MDP and RL: Q-learning, stochastic approximation – some exercises

Nicolas Gast

October 1, 2024

1 Stochastic approximation

1.1 Generalized Polya urn

We consider an urn that contains N balls that are black or red. At each time step, we pick three balls from the bin (at random, with replacement[1]) and observe their color: If there are two or more balls of the same color (say red), we add another ball of the same color and replace all balls in the urn.

Let x_n be the fraction of red balls after n such events.

1. Show that x_n can be written as a stochastic approximation algorithm: $x_{n+1} = x_n + \alpha_n(f(x_n) + M_{n+1})$. What is the step-size n and the function f?

2. Describe the trajectories of the ODE $\dot{x} = f(x)$ (*i.e.*, its limits points).

3. Use that to describe the long term behavior of x_n when n goes to infinity.

To go further: *Limit distributions for large Pólya urns* by Brigitte Chauvin, Nicolas Pouyanne, Reda Sahnoun. Ann. Appl. Probab.

1.2 Stochastic gradient descent and Kiefer–Wolfowitz Procedure

Let $f : \mathbb{R}^d \to \mathbb{R}$ where $f(\theta) = \mathbb{E}\left[F(\theta, X)\right]$ for some random variable X, and let us assume that F is differentiable with a continuous derivative. Let x_{n+1} be defined as:

$$x_{n+1} = x_n + \alpha_n \nabla F(x_n, Z_n), \tag{1}$$

where Z_n is a sequence of *i.i.d.* noise.

1. Show that (1) can be represented as a stochastic approximation algorithm.

[1] A more natural model would be to pick the balls with replacement. This would complexify the computations because instead of f, one would have $x_{n+1} = x_n + \alpha_n(f(x_n) + M_{n+1} + \varepsilon_n)$ with $\varepsilon_n = O(1/n)$. One could show that it does not change much the analysis.

2. Assume that f is strictly convex, and has a local minimum x^*. Show that all trajectories of the ODE $\dot{x} = \nabla f(x)$ converge to x^*.

3. Assume in addition that x_n stays bounded almost surely and that α_n satisfy the Robins-Monroe conditions, and that the co-variance of $\nabla F()$ is bounded. Show that x_{n+1} converge almost surely to x^*.

The Kiefer–Wolfowitz Procedure. We suppose now that we do not have access to ∇F. We consider the following procedure:

$$\begin{aligned} G_{n+1,i} &= \frac{1}{2\epsilon_n}\left[F(y_n + \epsilon_n e_i, Z_n) - F(y_n - \epsilon_n e_i, Z_n)\right], \\ y_{n+1} &= y_n + \alpha_n G_{n+1}. \end{aligned}$$

where e_i is a the unit-vector with a 1 on the ith coordinate.

4. Show that G_{n+1} is a noisy estimate of the gradient $\nabla f(y_n)$. Compute its bias and is variance.

5. Assume that $\epsilon_n = n^{-1/3}$ and $\alpha_n = n^{-1/2}$. Show that x_{n+1} converge to x^* almost surely.

To go further: Book of *Kunsher-Yin* (Stochastic approximation and recursive algorithms.)

2 Concentration inequalities

2.1 An unbiased estimators with limited observations

Let $X_1 \dots X_n$ be a sequence of n random variables and let $p = (p_1 \dots p_n)$ be a probability distribution over n variables (*i.e.* $p_i > 0$ and $\sum_{i=1}^n p_i = 1$). Consider the following procedure:

- Sample I according to the distribution p and observe X_I.
- Set $\tilde{X}_k$ as:

$$\tilde{X}_k = \begin{cases} \frac{X_I}{p_I} & \text{if } k = I \\ 0 & \text{otherwise.} \end{cases}$$

1. Show that $\tilde{X}_k$ is an *unbiased* estimator of $\mathbb{E}\left[X_k\right]$ for all k.

2. Assume that $0 \le X_k \le 1$ for all k. Compute a bound on the variance of the estimator, *i.e.*, $\mathrm{var}(\tilde{X}_k)$.

Going further: This exercise show that one can bound an estimator of n variables by observing one variable. See the lecture about bandits.

2.2 Hoeffding inequality

Let $(X_n)_{n\geq 0}$ be a sequence of *i.i.d.* random variable such that $X_n \in [-1,1]$ almost surely and that $\mathbb{E}[X_n] = 0$. Let $S_n = \sum_{k=1}^{n} X_k$. We want to show that for all $x > 0$:

$$\mathbb{P}(S_n > t) \leq \exp(-\frac{t}{2n})$$

1. Prove that $\mathbb{P}(S_n > x) \leq e^{-\lambda t}(\mathbb{E}\left[e^{\lambda X_1}\right])^n$.
2. By using convexity[2], prove that $\mathbb{E}\left[e^{\lambda X_1}\right] \leq e^{\lambda^2/2}$.
3. Conclude (hint: find the best λ)

Going further: https://en.wikipedia.org/wiki/Hoeffding%27s_inequality

2.3 Markov / Chebyshev inequality for Martingale

Let X_n be a sequence of random variables such that $\mathbb{E}[X_{n+1}|\mathcal{F}_n] = 0$ and $\mathrm{var}(X_{n+1}|\mathcal{F}_n) = \sigma_n^2 < \infty$. Let $S_n = \sum_{k=1}^{n} X_k$.

Let $a > 0$ and define

$$M_{k+1} = \begin{cases} S_k + X_k & \text{if } M_k \geq a \\ a & \text{if } M_k = a \end{cases}$$

1. Show that $\mathbb{P}(\sup_{1\leq k\leq n} S_k \geq a) = \mathbb{P}(M_k \geq a)$.
2. Show that $\mathbb{E}[\max(M_n, 0)] \leq \mathbb{E}[\max(S_n, 0)]$.
3. Use to conclude that $\mathbb{P}(\sup_{1\leq k\leq n} S_k \geq a) \leq \mathbb{E}[\max(S_n, 0)]/a$.
4. Prove that $\mathrm{var}(S_n) = \sum_{k=1}^{n} \sigma_k^2$ and use it to conclude thématique

$$\mathbb{P}(\sup_{1\leq k\leq n} S_k \geq a) \leq \frac{\sum_{k=1}^{n} \sigma_k^2}{a^2}.$$

3 MDPs

3.1 A game of dice: "stop or continue"

Consider an unbiased d face dice with faces numbered from 1 to d. You can throw the dice up to d times (each time with a random throw). You gain money as follows:

- After any throw, you can stop and earn the sum of the values that you obtained.
- If, after a throw, you obtain a value that you have already observed, you earn nothing.

[2] This is quite technical. To show that, one can show that $\mathbb{E}\left[e^{\lambda X_1}\right] \leq (e^{\lambda} - e^{-\lambda})/2$ by convexity, and that $g(\lambda) = \log((e^{\lambda} - e^{-\lambda})/2) \leq \lambda^2/2$ by showing that $g''(\lambda) \leq 1$.

For instance, if your sequence of throws is “1, 3”, you can stop and earn 4 or throw again. If you throw again: if you obtain a “2”, you can stop and earn 6 or throw again, but if you obtain “3”, you have to stop and earn nothing.

1. Consider $d = 3$. Formulate the problem as a MDP, compute the optimal policy.
2. Can you guess the optimal policy for any d? Hint: you can show that the problem is monotone (*i.e.*, show that if it is optimal to stop in a given state, then it is also optimal to stop in any state that is “larger” for a good definition of “larger”).

3.2 Optimal grid movement and Q-learning

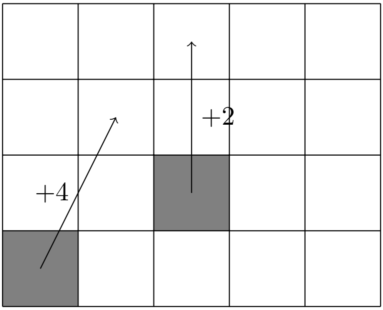

Figure 1: Grid

Consider a robot moving in the grid represented in Figure 1.

- If you are in one of the white cases, you can move in either of the four directions. This costs 0. Actions that would let you out of the grid leave your position unchanged but would cost you -1.
- If you are in a gray case, there is no actions to take and you move directly to the cases indicated by the arrow. The reward that you obtain is the one of the arrow.

You objective is to maximize the total reward with discount factor γ.

1. Compute the value of the RANDOM policy, that move randomly in any of the direction (with probability 1/4).
2. Write a program that uses **value iteration** and computes an optimal policy (consider $\gamma = 0.9$).
3. Write a program that uses **policy iteration** and computes the optimal policy.
4. (*) Write a program that uses the **Q-learning** update to compute the optimal policy. How many iteration does it take?