(s') \pi^\theta(a; s')}_{\text{probability weights}} \nabla(\log(\pi^\theta(a; s'))) Q^{\pi^\theta}(s', a) \\
&= \frac{1}{1-\gamma} \mathbb{E}_{S \sim d_s^\pi, A \sim \pi^\theta(\cdot; S)}[\nabla \log(\pi^\theta(A; S)) Q^{\pi^\theta}(S, A)].
\end{aligned}$$

□

The problem now is that we do not know d_s^π and we do not know Q^{π^θ}.
How to estimate the Q-value has already been talked about. How to estimate $d_\mu^{\pi^\theta}$ could be done in the following way. Sample one "infinite" rollout and then count the empirical visitation of every state $s \in \mathcal{S}$. Then using this construct the probability weights of this empirical measure and then take a sample from it, to use it as one sample for the expectation, i.e. $S \sim \hat{d_\mu^{\pi^\theta}}$. So we build with one rollout our empirical distribution and then sample from it.
Problem: Infinite rollout is not realistic!
Solution: Write the infinite sum as a sum up to a geometrically $Geo(1-\gamma)$ distributed terminating time:

$$\begin{aligned}
\rho_\mu^\pi(s) &:= \mathbb{E}_\mu^\pi[\sum_{t=0}^{\infty} \gamma^t \mathbf{1}_{S_t=s}] \\
&= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\mu^\pi[\mathbf{1}_{S_t=s}] \\
&= \sum_{t=0}^{\infty} (1-(1-\gamma)^t \mathbb{E}_\mu^\pi[\mathbf{1}_{S_t=s}] \\
&\overset{DefGeo}{=} \sum_{t=0}^{\infty} \mathbb{P}(t \leq T) \mathbb{E}_\mu^\pi[\mathbf{1}_{S_t=s}] \\
&\overset{independent}{=} \sum_{t=0}^{\infty} \mathbb{E}_\mu^\pi[\mathbf{1}_{t\leq T} \mathbf{1}_{S_t=s}] \\
&= \mathbb{E}_\mu^\pi[\sum_{t=0}^{\infty} \mathbf{1}_{t\leq T} \mathbf{1}_{S_t=s}] \\
&= \mathbb{E}_\mu^\pi[\sum_{t=0}^{T} \mathbf{1}_{S_t=s}]
\end{aligned}$$

where we used that for a geometric rv Y we have that

$$\begin{aligned}
\mathbb{P}(Y \geq t) &= \sum_{i=t}^{\infty} \gamma(1-\gamma)^{i-1} \\
&= \sum_{i=t}^{\infty} \gamma(1-\gamma)^{i-1+t-t} \\
&= \sum_{k=0}^{\infty} \gamma(1-\gamma)^{t-1}(1-\gamma)^k, \quad k := i-t \\
&= \gamma(1-\gamma)^{t-1} \frac{1}{\gamma} = (1-\gamma)^{t-1}
\end{aligned}$$

although in the sum above $t = 0, ..., T$ but Geometric random variables are only defined on $t \geq 1$, thus doing an index shift yields γ^t.

Remark 1.1

In order to avoid the state visitation measure we can also write the gradient in the following way.

Theorem 1.10

Suppose that $(s,a) \mapsto \nabla \log(\pi^\theta(a;s))Q^{\pi^\theta}(s,a)$ is bounded, then

$$\nabla J_s(\theta) = \mathbb{E}_s^{\pi^\theta}[\sum_{t=0}^{\infty} \gamma^t \nabla_\theta \log(\pi^\theta(A_t;S_t))Q^{\pi^\theta}(S_t,A_t)].$$

Proof. Again using the first Theorem of this section we have that

$$\begin{aligned}
\nabla J_s(\theta) &= \sum_{s'\in\mathcal{S}} \sum_{a\in\mathcal{A}_{s'}} \rho_s^\pi(s') \nabla \pi^\theta(a;s') Q^{\pi^\theta}(s',a) \\
&= \sum_{s'\in\mathcal{S}} \sum_{a\in\mathcal{A}_{s'}} \sum_{t=0}^{\infty} \gamma^t p(s\to s',t) \nabla \pi^\theta(a;s') Q^{\pi^\theta}(s',a) \\
&= \sum_{t=0}^{\infty} \sum_{s'\in\mathcal{S}} \gamma^t p(s\to s',t) \sum_{a\in\mathcal{A}_{s'}} \nabla \pi^\theta(a;s') Q^{\pi^\theta}(s',a) \\
&= \sum_{t=0}^{\infty} \mathbb{E}_s^{\pi^\theta}[\sum_{a\in\mathcal{A}_{s'}} \gamma^t \nabla \pi^\theta(a;S_t) Q^{\pi^\theta}(S_t,a)] \\
&= \sum_{t=0}^{\infty} \mathbb{E}_s^{\pi^\theta}[\sum_{a\in\mathcal{A}_{s'}} \gamma^t \nabla \log(\pi^\theta(a;S_t)) \pi^\theta(a;S_t) Q^{\pi^\theta}(S_t,a)] \\
&= \sum_{t=0}^{\infty} \mathbb{E}_s^{\pi^\theta}[\gamma^t \nabla(\log(\pi^\theta(A_t;S_t)))Q^{\pi^\theta}(S_t,A_t)].
\end{aligned}$$

□

Claim 1.11

We assume that the policy π^θ is $L-$smooth wrt. to θ and that the score function is bounded by above, i.e.

$$\forall\theta\in\Theta \exists B_\theta : \quad \|\nabla \log(\pi^\theta(a;s))\| \le B_\theta$$

A dirty trick computer scientist use is to cut of the infinite sum. This is wrong. Here again we can write the infinite sum as a geometric sum, but this time with two geometric random variables.

Theorem 1.12

Suppose that the policy satisfies the assumptions 1.2 and $T \sim Geo(1-\gamma)$ and $T' \sim Geo(1-\gamma^{1/2})$ two independent random variables. Then

$$\nabla J_s(\theta) = \frac{1}{1-\gamma} \mathbb{E}_s^{\pi^\theta}[\nabla \log(\pi^\theta(A_T;S_T)) \sum_{t=T}^{T+T'} \gamma^{(t-T)/2} R(S_t,A_t)].$$

Here the Q-function is replaced by one sample and the T' comes from the infinite sum of the Q-function. This is also why there is the term $1/2$ in the exponent, because $T' \sim (1-\gamma^{1/2})$ and we can write value function as

$$V^\pi(s) := \mathbb{E}_s^\pi[\sum_{t=0}^{\infty} R_t] = \mathbb{E}_s^\pi[\sum_{t=0}^{T'} (\gamma^{1/2})^t R_t], \quad T' \sim Geo(1-\gamma^{1/2}).$$

1.3 Convergence of REINFORCE

Lemma 1.13

Under the assumptions 1.2 it follows that $J_{s_0}(\theta)$ is L-smooth wrt. θ.

Lemma 1.14

The estimator $\nabla \hat{J_s}(\theta)$ from the mini batch REINFORCE algorithm is bounded, i.e.

$$\|\nabla \hat{J_s}(\theta)\| \leq C$$

and has bounded variance, i.e.

$$\mathbb{E}[\|\nabla \hat{J_s}(\theta) - \nabla J_s(\theta)\|^2] \leq B.$$

Thus under specific assumptions the REINFORCE algorithm satisfies the requirements of the SGD theorem. Note that it only converges to a θ^* with $\|\nabla J_s(\theta^*)\|^2 = 0$ which is not necessarily a maximum.
Under strong assumptions on the Policy then $J_s(\theta)$ can satisfy the weak PL-inequality and then even converge to the maximum. This is then a SGD version where we have instead of L-smoothnes the weak PL-inequality (not gone into further detail).

1.4 Variance Reduction methods

In bandits we used a baseline $(X_A - b)$ where we can control with b how far we go in the direction of the gradient. In the finite time setting we can then show that

Theorem 1.15

With a finite time (S, A, R) T-step MDP with finite state and action spaces and differentiable parameterized family of policies it holds that for all $b \in \mathbb{R}$

$$\nabla_\theta J_s(\theta) = \mathbb{E}_s^{\pi^\theta}[\sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(A_t; S_t)) \left(\underbrace{Q_t^{\pi^\theta}(S_t, A_t)}_{\text{or } R_0^T \text{ or } R_t^T} - b \right)]$$

Proof. We only need to show that for every $t = 0, ..., T-1$ it holds that

$$\begin{aligned}
\mathbb{E}_s^{\pi^\theta}[\nabla_\theta \log(\pi^\theta(A_t, S_t))b] &= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \nabla_\theta \log(\pi^\theta(a;s))b\mathbb{P}_s^{\pi^\theta}(A_t = a, S_t = s) \\
&= \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}_s} \nabla_\theta \log(\pi^\theta(a;s))b\mathbb{P}_s^{\pi^\theta}(S_t = s)\pi^\theta(a;s) \\
&= \sum_{s\in\mathcal{S}} b\mathbb{P}_s^{\pi^\theta}(S_t = s) \sum_{a\in\mathcal{A}_s} \nabla_\theta \pi^\theta(a;s) \frac{1}{\pi^\theta(a;s)} \pi^\theta(a;s) \\
&= \sum_{s\in\mathcal{S}} b\mathbb{P}_s^{\pi^\theta}(S_t = s) \underbrace{\nabla_\theta \sum_{a\in\mathcal{A}_s} \pi^\theta(a;s)}_{=\nabla_\theta 1 = 0} = 0.
\end{aligned}$$

□

One can show that the bias b can depend on everything except for a_t. Reasonable is of course that it only depends on the past.
In the bandit chapter we had a choice for b that reduces the variance. A similiar thing could be derived here. Note that this optimal bias was an expectation and thus not obtainable without errors.
Another approach was to choose the bias as the current estimated reward, i.e.

$$b_t := \hat{V}_t^\pi(s).$$

One then calls $\hat{Q}_t^{\pi_{\theta_k}}(s,a) - \hat{V}_t^{\pi_{\theta_k}}(s)$ advantage function, i.e. the rewards are only positive if we get better rewards than the current policy if we choose action a_t.???

Importance Sampling:
The idea of importance sampling is that assume we want to calculate $\mathbb{E}[g(X)]$ but $g(X)$ has a high variance. Then sample from from $\tilde{g}(Y)$ with lower variance, i.e.

$$\mathbb{E}[Y] = \int g(x) f_X(x) dx = \int \underbrace{g(x)\frac{f_X(x)}{f_Y(x)}}_{=\tilde{g}(x)} f_Y(x) dx = \mathbb{E}[\tilde{g}(Y)].$$

So we need to find Y in order to reduce the variance. In RL it is completely different. Here we cannot sample from X and thus have to sample from Y. Of course we also want to find a Y that reduces the variance. In RL we cannot sample from π^θ and thus have to sample from π^b.

Sparse Rewards:
In an untrained model with an environment that has sparse rewards, then in the beginning the model learns nothing, because random action rarely lead to any reward. Thus it is of great importance to use pretrained models if possible.

Lemma 1.16: Likelihood-Ration Trick

Suppose π^b is a policy with strict positive weights (we call it behavioral policy) then

$$\nabla_\theta J_\mu(\theta) := \mathbb{E}_\mu^{\pi^b}[\prod_{i=0}^{T-1} \frac{\pi^\theta(A_i;S_i)}{\pi^b(A_i;S_i)} \sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(A_t;S_i))Q_t^{\pi^\theta}(S_t, A_t)].$$

Proof. With a Lemma from before we have that

$$\begin{aligned}
\nabla_\theta J_\mu(\theta) &= \mathbb{E}_\mu^{\pi^\theta}[\sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(A_t;S_t)) Q_t^{\pi^\theta}(S_t,A_t)] \\
&= \sum_{\tau\in\mathcal{T}} \sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(a_t;s_t)) Q_t^{\pi^\theta}(s_t,a_t) \mathbb{P}_\mu^{\pi^\theta}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_{T-1},r_{T-1})) \\
&= \sum_{\tau\in\mathcal{T}} \sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(a_t;s_t)) Q_t^{\pi^\theta}(s_t,a_t) \frac{\mathbb{P}_\mu^{\pi^\theta}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_{T-1},r_{T-1}))}{\mathbb{P}_\mu^{\pi^b}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_{T-1},r_{T-1}))} \mathbb{P}_\mu^{\pi^b}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_T \\
&\overset{(*)}{=} \sum_{\tau\in\mathcal{T}} \sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(a_t;s_t)) Q_t^{\pi^\theta}(s_t,a_t) \prod_{i=0}^{T-1} \frac{\pi^\theta(a_i;s_i}{\pi^b(a_i;s_i)} \mathbb{P}_\mu^{\pi^b}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_{T-1},r_{T-1})) \\
&= \mathbb{E}_\mu^{\pi^b}[\sum_{t=0}^{T-1} \nabla_\theta \log(\pi^\theta(A_t;S_t)) Q_t^{\pi^\theta}(S_t,A_t) \prod_{i=0}^{T-1} \frac{\pi^\theta(A_i;S_i)}{\pi^b(A_i;S_i)}]
\end{aligned}$$

altough we used in $(*)$ that

$$\begin{aligned}
&\frac{\mathbb{P}_\mu^{\pi^\theta}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_{T-1},r_{T-1}))}{\mathbb{P}_\mu^{\pi^b}(\tau = (s_0,a_0,r_0,...,s_{T-1},a_{T-1},r_{T-1}))} \\
&= \frac{\mu(s_0)\pi^\theta(a_0;s_0)\prod_{i=1}^{T-1} p(s_i,r_{i-1};s_{i-1},a_{i-1})\pi^\theta(a_i;s_i)}{\mu(s_0)\pi^b(a_0;s_0)\prod_{i=1}^{T-1} p(s_i,r_{i-1};s_{i-1},a_{i-1})\pi^b(a_i;s_i)} \\
&= \frac{\prod_{i=0}^{T-1}\pi^\theta(a_i;s_i)}{\prod_{i=0}^{T-1}\pi^b(a_i;s_i)}.
\end{aligned}$$

□

Note that numerically the term

$$\frac{\prod_{i=0}^{T-1}\pi^\theta(a_i;s_i)}{\prod_{i=0}^{T-1}\pi^b(a_i;s_i)}$$

can explode when π^b is small and differs a lot from π^θ. Thus one wants to choose a π^b that is similiar to π^θ in order to reduce varinance!
Another way is to reuse samples, i.e. sample from π^b and then use it to evaluate all following π^{θ_k}. This only works well if this one behavioral is similiar to the others. Another way is to do clustering on the space of the policies, i.e. on $\{x \in [0,1]^{\mathcal{A}} \mid \sum_{a\in\mathcal{A}} x_a = 1\}$. Here one needs a different definition a distance! Then using these clusters one can sample from the cluster centers in order to evaluate the policies from these clusters using the samples from the cluster center.

CHAPTER 6

IMPORTANT EXERCISES (29)

Exercises 1 Nr. 2
a) If the failure probability does not decay to zero, then the regret grows linearly, i.e.

$$\tau_t(\pi) := \mathbb{P}^\pi(Q_{A_t} \neq Q_*) \to c > 0, \quad t \to \infty \quad \Rightarrow \quad R_n(\pi) \in \mathcal{O}(n).$$

Proof. If the failure probability does not decay to zero (and does not alternate) then

$$\exists c > 0, T \geq 1 \forall t > T : \quad \tau_t(\pi) > c.$$

Then it holds for all $n > T$ that

$$\begin{aligned} R_n(\pi) &\geq \min_{a \neq a^*} \Delta_a \left(\sum_{t=1}^{T} \tau_t(\pi) + (n-T)c \right) \\ &\geq \min_{a \neq a^*} \Delta_a c \min\{T-1, n-T\} \end{aligned}$$

Thus $R_n(\pi)$ grows at least linear. It grows at most linear due to the fact that all learning strategies grow at most linear:

$$R_n(\pi) \leq \max_{a \in \mathcal{A}} \Delta_a \sum_{t=1}^{n} \tau_t(\pi) \leq \max_{a \in \mathcal{A}} \Delta_a \cdot n.$$

□

b) If the failure probability behaves like $\tau_t(\pi) \in \mathcal{O}(\frac{1}{n})$ then the regret is in $R_n(\pi) \in \mathcal{O}(\sum_{a \in \mathcal{A}} \log(n)) = \mathcal{O}(\log(n))$.

Proof. Note that $\{1, ..., n\} \subset [1, n]$ and that and that we can interpret $\sum_{t=2}^{n} \frac{1}{t}$ as upper bound of the integral and $\sum_{t=1}^{n-1} \frac{1}{t}$, where each decompose $[1, n]$ into n-disjoint subsets of length 1:

$$\sum_{t=2}^{n} \frac{1}{t} \leq \int_1^n \frac{1}{t} dt \leq \sum_{t=1}^{n-1} \frac{1}{t}.$$

Thus it holds

$$\sum_{t=1}^{n} \frac{1}{t} \geq \sum_{t=1}^{n-1} \geq \int_1^n \frac{1}{t} dt = \log(n)$$

and

$$\sum_{t=1}^{n} \frac{1}{t} \leq 1 + \sum_{t=2}^{n} \frac{1}{t} \leq 1 + \int_{1}^{n} \frac{1}{t} dt = 1 + \log(n).$$

Plugging all in yields the upper bound

$$R_n(\pi) \leq \max_{a \in \mathcal{A}} \Delta_a \sum_{t=1}^{n} \tau_t(\pi) \leq \max_{a \in \mathcal{A}} \Delta_a \sum_{t=1}^{n} \frac{1}{t} \leq \max_{a \in \mathcal{A}} \Delta_a (1 + \log(n)).$$

and the lower bound

$$R_n(\pi) \geq \min_{a \neq a^*} \Delta_a \sum_{t=1}^{n} \tau_t(\pi) \geq \min_{a \neq a^*} \Delta_a \sum_{t=1}^{n} \frac{1}{t} \geq \min_{a \neq a^*} \Delta_a \log(n).$$

□

Exercises 1 Nr. 3

Exercises 1 Nr. 5

Let π be an ϵ-greedy learning strategy that first plays every arm once. For a 1-Subgaussian Bandit model ν it holds that

$$\lim_{n \to \infty} \frac{R_n(\pi)}{n} = \frac{\epsilon}{K} \sum_{a \in \mathcal{A}} \Delta_a.$$

Proof. We have for all $t \geq K$ (i.e. after exploring every arm once) that

$$\mathbb{P}(A_t^\pi = a) = \frac{\epsilon}{K} + (1 - \epsilon) \mathbb{P}(\hat{Q}_a(t) \geq \max_b \hat{Q}_b(t))$$

So with probability ϵ uniform and with probability $(1 - \epsilon)$ the best arm. The lower bound follows easily by leaving a summand away:

$$\begin{aligned} \lim_{n \to \infty} \frac{R_n(\pi)}{n} &= \lim_{n \to \infty} \frac{1}{n} \sum_{a \in \mathcal{A}} \Delta_a \sum_{t=1}^{n} \mathbb{P}(A_t^\pi = a) \\ &\geq \sum_{a \in \mathcal{A}} \Delta_a \lim_{n \to \infty} \frac{1}{n} \sum_{t=1}^{n} \frac{\epsilon}{K} \\ &= \sum_{a \in \mathcal{A}} \Delta_a \frac{\epsilon}{K} \end{aligned}$$

Now one needs to show that

$$\sum_{t=1}^{\infty} \mathbb{P}(\hat{Q}_a(t) \geq \max_b \hat{Q}_b(t)) \leq C < \infty.$$

This is very long and difficult. We skip this. Then one can show that

$$\begin{aligned}\lim_{n\to\infty}\frac{R_n(\pi)}{n} &= \lim_{n\to\infty}\frac{1}{n}\sum_{a\in\mathcal{A}}\Delta_a\sum_{t=1}^{n}\mathbb{P}(A_t^\pi = a)\\ &\leq \sum_{a\in\mathcal{A}}\Delta_a\lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n}\Big(\frac{\epsilon}{K}+\mathbb{P}(\hat{Q}_a(t)\geq\max_b\hat{Q}_b(t))\Big)\\ &\leq \sum_{a\in\mathcal{A}}\Delta_a(\frac{\epsilon}{K}+\lim_{n\to\infty}\frac{C}{n})\\ &= \sum_{a\in\mathcal{A}}\Delta_a\frac{\epsilon}{K}.\end{aligned}$$

The assertion follows. □

Exercises 3 Nr. 1
Suppose a 1−Subgaussian Bandit model ν and π as the UCB learning strategy. Then it holds that the event $\hat{Q}_a(t) < Q_a(t) + \Delta$ given that $T_a(t) > \frac{2\log(1/\delta)}{\Delta_a^2}$ is given by at most $1-\delta$.

Proof. First note that that the UCB algorithm always plays every arm once in the beginning, i.e. it definitely holds that for n rounds that

$$\forall m = 1,...,n-(K-1): \quad \mathbb{P}^\pi(T_a(t)=m) > 0.$$

Further, it clearly the following equivalence holds

$$T_a(t) > \frac{2\log(1/\delta)}{\Delta_a^2} \quad\Longleftrightarrow\quad \Delta_a^2 > \frac{2\log(1/\delta)}{T_a(t)}.$$

Next we can calculate the following lower bound

$$\begin{aligned}&\mathbb{P}^\pi\left(\hat{Q}_a(t) < Q_a+\Delta_a \mid T_a(t) > \frac{2\log(1/\delta)}{T_a(t)}\right) \geq \mathbb{P}^\pi\left(\hat{Q}_a(t)-Q_a < \frac{2\log(1/\delta)}{T_a(t)} \mid T_a(t) > \frac{2\log(1/\delta)}{T_a(t)}\right)\\ &= \mathbb{P}^\pi\left(\hat{Q}_a(t)-Q_a < \frac{2\log(1/\delta)}{T_a(t)} \mid \bigcup_{m=\lceil\frac{2\log(1/\delta)}{T_a(t)}\rceil}^{n-(K-1)} T_a(t)=m\right)\\ &\overset{???}{\geq} \frac{\sum_{m=\lceil\frac{2\log(1/\delta)}{T_a(t)}\rceil}^{n-(K-1)}\mathbb{P}^\pi\left(\hat{Q}_a(t)-Q_a < \frac{2\log(1/\delta)}{T_a(t)}, T_a(t)=m\right)}{\sum_{m=\lceil\frac{2\log(1/\delta)}{T_a(t)}\rceil}^{n-(K-1)}\mathbb{P}^\pi(T_a(t)=m)}\\ &\overset{CounterProb}{=} \frac{\sum_{m=\lceil\frac{2\log(1/\delta)}{T_a(t)}\rceil}^{n-(K-1)}\left(\mathbb{P}^\pi(T_a(t)=m)-\mathbb{P}^\pi\left(\hat{Q}_a(t)-Q_a \geq \frac{2\log(1/\delta)}{T_a(t)}, T_a(t)=m\right)\right)}{\sum_{m=\lceil\frac{2\log(1/\delta)}{T_a(t)}\rceil}^{n-(K-1)}\mathbb{P}^\pi(T_a(t)=m)}.\end{aligned}$$

Now we shortly focus on of the terms above

$$\begin{aligned}
\mathbb{P}^\pi\left(\hat{Q}_a(t) - Q_a \geq \frac{2\log(1/\delta)}{T_a(t)}, T_a(t) = m\right) &= \mathbb{P}^\pi\left(\frac{1}{T_a(t)}\sum_{i=1}^{t} X_i \mathbf{1}_{A_i=a} - Q_a \geq \frac{2\log(1/\delta)}{T_a(t)}, T_a(t) = m\right) \\
&= \mathbb{P}^\pi\left(\frac{1}{m}\sum_{i=1}^{t} X_i \mathbf{1}_{A_i=a} - Q_a \geq \frac{2\log(1/\delta)}{m}, T_a(t) = m\right) \\
&= \mathbb{P}^\pi\left(\frac{1}{m}\sum_{i=1}^{t} X_i \mathbf{1}_{A_i=a} - Q_a \geq \frac{2\log(1/\delta)}{m} \mid T_a(t) = m\right)\mathbb{P}^\pi(T_a(t) = m) \\
&\leq \delta \mathbb{P}^\pi(T_a(t) = m),
\end{aligned}$$

where we used in the last step Hoeffdings Lemma, because the conditional probability is also a measure. Now plugging this into the main calculation leads to

$$\begin{aligned}
&\mathbb{P}^\pi\left(\hat{Q}_a(t) < Q_a + \Delta_a \mid T_a(t) > \frac{2\log(1/\delta)}{T_a(t)}\right) \\
&\geq \frac{\sum_{m=\lceil \frac{2\log(1/\delta)}{T_a(t)} \rceil}^{n-(K-1)} \left(\mathbb{P}^\pi(T_a(t) = m) - \mathbb{P}^\pi\left(\hat{Q}_a(t) - Q_a \geq \frac{2\log(1/\delta)}{T_a(t)}, T_a(t) = m\right)\right)}{\sum_{m=\lceil \frac{2\log(1/\delta)}{T_a(t)} \rceil}^{n-(K-1)} \mathbb{P}^\pi(T_a(t) = m)} \\
&\geq \frac{\sum_{m=\lceil \frac{2\log(1/\delta)}{T_a(t)} \rceil}^{n-(K-1)} \left(\mathbb{P}^\pi(T_a(t) = m) - \delta\mathbb{P}^\pi(T_a(t) = m)\right)}{\sum_{m=\lceil \frac{2\log(1/\delta)}{T_a(t)} \rceil}^{n-(K-1)} \mathbb{P}^\pi(T_a(t) = m)} \\
&= 1 - \delta.
\end{aligned}$$

□

Exercises 3 Nr. 2

Exercises 3 Nr. 3

For the estimator unbiased estimator

$$(X_A - b)\nabla \log(\pi^\theta(A)), \quad A \sim \pi^\theta, X_A \sim P_A$$

the choice of $b \in \mathbb{R}$ that minimizes the variance is given by

$$b^* = \frac{\mathbb{E}[X_A \|\nabla \log(\pi^\theta(A))\|_2^2]}{\mathbb{E}[\|\nabla \log(\pi^\theta(A))\|_2^2]}.$$

Proof. First note that for a random vector $X \in \mathbb{R}^d$ it holds that

$$\mathbb{V}[X] = \mathbb{E}[XX^T] - \mathbb{E}[X]\mathbb{E}[X]^T = \mathbb{E}[\|X\|_2^2] - \|\mathbb{E}[X]\|_2^2.$$

We then get

$$\begin{aligned}
\mathbb{V}[(X_A - b)\nabla \log(\pi^\theta(A))] &= \mathbb{E}[(X_A - b)^2 \|\nabla \log(\pi^\theta(A))\|_2^2] - \|\mathbb{E}[(X_A - b)\nabla \log(\pi^\theta(A))]\|_2^2 \\
&\overset{BaselineTrick}{=} \mathbb{E}[(X_A - b)^2 \|\nabla \log(\pi^\theta(A))\|_2^2] - \|\mathbb{E}[X_A \nabla \log(\pi^\theta(A))]\|_2^2 \\
&= \mathbb{E}[(X_A - b)^2 \|\nabla \log(\pi^\theta(A))\|_2^2] - \|\mathbb{E}[X_A \nabla \log(\pi^\theta(A))]\|_2^2
\end{aligned}$$

We now make this to an easy quadratic equation by defining $f(A) := \|\nabla \log(\pi^\theta(A))\|_2$ and then get that

$$\begin{aligned}\frac{\partial}{\partial b}\mathbb{V}[(X_A - b)\nabla \log(\pi^\theta(A))] &= \mathbb{E}[X_A^2 f(A)^2] - 2\mathbb{E}[X_A b f(A)^2] + b^2\mathbb{E}[f(A)^2] - \|\mathbb{E}[X_A \nabla \log(\pi^\theta(A))]\|_2^2 \\ \overset{!}{=} 0 \quad &\iff \quad 2b\mathbb{E}[f(A)^2] - 2\mathbb{E}[X_A f(A)^2] = 0\end{aligned}$$

which leads to

$$b^* = \frac{\mathbb{E}[X_A f(A)^2]}{\mathbb{E}[f(A)^2]} = \frac{\mathbb{E}[X_A \|\nabla \log(\pi^\theta(A))\|_2^2]}{\mathbb{E}[\|\nabla \log(\pi^\theta(A))\|_2^2]}.$$

This also a minimum due to the fact that

$$\frac{\partial^2}{\partial b^2}\mathbb{V}[(X_A - b)\nabla \log(\pi^\theta(A))] = 2\mathbb{E}[f(A)^2] \geq 0, \quad a.s.???$$

□

Exercises 3 Nr. 4
Show that given an MDP and a stationary policy it holds that

$$\begin{aligned}&\mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_0, A_0) = (s_0, a_0), ..., (S_t, A_t) = (s_t, a_t))) \\ =&\mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_t, A_t) = (s_t, a_t)))\end{aligned}$$

is time homogenuous Markov rewards chain with transition probability

$$p_{(s,a),(s',a',r')} = p(s', r'; s, a)\pi(a'; s').$$

Proof. The proof is anlogously to the Markov chain proof:

$$
\begin{aligned}
&\mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_t, A_t) = (s_t, a_t)))\\
&= \frac{\mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}), (S_t, A_t) = (s_t, a_t)))}{\mathbb{P}((S_t, A_t) = (s_t, a_t))}\\
&= \frac{\sum_{s_0,a_0,r_0} \cdots \sum_{s_{t-1},a_{t-1},r_{t-1}} \sum_{r_t} \mathbb{P}(S_0 = s_0, A_0 = a_0, R_0 = r_0, ..., S_{t+1} = s_{t+1}, A_{t+1} = a_{t+1}, R_{t+1} = r_{t+1})}{\sum_{s_0,a_0,r_0} \cdots \sum_{s_{t-1},a_{t-1},r_{t-1}} \sum_{r_t} \mathbb{P}(S_0 = s_0, A_0 = a_0, R_0 = r_0, ..., S_t = s_t, A_t = a_t, R_t = r_t)}\\
&= \frac{\sum_{s_0,a_0,r_0} \cdots \sum_{s_{t-1},a_{t-1},r_{t-1}} \sum_{r_t} \mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t} p(s_{i+1}, r_i; s_i, a_i)\pi(a_{i+1}; s_{i+1})p(\mathcal{S} \times \{r_{t+1}\}; s_{t+1}, a_{t+1})}{\sum_{s_0,a_0,r_0} \cdots \sum_{s_{t-1},a_{t-1},r_{t-1}} \sum_{r_t} \mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}, r_i; s_i, a_i)\pi(a_{i+1}; s_{i+1})p(\mathcal{S} \times \{r_t\}; s_t, a_t)}\\
&= \frac{\sum_{s_0,a_0,r_0} \cdots \sum_{s_{t-1},a_{t-1},r_{t-1}} \sum_{r_t} \mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}, r_i; s_i, a_i)\pi(a_{i+1}; s_{i+1})}{\sum_{s_0,a_0,r_0} \cdots \sum_{s_{t-1},a_{t-1},r_{t-1}} \mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}, r_i; s_i, a_i)\pi(a_{i+1}; s_{i+1})p(\mathcal{S} \times \{r_t\}; s_t, a_t)}\\
&\cdot p(s_{t+1}, r_t; s_t, a_t)\pi(a_{t+1}; s_{t+1})p(\mathcal{S} \times \{r_{t+1}\}; s_{t+1}, a_{t+1})\\
&= \frac{\sum_{r_t} p(s_{t+1}, r_t; s_t, a_t)\pi(a_{t+1}; s_{t+1})p(\mathcal{S} \times \{r_{t+1}\}; s_{t+1}, a_{t+1})}{\sum_{r_t} p(\mathcal{S} \times \{r_t\}; s_t, a_t)}\\
&= p(s_{t+1}, r_{t+1}; s_t, a_t)\pi(a_{t+1}; s_{t+1})p(\mathcal{S} \times \{r_{t+1}\}; s_{t+1}, a_{t+1})\\
&= \frac{\mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}; s_i, a_i)\pi(a_{i+1}; s_{i+1})}{\mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}; s_i, a_i)\pi(a_{i+1}; s_{i+1})}\\
&\cdot p(s_{t+1}, r_{t+1}; s_t, a_t)\pi(a_{t+1}; s_{t+1})p(\mathcal{S} \times \{r_{t+1}\}; s_{t+1}, a_{t+1})\\
&= \frac{\sum_{r_0,...,r_t} \mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}, r_i; s_i, a_i)\pi(a_{i+1}; s_{i+1})p(\mathcal{S} \times \{r_t\}; s_t, a_t)}{\sum_{r_0,...,r_t} \mu(s_0)\pi(a_0; s_0) \prod_{i=0}^{t-1} p(s_{i+1}, r_i; s_i, a_i)\pi(a_{i+1}; s_{i+1})p(\mathcal{S} \times \{r_t\}; s_t, a_t)}\\
&\cdot p(s_{t+1}, r_{t+1}; s_t, a_t)\pi(a_{t+1}; s_{t+1})p(\mathcal{S} \times \{r_{t+1}\}; s_{t+1}, a_{t+1})\\
&= \frac{\sum_{r_0,...,r_t} \mathbb{P}((S_0, A_0, R_0) = (s_0, a_0, r_0), ..., (S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}))}{\sum_{r_0,...,r_t} \mathbb{P}((S_0, A_0, R_0) = (s_0, a_0, r_0), ..., (S_t, A_t, R_t) = (s_t, a_t, r_t))}\\
&= \mathbb{P}((S_{t+1}, A_{t+1}, R_{t+1}) = (s_{t+1}, a_{t+1}, r_{t+1}) \mid (S_0, A_0) = (s_0, a_0), ..., (S_t, A_t) = (s_t, a_t)))
\end{aligned}
$$

where we used in $(*)$ that □

Exercises 4 Nr. 2

Exercises 4 Nr. 3

Exercises 5 Nr. 1

Exercises 5 Nr. 2

Exercises 5 Nr. 3

Exercises 6 Nr. 1

Exercises 6 Nr. 2

Exercises 6 Nr. 3

Exercises 7 Nr. 1

Exercises 8 Nr. 1

Exercises 8 Nr. 2

Exercises 8 Nr. 3

Exercises 9 Nr. 1

Exercises 9 Nr. 2

Exercises 10 Nr. 1

Exercises 10 Nr. 2

Exercises 10 Nr. 3

Exercises 10 Nr. 4

Exercises 10 Nr. 5

Exercises 11 Nr. 1

Exercises 11 Nr. 2

CHAPTER 7

DEEP REINFORCEMENT LEARNING

1 Neural Networks

...

2 Distributional Reinforcement Learning

Remember that we defined $\mathbb{P}^\pi_{s,a} := \mathbb{P}^\pi \otimes \delta_{S_0}(s) \otimes \delta_{A_0}(a)$ as the probability measure of the Markov reward process (S, A, R) started in (s, a). We define the random variable of the return under policy π as

$$Z^\pi := \sum_{t=0}^{\infty} \gamma^t R_t, \quad \gamma \in (0, 1).$$

Unlike the methods before, where we were interested in the expected reward $Q^\pi(s, a) = \mathbb{E}^\pi_{s,a}[Z^\pi]$, we are now interested in the distribution of these cumulative rewards. For that define

$$\eta^\pi_{s,a}(B) := \mathbb{P}^\pi_{s,a}(Z^\pi \in B), \quad B \in \mathcal{B}(\mathbb{R}).$$

In analogy to weak solutions for PDEs, i.e. in sense of distribution, the return distribution $\eta^\pi_{s,a}$ satisfies the Bellman equation in sense of distribution:

$$\forall \phi \in C_b(\mathbb{R}) : \quad \int_{\mathbb{R}} \phi(z) d\eta^\pi_{s,a}(z) = \mathbb{E}^\pi_{s,a}[\int_{\mathbb{R}} \phi(R + \gamma z) d\eta_{S',A'}(z)].$$

We define $f_{r,\gamma}(z) := r + \gamma z$ and the push forward

$$((\eta^\pi_{s,a})_{f_{r,\gamma}})(B) := \eta^\pi_{s,a}(f^{-1}_{r,\gamma}(B)), \quad B \in \mathcal{B}(\mathbb{R}).$$

then, the above can be written as

$$\begin{aligned} &\forall \phi \in C_b(\mathbb{R}) : \quad \int_{\mathbb{R}} \phi(z) d\eta^\pi_{s,a}(z) = \mathbb{E}^\pi_{s,a}[\int_{\mathbb{R}} \phi(R + \gamma z) d\eta_{S',A'}(z)] = \mathbb{E}^\pi_{s,a}[\int_{\mathbb{R}} \phi(z) d(\eta_{S',A'})_{f_{R,\gamma}}(z)] \\ \iff &\forall \phi \in C_b(\mathbb{R}) : \quad \langle \phi, \eta^\pi_{s,a} \rangle = \mathbb{E}^\pi_{s,a}[\langle \phi, (\eta^\pi_{S',A'})_{f_{R,\gamma}} \rangle]. \end{aligned}$$

For any $\phi \in C_b(\mathbb{R})$ and any measure ν we have

$$\begin{aligned} (\nu * \delta_r)(\phi) &= \int_{\mathbb{R}} \phi(z) d(\nu * \delta_r)(z) = \int_{\mathbb{R}} \int_{\mathbb{R}} \phi(x + y) d\nu(x) d\delta_r(y) \\ &= \int_{\mathbb{R}} \phi(x + r) d\nu(x) = (\nu(\cdot - r))(\phi). \end{aligned}$$

Next, define $D_\gamma(x) := \gamma x$, then if $Z \sim \eta^\pi(x,a)$ then $\gamma Z \sim (\eta^\pi(s,a))_{D_\gamma}$. In total

$$Z\gamma + r \sim (\eta^\pi(s,a))_{D_\gamma}(\cdot - r) = ((\eta^\pi(s,a))_{D_\gamma} * \delta_r)$$

holds in sense of distribution. Now define the distributional Bellman operator in sense of distribution as

$$\mathcal{T}^\pi Z^\pi(s,a) = R(s,a) + \gamma Z^\pi(S', A'),$$

i.e. the distribution

$$\mathcal{T}^\pi : (\mathcal{P}(\mathbb{R}))^{\mathcal{S}\times\mathcal{A}} \to (\mathcal{P}(\mathbb{R}))^{\mathcal{S}\times\mathcal{A}}, (\nu_{s,a})_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mapsto ((\mathcal{T}^\pi\nu)_{s,a})_{(s,a)\in\mathcal{S}\times\mathcal{A}}$$

is point wise defined for all $\phi \in C_b(\mathbb{R})$ as

$$(\mathcal{T}^\pi\nu)_{s,a}(\phi) := \mathbb{E}^\pi_{s,a}[((\nu(S',A'))_{D_\gamma} * \delta_R)(\phi)] = \sum_{s',a',r} \pi(a';s')p(s',r;s,a)((\nu_{s',a'})_{D_\gamma} * \delta_r)(\phi)$$

3 Deep Q-Networks

Deep Q-Networks (DQN) is a Q-learning variant that utilizes deep neural networks to approximate the Q-value. This algorithm addresses how to deal with large, continuous state action spaces $(\mathcal{S}, \mathcal{A})$. The idea is to reduce dimensionality by plugging in the state space $\mathcal{S}$ (e.g. pixels in an Atari game) into a convolutional neural layer, which is the fed into a feed forward neural network that outputs the Q-value for each action $a \in \mathcal{A}$. Problems with simple neural networks are

- They can forget
- Are unstable: small changes in Q-values can lead to big changes in the policy, i.e. the action distribution
- Correlation: In a trajectory τ_i of a rollout, the consecutive states s_t, s_{t+1} are highly correlated, which violates the i.i.d. assumption for estimating the expectation.
- The loss is non-stationary, since the target values depend on the parameters of the network itself, meaning
  $$\mathcal{L}(\theta) := \mathbb{E}[(\mathbb{E}[R + \gamma \max_{a'} Q^\theta(S', a')] - Q^\theta(S, A))^2],$$
  where both Q terms depend on θ.

In DQN these issues where solved by the following

- Experience Replay: A behavoiral policy π^b samples Rollouts, that are stored in a buffer $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$. In order to break the correlation between consecutive states, mini batches are sampled uniformly from this buffer to update the Q-network.
- Target Network: A second network with parameters θ^- is introduced, which is bootstrapped, i.e. every C steps the parameters of the Q-network are copied to the target network $\theta^- \leftarrow \theta$. This delay stabilizes the training.

Tabular Q-learning can be interpreted in the following two ways:

- ML view: Fit a target Q_n to the data Q^{π^*}. Define $Y_n := R + \gamma \max_{a'\in\mathcal{A}} Q_n(S', a')$ and
  $$\tilde{L}_n^Q := \mathbb{E}[(\mathbb{E}_{S,A}[Y_n] - Q_n(S,A))^2] = \mathbb{E}[(Y_n - Q_n(S,A))^2] + \mathbb{E}[\mathbb{V}_{S,A}(Y_n)] =: L_n^Q + \mathbb{E}[\mathbb{V}_{S,A}(Y_n)].$$

- RL view: We want to approximate the Bellman optimality operator

$$T^*Q(s,a) := \mathbb{E}_{s,a}[R + \gamma \max_{a' \in \mathcal{A}} Q(S', a')].$$

  Because it is a contraction in $\|\cdot\|_\infty$, the fixed point iteration converges.

Because the expectation is the minimizer of the L^2 error (variance), it holds that $\mathbb{E}[X] = \arg\min_\theta \mathbb{E}[(X-\theta)^2]$ which leads to

$$T^*Q(s,a) = \mathbb{E}_{s,a}[R + \gamma \max_{a' \in \mathcal{A}} Q(S', a')] = \arg\min_\theta \mathbb{E}_{s,a}[(R + \gamma \max_{a' \in \mathcal{A}} Q(S', a') - \theta)^2].$$

Thus doing one step SGD (with step size $\frac{\alpha}{2}$???) on L_n^Q or doing the iteration for Q-learning yields the same scheme:

$$Q_{n+1}(s,a) = Q_n(s,a) + \alpha(y - Q_n(s,a)).$$

Remark 3.1

When doing Q-learning with neural networks, we no longer do the gradient on $(s,a) \in \mathcal{S} \times \mathcal{A}$, but instead on a smaller parameter space $\Theta \subset \mathcal{S} \times \mathcal{A}$. Additionally, parameterizing the target with a different set of parameters θ^- the gradient of the loss becomes

$$\nabla_\theta L_n^Q = \mathbb{E}[\nabla_\theta (Y_n^{\theta^-} - Q_n^\theta(S,A))^2] = -\mathbb{E}[2(Y_n^{\theta^-} - Q_n^\theta(S,A)) \nabla_\theta Q_n^\theta(S,A)].??? -$$

So the gradient step for SGD becomes

$$Q_{n+1}^\theta(s,a) = Q_n^\theta(s,a) + \alpha(y_n^{\theta^-} - Q_n^\theta(s,a)) \nabla_\theta Q_n^\theta(s,a).$$

Remark 3.2

In a convolutional neural network a convolutional layer is given by

$$(f_k * h)(i) = \sum_{j=-n}^{n} f_k(i-j) h(i),$$

where f_i is a filter $k = 1, ..., K$ and h is some input vector. These K filters operator simultaninously to and produce $(f_k * h)(i), k = 1, ..., K$ as output. K is the number of Filters. In this setting the step size is called stride and was in the above 1. In general it is

$$(f_k * h)(i) = \sum_{j=-n}^{n} f_k(i+s-j) h(i), \quad s \in \mathbb{N}.$$

Remark 3.3

For Atari games the Architecture is as follows:

- Preprocessing: The environment 84×84 pixels is converted into grayscale. Let $x_t \in \mathbb{R}^{H \times W}$ be a greyscale image at time t, where $H = W = 84$. To create a sense of motion four consecutive images are stacked: $s_t := (x_t, ..., x_{t-3}) \in \mathbb{R}^{H \times W \times 4}$. Together with the scores, this is fed into the neural network.

- Convolutional Layers: There are three convolutional layers used. Each convolutional layers is followed by a ReLU activation function. The first layer has 32 filters of size 8×8 with stride 4. The second layer has 64 filters of size 4×4 with stride 2. The third layer has 64 filters of size 3×3 with stride 1.
- Feed Forward Layers: The output of the last convolutional layer is flattened and fed into a fully connected layer with 512 units, followed by a ReLU activation function.
- Output: The final layer in the network is a fully connected layer that outputs a vector of Q-values, one for each possible action in the Atari game.

Remark 3.4

- Experience Replay: This solves two problems: From the buffer $\mathcal{D}$ mini batches are chosen following an ϵ-greedy policy, which solves the problem of the correlation between rollouts and the policy. Further, it also solves the problem, that neural networks are prone to forget old data.
  If the buffer is full, the oldest transitions are removed first.
- Target Networks:
  - Different Parametrization: The target distribution is parametrized with θ^- Every N steps it is updated with a soft update rule

    $$\theta^- \leftarrow \tau\theta^- + (1-\tau)\theta.$$

    This stabilizes training:

    $$\tilde{L}_n(\theta) := \mathbb{E}[\Big(\mathbb{E}_{S,A}[Y_n^{\theta^-}] - Q_n^\theta(s,a)\Big)^2] = L_n(\theta) + \mathbb{E}[\mathbb{V}_{S,A}[Y_n^{\theta^-}]],$$

    because $\tilde{L}_n(\theta)$ no longer depends on the expected variance.
  - Update every N steps: With the second point of view of Q learning, updating every N steps is actually N step SGD of

    $$T^*Q^{\theta^-}(s,a) = \mathbb{E}_{s,a}[R + \gamma Q^{\theta^-}(S',a')] = \arg\min_\theta \mathbb{E}[(Y_n^{\theta^-} - \theta)^2].$$

Algorithm 19: The Deep Q-Network (DQN) algorithm.

Data: Replay buffer capacity $|D|$, discount factor γ, exploration rate ε, target update frequency N
Result: Trained Q-network approximating Q^*
Initialise replay buffer D
Initialise Q-network with random weights θ
Copy weights to target network: $\theta^- \leftarrow \theta$
for *each episode* **do**
  Pre-process the initial observation s_1 to obtain state representation ϕ_1
  for *each time-step* t **do**
    Select action a_t using ε-greedy policy based on $Q(\phi_t, a; \theta)$
    Execute action a_t, observe reward r_t and next observation s_{t+1}
    Pre-process s_{t+1} to obtain ϕ_{t+1}
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in replay buffer D
    Sample mini-batch B of transitions from D
    for *each transition* $i \in B$ **do**
      Compute target:

$$Y_i = \begin{cases} r_i, & \text{if terminal transition,} \\ r_i + \gamma \max_{a'} Q(\phi_i', a'; \theta^-), & \text{otherwise.} \end{cases}$$

    end
    Perform gradient descent on loss:

$$L_B(\theta) = \frac{1}{|B|} \sum_{i \in B} (Y_i - Q(\phi_i, a_i; \theta))^2$$

    Update Q-network weights θ
    if *every* N *steps* **then**
      Update target network: $\theta^- \leftarrow \theta$
    end
    However, target networks increase memory, as it another neural network needs to be stored and updating the weights increases computational costs.
  end
end

Remark 3.5

- Double Q-learning and Double DQN: Q-learning over estimates the action values due to the max. operator, In Double Q-learning we maintain two independent networks Q^A and Q^B to estimate Q^{π^*}:

$$\begin{aligned} Q^A(s,a) &\leftarrow (1-\alpha) Q^A(s,a) + \alpha \left(r + \gamma Q^B(s', \arg\max_{a' \in \mathcal{A}} Q^A(s', a')) \right) \\ &= (1-\alpha) Q^A(s,a) + \alpha \left(r + \gamma Q^A(s', \arg\max_{a' \in \mathcal{A}} Q^A(s', a')) \right) \\ &+ \gamma \left(Q^B(s', \arg\max_{a' \in \mathcal{A}} (s', a')) - Q^A(s', \arg\max_{a' \in \mathcal{A}} Q^A(s', a')) \right). \end{aligned}$$

In Double Q-learning the Networks are updated in every step and independendly. In Double DQN we give one network the role of the target network and the other the Q-network. The Q-network is updated every step. The target network is updated only every N step with the soft update rule, which depends on the Q-network. So indepence and simultaninous updates do not happen.

- Clipped Double Q/TD3: Another approach is to clip the bias term (third summand in the above) by taking its negative part: $b^- := \min\{b, 0\}$. The deep variant of this method is called Twin delayed double Deep Q Networks.

4 Proximal Policy Gradient

5 Advantage Estimation

6 Variants to PPO

7 Stable Baselines3

CHAPTER A

5 MINUTEN ÜBER FRAKTALE DER VALUE FUNCTION

Definition 0.1: δ-Cover

Let $(U_i)_{i\in\mathbb{N}}$ a sequence of sets that have a diameter of at most $\delta > 0$, i.e. $|U_i| := \sup\{\|x-y\| \mid x, y \in U_i\} \leq \delta$ for all $i \in \mathbb{N}$. For $F \subset \mathbb{R}^N$ we call $(U_i)_{i\in\mathbb{N}}$ $\delta-$cover if $F \subset \bigcup_{i\in\mathbb{N}} U_i$.

Definition 0.2

For any $F \subset \mathbb{R}^N$ and $s \geq 0$ we define

$$\mathcal{H}^s_\delta(F) := \inf\{\sum_{i\in\mathbb{N}} |U_i|^s \mid F \subset \bigcup_{i\in\mathbb{N}} U_i\}.$$

We call $\mathcal{H}^s(F) := \lim_{\delta\to 0} \mathcal{H}^s_\delta(F)$ the Hausdorff Measure of F.

Definition 0.3

For any $F \subset \mathbb{R}^N$, we call

$$dim_H(F) := \inf\{s \geq 0 \mid \mathcal{H}^s(F) = 0\} = \sup\{s \geq 0 \mid \mathcal{H}^s(F) = \infty\}$$

its Hausdorff Dimension.

Proof. of the result in the definition: Common in Literature □

Definition 0.4

we define a function $\eta : \mathbb{R}^d \to \mathbb{R}$ which we call $\alpha-$Hölder continuous at $x \in \mathbb{R}^d$ if

$$\exists C > 0 \epsilon > 0 \forall y \in B_\epsilon(x) : \quad \|\eta(x) - \eta(y)\| \leq C\|x-y\|^\alpha.$$

It is Lipschitz if $\alpha = 1$. Consider a deterministic MDM

$$s_{t+1} = f(s_t, a_t)$$

that is continuous on $\mathcal{A}$, $\mathcal{S} \subseteq \mathbb{R}$ (which are usually compact). Our reward is deterministically defined as $r(s,a) = |s|$ and the policy $\pi^\theta(s) := s\theta$ for $\theta > 1$. Define the transition function as

$$f(s,a) = \begin{cases} 1 & , a \geq 1 \\ a & , a \in (0,1) \\ -1 & , a \leq -1. \end{cases}$$

Then the value function started in $1 > \delta > 0$ with $T_0(1)$ the first hitting time of state 1 (because $r_t = |s_t| = \theta^t\delta$ is a non decreasing sequence and stays constant after reaching state 1) satisfies

$$V^{\pi^\theta}(\delta) = \sum_{t=0}^{\infty} \gamma^t r_t = \sum_{t=0}^{\infty} \gamma^t(\theta^t\delta \mathbf{1}_{t\leq T_0(1)} + \mathbf{1}_{t>T_0(1)}) = \sum_{t=0}^{T_0(1)} (\gamma\theta)^t\delta + \sum_{t=T_0(1)+1}^{\infty} \gamma^t \geq \sum_{t=T_0(1)+1}^{\infty} \gamma^t = \frac{\gamma^{T_0(1)}}{1-\gamma}$$

where $T_0(1) = 1 + \lfloor \frac{-\log(\delta)}{\log(\theta)} \rfloor$ (Solve $\theta^t\delta < 1$ and $1 \leq \theta^{t+1}\delta$). Thus the value function is in zero $\frac{-\log(\gamma)}{\log(\theta)}$ continuous:

$$\begin{aligned} |V^{\pi^\theta}(\delta) - V^{\pi^\theta}(0)| = V^{\pi^\theta}(\delta) &\geq \frac{\gamma^{1+\lfloor \frac{-\log(\delta)}{\log(\theta)} \rfloor}}{1-\gamma} \geq \frac{\gamma}{1-\gamma}\gamma^{\frac{-\log(\delta)}{\log(\theta)}} \\ &= \frac{\gamma}{1-\gamma} e^{\log(\gamma)\frac{-\log(\delta)}{\log(\theta)}} = \frac{\gamma}{1-\gamma} e^{-\log(\gamma)\frac{\log(\delta)}{\log(\theta)}} \\ &= \frac{\gamma}{1-\gamma} e^{\log(\delta)\frac{-\log(\gamma)}{\log(\theta)}} = \frac{\gamma}{1-\gamma} \delta^{\frac{-\log(\gamma)}{\log(\theta)}} \end{aligned}$$

Finally, in the paper it was shown that $V^{\pi^\theta}(\delta)$ cannot be p-hölder smooth for $p > \frac{-\log(\gamma)}{\log(\theta)}$.

Theorem 0.5

An α-Hölder continuous function $\eta : \mathbb{R}^k \to \mathbb{R}$ satisfies for

$$Graph(\eta) := \{(x, \eta(x)) \in \mathbb{R}^{k+1} \mid x \in \mathbb{R}^k\}$$

that $dim_H Graph(\eta) \leq 2 - \alpha$.

Proof. Common in Literature. □

Thus if we choose γ and θ such that $0 < \frac{-\log(\gamma)}{\log(\theta)} < 1$ the value function is a fractal in zero .

We have $Y_1, ..., Y_n$ iid samples of a random variable $X \sim F$, then the least squares minimizer of linear regression is give by

$$\hat{\beta} = (X^T X)^{-1} Y^T Y = \frac{1}{n}\sum_{i=1}^{n} Y_i, \quad X \in \mathbb{R}^{n\times 1}$$

i.e. anonlogusly to Monte Carlo. Assume further we want to do regression on a discrete set $\mathcal{A}$ then this is hardly possible, as there are no distances defined. What we can do is do regression on the Psuedo-inverse on $[0,1]$. We define

$$W_p(U,Y) = (\int_0^1 |F_Y^{-1}(\omega) - F_U^{-1}(\omega)|^p d\omega)^{1/p}.$$

Then

1 Paar Ideen

Let Φ_a be a PPP on $\mathbb{R}$ with intensity measure $\Lambda_a(dt) := e^{Q_a\theta_a}e^{-t}dt$ (locally finite). Then $\Phi := \sum_{a=1}^K \Phi_a$ is a PPP with intensity measure $\Lambda(dt) := \sum_{a=1}^K \Lambda_a(dt)$. Further define $X := \max supp(\Phi)$. Then

$$\begin{aligned}
\mathbb{P}(X \leq x) = \mathbb{P}(\Phi(x,\infty) = 0) &= e^{-\Lambda((0,\infty))} \\
&= e^{-\sum_{a=1}^K e^{Q_a\theta_a}\int_x^\infty e^{-t}dt} \\
&= e^{-\sum_{a=1}^K e^{Q_a\theta_a}e^{-x}} \\
&= e^{-e^{-x+\log(\sum_{a=1}^K e^{Q_a\theta_a})}} \\
&= e^{-e^{-(x-\log(\sum_{a=1}^K e^{Q_a\theta_a}))}} \\
&= \mathbb{P}(Y \leq x - \log(\sum_{a=1}^K e^{Q_a\theta_a})), \quad Y \sim Gumbel(0,1)
\end{aligned}$$

Further let the joint distribution (X, A) is given by

$$f_{X,A}(x,a) = \Lambda_a(dx)\cdot\mathbb{P}^X(dx) = e^{Q_a\theta_a}e^{-x}\cdot e^{-\sum_{a=1}^K e^{Q_a\theta_a}e^{-x}}.$$

Then

$$\begin{aligned}
\mathbb{P}(A = a \mid X = x) &= \frac{f_{X,A}(x,a)}{\sum_{b=1}^K f_{X,A}(x,b)} \\
&= \frac{e^{Q_a\theta_a}e^{-x}\cdot e^{-\sum_{a=1}^K e^{Q_a\theta_a}e^{-x}}}{\sum_{b=1}^K e^{Q_b\theta_b}e^{-x}\cdot e^{-\sum_{c=1}^K e^{Q_c\theta_c}e^{-x}}} \\
&= \frac{e^{Q_a\theta_a}}{\sum_{b=1}^K e^{Q_b\theta_b}}.
\end{aligned}$$

Finally due to

$$\begin{aligned}
\mathbb{P}(A = a) &= \int_{\mathbb{R}} f_{X,A}(x,a)dx \\
&= e^{Q_a\theta_a}\int_{\mathbb{R}} e^{-x}\cdot e^{-\sum_{a=1}^K e^{Q_a\theta_a}e^{-x}}dx \\
&\overset{(*)}{=} e^{Q_a\theta_a}\int_\infty^0 \frac{u}{\sum_{a=1}^K e^{Q_a\theta_a}}e^{-u}(-\frac{du}{u}) \\
&= \frac{e^{Q_a\theta_a}}{\sum_{a=1}^K e^{Q_a\theta_a}}
\end{aligned}$$

although we used in $(*)$ that $u := \sum_{a=1}^K e^{Q_a\theta_a}e^{-x}$ and $du = -u\,dx$ which is equivalent to $dx = \frac{-du}{u}$ and finally the fact that $x \to -\infty \quad \Rightarrow \quad u \to \infty$ and $x \to +\infty \quad \Rightarrow \quad u \to 0$.
This implies that

$$\mathbb{P}(A = a \mid X = x) = \mathbb{P}(A = a).$$

Thus simulating the PPP Φ with intensity Λ we can get a softmax distribution if we define $X := \max supp(\Phi)$ and consider the marginal distribution $\mathbb{P}^A$ of the joint distribution $\mathbb{P}^{(X,A)}$.

Where does the Gumble come from? If we consider $X_a := \max supp(\Phi_a)$ then

$$\begin{aligned}
\mathbb{P}(X_a \leq x) = \mathbb{P}(\Phi(x,\infty) = 0) &= e^{-\Lambda_a((x,\infty))} \\
&= e^{-e^{Q_a\theta_a}\int_x^\infty e^{-t}dt} \\
&= e^{-e^{Q_a\theta_a}e^{-x}} \\
&= e^{-e^{-(x-Q_a\theta_a)}} \\
&= \mathbb{P}(Y \leq x - Q_a\theta_a), \quad Y \sim Gumbel(0,1).
\end{aligned}$$