Report RL tutorial 1

Antoine Gorceix

November 17, 2025

Party I

Exercise 1: Simple MDP

(i)

Policy: Because all rewards before reaching G are 0 and $r > 0$ is obtained *only* in G, the agent should minimize time-to-goal. Hence, for every s_i the optimal action moves *right* to s_{i+1}. In G any action is optimal (both keep you in G). This policy does *not* depend on γ for any $\gamma \in (0, 1)$ (earlier arrival is always weakly better).

Values: Let $d_i \triangleq n - i$ be the number of right moves from s_i to reach G. Since intermediate rewards are 0,

$$V^\star(G) = r + \gamma r + \gamma^2 r + \cdots = \frac{r}{1-\gamma}, \qquad V^\star(s_i) = \gamma^{d_i}\, V^\star(G) = \gamma^{n-i}\,\frac{r}{1-\gamma}, \quad i = 1, \ldots, n-1.$$

(ii)

Let $r'_t = r_t + \beta$ for every state–action and time step. For any policy π,

$$V'_\pi(s) \;=\; \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t (r_t + \beta) \mid s_0 = s\right] = V_\pi(s) + \frac{\beta}{1-\gamma}.$$

Therefore the *optimal* values are shifted uniformly:

$$V^{\star\prime}(s) \;=\; V^\star(s) + \frac{\beta}{1-\gamma} \quad \text{for all } s,$$

and the optimal policy is unchanged (adding a constant does not affect action comparisons).

(iii)

Let $r''_t = \alpha(r_t + \beta) = \alpha r_t + \alpha\beta$ with constant $\alpha \in \mathbb{R}$. For any policy π,

$$V''_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t \big(\alpha r_t + \alpha\beta\big) \mid s_0 = s\right] = \alpha V_\pi(s) + \frac{\alpha\beta}{1-\gamma}.$$

Hence the optimal values satisfy

$$V^{\star\star}(s) \;=\; \alpha V^\star(s) + \frac{\alpha\beta}{1-\gamma}, \quad \forall s.$$

Policy effect.

- If $\alpha > 0$: the optimal policy is unchanged (positive affine transform preserves argmax).
- If $\alpha = 0$: all policies are optimal and $V^{\star\star}(s) = \frac{\alpha\beta}{1-\gamma}$ for all s.
- If $\alpha < 0$: action preferences are reversed; since $r > 0$ in G, the agent prefers *avoiding* G (move left forever from non-goal states).

Exercise 2: Policy evaluation

Consider a finite MDP with state set $S = \{s_1, \dots, s_n\}$ and actions A. For an infinite-horizon discounted problem and a fixed policy π, the Bellman equation reads

$$V^\pi(s) = \sum_{a\in A} \pi(a \mid s) \sum_{s'\in S} P(s' \mid s, a)\big[R(s', s, a) + \gamma V^\pi(s')\big], \qquad 0 < \gamma < 1. \tag{1}$$

(i)

Matrix form: Define vectors and a matrix indexed by the states s_i:

$$v_i^\pi \triangleq V^\pi(s_i), \qquad r_i^\pi \triangleq \sum_{a\in A} \pi(a \mid s_i) \sum_{s'\in S} P(s' \mid s_i, a)\, R(s', s_i, a),$$

$$P_{ij}^\pi \triangleq \sum_{a\in A} \pi(a \mid s_i)\, P(s_j \mid s_i, a).$$

Let $\mathbf{v}^\pi = (v_1^\pi, \dots, v_n^\pi)^\top$, $\mathbf{r}^\pi = (r_1^\pi, \dots, r_n^\pi)^\top$ and $P^\pi = [P_{ij}^\pi] \in \mathbb{R}^{n\times n}$. Stacking for all states yields the linear system

$$\mathbf{v}^\pi = \mathbf{r}^\pi + \gamma P^\pi \mathbf{v}^\pi. \tag{2}$$

(ii)

Row-stochasticity of P^π: For every i,

$$\sum_{j=1}^n P_{ij}^\pi = \sum_a \pi(a \mid s_i) \sum_{s'} P(s' \mid s_i, a) = \sum_a \pi(a \mid s_i) \cdot 1 = 1,$$

and $P_{ij}^\pi \geq 0$. Hence P^π is row-stochastic. Consequently,

$$\|P^\pi\|_\infty = \max_i \sum_j |P_{ij}^\pi| = 1, \qquad \text{and} \qquad \rho(P^\pi) \leq \|P^\pi\|_\infty = 1,$$

so every eigenvalue $\lambda_i(P^\pi)$ satisfies $|\lambda_i| \leq 1$.

Invertibility of $I - \gamma P^\pi$: Because $0 < \gamma < 1$, we have $\rho(\gamma P^\pi) \leq \gamma\, \rho(P^\pi) \leq \gamma < 1$ and $\|\gamma P^\pi\|_\infty = \gamma < 1$. Therefore $I - \gamma P^\pi$ is nonsingular and admits the Neumann-series inverse

$$(I - \gamma P^\pi)^{-1} = \sum_{k=0}^\infty (\gamma P^\pi)^k.$$

Thus the unique solution of the equation is

$$\mathbf{v}^\pi = (I - \gamma P^\pi)^{-1} \mathbf{r}^\pi. \tag{3}$$

Exercise 3: Policy and Value iteration

(i)

Claim (discounted, infinite-horizon): For a finite MDP with time-homogeneous ("static") transitions, bounded rewards and discount $\gamma \in (0, 1)$, there exists an *optimal stationary deterministic* policy $\pi^\star$; i.e., there is a mapping $s \mapsto a^\star(s)$ such that $V^{\pi^\star}(s) = V^\star(s)$ for all s.

Why deterministic? Let T be the Bellman optimality operator

$$(TV)(s) \;=\; \max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \right].$$

Let $V^\star$ be its unique fixed point. For each s, choose

$$a^\star(s) \in \arg\max_a \left[R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V^\star(s') \right].$$

The greedy policy $\pi^\star(s) = a^\star(s)$ is deterministic and satisfies $T^{\pi^\star} V^\star = TV^\star = V^\star$, hence $V^{\pi^\star} = V^\star$. Randomization is unnecessary because the right-hand side is a maximum of finitely many *linear* functions of the action distribution, attained at an extreme point (a Dirac mass).

Can an optimal policy be stochastic? Yes, but only in a degenerate sense: if multiple actions tie for the maximum in some state s (i.e., $Q^\star(s,a)$ is equal for several a), then *any* distribution supported on those maximizers is also optimal in s. A stochastic policy is never strictly better than choosing one of the maximizers deterministically.

(iii)

Let T^π be the policy evaluation operator

$$(T^\pi V)(s) \;=\; \sum_a \pi(a|s) \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V(s') \Big].$$

Suppose π' is greedy with respect to Q^π, i.e.

$$\sum_a \pi'(a|s)\, Q^\pi(s,a) \;=\; \max_a Q^\pi(s,a) \qquad \forall s.$$

Since $Q^\pi(\cdot,\cdot) = R + \gamma P(\cdot|\cdot,\cdot)\, V^\pi$ and $V^\pi = T^\pi V^\pi$, we get

$$(T^{\pi'} V^\pi)(s) = \sum_a \pi'(a|s) \Big[R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s') \Big] = \max_a Q^\pi(s,a) \;\geq\; \sum_a \pi(a|s) Q^\pi(s,a) = (T^\pi V^\pi)(s) =$$

Thus $T^{\pi'} V^\pi \;\geq\; V^\pi$ componentwise. Because $T^{\pi'}$ is a γ-contraction and monotone (order-preserving),

$$V^{\pi'} \;=\; \lim_{k\to\infty} (T^{\pi'})^k V^\pi \;\geq\; V^\pi.$$

Moreover, if the inequality is strict for some state, then $V^{\pi'}(s) > V^\pi(s)$ for all states reachable under π' from that state. This proves the policy improvement theorem.

Exercise 4: Control without a model

(i)

- **Monte Carlo (MC):** The MC target is the (sampled) full return and does not bootstrap from $\hat{V}$. Hence, for episodic tasks and any fixed policy π,

  $$\mathbb{E}\Big[\hat{R}_t^{\mathrm{MC}} \mid s_t \Big] = V^\pi(s_t),$$

  so the target is *unbiased*. However it aggregates randomness from all future rewards and transitions up to termination, which yields *high variance*, especially for long horizons (large H) or γ close to 1.

- **TD(0):** The TD target *bootstraps* through $\hat{V}^\pi(s_{t+1})$. Conditional on (s_t, a_t, s_{t+1}) it depends only on r_t and one value lookup, so its variability is much smaller: TD(0) has *lower variance*. But because it uses an imperfect estimate $\hat{V}^\pi$, the target is generally *biased*.

(ii)

Target vs. behaviour: The *target policy* π is the policy we wish to evaluate or improve. The *behaviour policy* μ is the policy that actually generates data. In on-policy control we take $\mu = \pi$; in off-policy methods, $\mu \neq \pi$.

Need for ε-greedy: Exploration is necessary so that all relevant state–action pairs are sampled often enough to learn their values and to avoid getting stuck with a suboptimal deterministic policy. An ε-greedy policy guarantees nonzero probability for every action:

$$\mu(a \mid s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|\mathcal{A}(s)|}, & a = \arg\max_{a'} Q(s, a'), \\ \frac{\varepsilon}{|\mathcal{A}(s)|}, & \text{otherwise}, \end{cases}$$

ensuring coverage.

When can a policy be used as behaviour?

- **On-policy (e.g. SARSA/MC control):** The behaviour must explore *infinitely often*. A standard sufficient condition is GLIE (Greedy in the Limit with Infinite Exploration): every (s, a) is visited infinitely many times while $\varepsilon_t \downarrow 0$ (e.g. $\sum_t \varepsilon_t = \infty$ and $\varepsilon_t \to 0$), so learning remains exploratory yet becomes greedy asymptotically.
- **Off-policy (importance sampling / TD with corrections):** The behaviour must have *support* for the target: $\mu(a \mid s) > 0$ whenever $\pi(a \mid s) > 0$. This guarantees well-defined importance ratios and finite-variance estimates under additional regularity.

(iii) Why is Q-learning off-policy? Advantages of off-policy methods

Off-policy nature: Q-learning updates

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \Big[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \Big].$$

The TD target uses the *greedy* action in s_{t+1} (the implicit target policy is $\pi^\star(a \mid s) = \mathbf{1}\{a \in \arg\max_{a'} Q(s, a')\}$), independent of the action actually *taken* under the behaviour policy μ (which may be ε-greedy for exploration). Hence Q-learning learns about the greedy/optimal policy while behaving according to a different, exploratory policy—it is *off-policy*.

Advantages of off-policy methods:

- Learn an optimal (often deterministic) target policy while collecting data with a safer or more exploratory behaviour policy.
- Reuse logged data generated by other policies (e.g. human demonstrations, historical logs) for both evaluation and control.
- Decouple exploration from the final policy: easier to satisfy safety constraints or business rules during data collection.
- Facilitate learning multiple target policies from the same experience stream (policy evaluation with different π).

Exercise 5: Importance Sampling

(i)

A sufficient (and standard) set of conditions is:

(a) *Support/absolute continuity:* p is absolutely continuous w.r.t. q, i.e. $q(x) > 0$ whenever $p(x)|f(x)| > 0$.

(b) *Integrability:* $\mathbb{E}_q\left[\left|f(X)\,\frac{p(X)}{q(X)}\right|\right] < \infty$.

Under (a)–(b) and i.i.d. sampling from q,

$$\mathbb{E}[\hat{s}_q] = \mathbb{E}_q\left[f(X)\frac{p(X)}{q(X)}\right] = \int f(x)\frac{p(x)}{q(x)}q(x)\,dx = \int f(x)p(x)\,dx = s_p,$$

so $\hat{s}_q$ is unbiased.

Common problems:

- **Huge / infinite variance.** If q puts too little mass where $|f|p$ is large (e.g., lighter tails than $|f|p$), the weights $w(x) = p(x)/q(x)$ explode and $Var(\hat{s}_q)$ can be enormous or infinite.
- **Weight degeneracy.** A few samples carry almost all the weight (tiny effective sample size).
- **Numerical instability.** With heavy-tailed weights, sums overflow/underflow in finite precision.

Typical remedies:

- **Choose a better proposal:** make q close to $q^\star(x) \propto |f(x)|p(x)$ and with tails at least as heavy as $|f|p$; use mixtures or multiple importance sampling (MIS).
- **Defensive mixture:** $q_\varepsilon = (1-\varepsilon)q + \varepsilon p$ which bounds weights by $1/\varepsilon$.
- **Variance reduction:** control variates, stratified sampling, or adaptive / population IS to tune q.

Party II

Exercise 6: Value Iteration

- **Implementation:** Maintain a sparse value table $V : \mathcal{S} \to \mathbb{R}$ (e.g., `Counter`) and perform synchronous Bellman updates

$$V_{k+1}(s) = \max_a \sum_{s'} P(s' \mid s, a)\left(R(s, a, s') + \gamma\, V_k(s')\right)$$

  for all s, using a temporary copy for V_{k+1} each iteration.

- **Initialization:** Use $V_0(s) = 0$ for all states (default of `Counter`); in particular, $V_0(s_{\text{terminal}}) = 0$.
- **Unknown state space:** Do lazy expansion: start at s_0 and, when transitions reveal new s', insert them into V; unseen states implicitly have value 0 until updated.

Exercise 7: Cliffworld MDP

- **Goal/Strategy:** The agent should follow a policy that maximizes expected discounted return. In Cliff World this means comparing short, risky routes that pass near the cliff versus longer, safer routes. A risky route is optimal only if its *expected* discounted value exceeds that of the safe alternative.

- **Effect of living reward r (non-terminal reward):** Negative r makes time costly $\Rightarrow$ reach an exit quickly (often the closest). Zero r leaves only terminal payoffs and risks to trade off. Positive r rewards lingering; with large enough r and high γ, the agent may prefer to *avoid exiting* altogether.

- **Effect of discount factor γ:** Small γ (myopic) favors short paths and the close +1 exit. Large γ (far-sighted) values the distant +10 more and tolerates longer detours. Because cliff penalties occur soon, increasing γ also *increases* the weight of near-term risks relative to far rewards, pushing policies away from risky edges for fixed noise.

- **Effect of randomness/noise n:** Higher action noise raises the chance of unintended sideways moves near the cliff, increasing expected losses; optimal policies shift to safer corridors (e.g., along the top) and add extra clearance from hazards. With very high noise:

  - if $r < 0$, the agent still rushes to an exit to avoid accruing costs;
  - if $r > 0$ and γ is large, the agent may prefer wandering indefinitely (avoiding both exits and the cliff).

Exercise 8: Pacman MDP

- **Q-table & invalid actions:** Store Q in a hash map (e.g., `Counter`) keyed by (s, a) with default 0, created lazily. When acting or computing $\max_a Q(s, a)$, restrict a to `getLegalActions`(s); if none exist, return value 0 and action `None`. Never store/update invalid actions.
- **Why larger grids fail:** Tabular Q-learning requires visiting many (s, a) pairs; $|S|$ explodes on bigger layouts, rewards are sparse, and memory/time blow up $\Rightarrow$ slow or no convergence and poor generalization.
- **How to scale:** Use function approximation $Q(s, a) \approx w^\top \phi(s, a)$ (`ApproximateQAgent`/DQN), state abstraction/tiling and informative features, eligibility traces or n-step updates, stronger exploration (decaying ϵ, optimism/bonus/UCB), reward shaping, and (optionally) experience replay/prioritized sampling.