CSC 860 / MATH 795: Stochastic Optimization & Simulation Methodology
Fall 2022 Assignment 3

Mariia Aleksandrovych, Yaffa Atkins, Atoosa Lotfi, Olga Kliuchnik

Exercises 3.2, 4.2, 4.3, 4.4 and 4.7

EXERCISE 3.2. This is a simple version of a *multi-armed bandit problem*. A slot machine has two arms and the winning probability of the two arms, denoted by p_A and p_B, respectively, are not known. It is assumed that the event of winning on each arm is independent of past history and also independent of the other arm's results. Let θ_n denote the belief that we have that arm A has higher winning probability. Consequently, we choose arm A at step n with probability θ_n (called "exploitation" in machine learning). The goal is to learn the correct value $\theta^* = 1$ if $p_A > p_B$, and $\theta^* = 0$ otherwise (assuming that $p_A \neq p_B$).

A straightforward learning approach is

$$\theta_{n+1} = \begin{cases} \theta_n & \text{if } n\text{-th game was a loss,} \\ \theta_n + \epsilon_n (1 - \theta_n) & \text{if win on arm A,} \\ \theta_n - \epsilon_n \theta_n & \text{if win on arm B.} \end{cases}$$

(a) Write the above recursion as stochastic approximation, specifying the feedback Y_n.

(b) Provide the target vector-field, assuming no bias term is involved.

(c) Argue that the traget vector field is coercive for the learning problem.

(d) *Optional:* Program the procedure to test various schemes for the learning rates $\{\epsilon_n\}$, including the case of constant ϵ. Discuss your results.

Felisa Vázquez-Abad and Bernd Heidergott. 107

a. Stochastic approximation of the problem is $\theta_{n+1} = \theta_n + \varepsilon_n Y_n$ where $Y_n = \begin{cases} 0 \\ 1-\theta_n \\ -\theta_n \end{cases}$. Yn = 0 when nth game was a loss. Yn = 1-theta n if there is a win on arm a. This causes theta n to increase which makes sense since A winning encourages our belief that PA > PB. Yn = -theta n if there is a win on arm b. This causes theta n to decrease which makes sense, since B winning encourages our belief that PA is not greater than PB.

b. By definition, a vector field is the expected value of feedback function on sigma algebra

$$\sigma(\Im_{n-1}) \quad G(\theta_n) = \mathrm{E}\left[Y_n \middle| \Im_{n-1} \right] = \begin{cases} \mathrm{E}(0) \\ \mathrm{E}(1-\theta_n) \\ \mathrm{E}(-\theta_n) \end{cases} \quad G(\theta_n) = \begin{cases} 0 \\ 1-\theta_n \\ -\theta_n \end{cases}$$

Since there is no bias, and constant condition on the feedback, the expected value is simply the value.

c. Theta* is bound between 0 and 1 so it cannot blow up, the vector field is locally Lipschitz continues because cannot vary more than 0 to 1, no matter how many times A wins in a row, or how many times B wins in a row. Theta moves a little bit with each iteration. The possible solutions are Theta* = 1 and Theta* = 0. You probably will never reach a solution but whichever solution you are moving towards is an asymptotically stable point of the ODE because if you see a lot more wins for A, PA is probably bigger and you will keep moving towards 1, and vice versa. Whichever point you are moving towards is a KKT of the NLP because it is stationary since it is once you reach 1 or zero you stop, and they satisfy the equality and inequality constraint of being between 0 and 1.

EXERCISE 4.2. Let $\{X_k\}$ be an iid sequence with mean μ and define

$$S_n = \sum_{k=1}^{n}(X_k - \mu), n \geq 1,$$

Show that S_n is a Martingale with respect to the natural filtration of the process, where $\mathfrak{F}_n = \sigma(X_1, \ldots, X_n)$.

4.2

$\{X_k\}$ iid, $S_n = \sum_{k=1}^{n}(X_k - \mu), n \geq 1$, show S_n is martingle.

S_n is martingle if $E[S_{n+1} | X_1 \cdots X_n] = S_n$,

$$E[S_{n+1} | X_1 \cdots X_n] = E[S_n + (X_{k+1} - \mu) | X_1 \cdots X_n]$$

S_n is constant on $X_1 \ldots X_n$ $\quad = S_n + E[X_{k+1} - \mu \,|\, X_1 \ldots X_n]$

X_{n+1} is independent of $X_1 \cdots X_n$ $\quad = S_n + E\underbrace{[X_{k+1} - \mu]}_{0}$

$= S_n \Rightarrow$ it's Martingle

EXERCISE 4.3. Let $\delta M_i = Y_i - \mathbb{E}[Y_i \,|\, \mathfrak{F}_{i-1}]$ be defined as in the martingale difference noise model. Show that the process $M_n \stackrel{\text{def}}{=} \sum_{i=0}^{n} \epsilon_i \delta M_i$ is a martingale process on $(\Omega, \mathbb{P}, \{\mathfrak{F}_n\})$. Show that $\mathbb{E}[\delta M_n \delta M_m] = 0$. for the basic definition and properties of martingale processes we refer to the Appendix.

Solution:

$M_n \overset{def}{=} \sum_{n=0}^{n} \epsilon_i \delta M_i$ is a martingale process on filtered probability space of $(\Omega, \mathcal{F}, \mathbb{P})$ when

$E[\delta M_n \delta M_m] = 0$ for all $m, n \geq 0$ with $m \neq n$

Let's assume that $0 \leq m < n$ then:

$$
\begin{aligned}
\sum \delta M_n \delta M_m &= \sum (M_n - M_{n-1})(M_m - M_{m-1}) \\
&= \sum M_n M_m - \sum M_n M_{m-1} - \sum M_{n-1} M_m + \\
&\quad + \sum M_{n-1} M_{m-1} \\
&= \sum E(M_n M_m \mid \mathcal{F}_m) - \sum E(M_n M_{m-1} \mid \mathcal{F}_{m-1}) \\
&\quad - \sum E(M_{n-1} M_m \mid \mathcal{F}_m) + \sum E(M_{n-1} M_{m-1} \mid \mathcal{F}_{m-1}) \\
&= \sum M_m E(M_n \mid \mathcal{F}_m) - \sum M_{m-1} E(M_n \mid \mathcal{F}_{m-1}) \\
&\quad - \sum M_m E(M_{n-1} \mid \mathcal{F}_m) + \sum M_{m-1} E(M_{n-1} \mid \mathcal{F}_{m-1}) \\
&= \sum M_m \cdot M_m - \sum M_{m-1} M_{m-1} - \\
&\quad - \sum M_m M_m + \sum M_{m-1} M_{m-1} \\
&= \sum M_m^2 - \sum M_{m-1}^2 - \sum M_m^2 + \sum M_{m-1}^2 \\
&= 0
\end{aligned}
$$

EXERCISE 4.4. Refer to the model in Example 3.4. Show that for a random variable X with finite variance,

$$\nabla J(\theta) = \begin{pmatrix} -\mathbb{E}[Z(X) - \theta_1 - \theta_2 X]) \\ -\mathbb{E}\left[X\, Z(X) - \theta_1 X - \theta_2 X^2\right] \end{pmatrix}. \tag{4.19}$$

For each experimental point x_n we obtain a random observation $\xi_n = Z(x_n)$ with $\mathbb{E}(Z(x_n)) = h(x_n)$. The feedback function is $Y_n = (\xi_n - \theta_n(1) - \theta_n(2)x_n)(1, x_n)^T$. Use (4.19) to show the claim that $\mathbb{E}[Y_n \mid \mathfrak{F}_{n-1}] = -\nabla J(\theta_n)$.

The model form the Exercise 3.4 says that we have cost function as

$$J(\theta) = \frac{1}{2}\mathrm{E}\left[\left(Z(X) - (\theta_1 + \theta_2 X)^2\right)\right]$$

Then we take partial derivatives over θ_1 and θ_2 to get the gradient

Then

$$\frac{\partial}{\partial \theta_1} J(\theta) = \frac{\partial}{\partial \theta_1}\left\{\frac{1}{2}\mathrm{E}\left[\left(Z(X)-(\theta_1+\theta_2 X)^2\right)\right]\right\} = \frac{\partial}{\partial \theta_1}\left\{\frac{1}{2}\int_\Omega \left[\left(Z(X)-(\theta_1+\theta_2 X)^2\right)\right]\mathrm{P}(dX)\right\}$$

$$\frac{\partial}{\partial \theta_2} J(\theta) = \frac{\partial}{\partial \theta_2}\left\{\frac{1}{2}\mathrm{E}\left[\left(Z(X)-(\theta_1+\theta_2 X)^2\right)\right]\right\} = \frac{\partial}{\partial \theta_2}\left\{\frac{1}{2}\int_\Omega \left[\left(Z(X)-(\theta_1+\theta_2 X)^2\right)\right]\mathrm{P}(dX)\right\}$$

The $J(\theta)$ $\frac{\partial}{\partial \theta_2} J(\theta), \frac{\partial}{\partial \theta_1} J(\theta)$ are integrable on $(\Omega, \mathfrak{I}_n)$Then using Leibniz internal rule, we can interchange derivative and integration operations. We have:

$$\frac{1}{2}\int_\Omega \left[\frac{\partial}{\partial \theta_1}\left(Z(X)-(\theta_1+\theta_2 X)^2\right)\right]\mathrm{P}(dX) = \frac{1}{2}\int_\Omega \left[-2\left(Z(X)-(\theta_1+\theta_2 X)\right)\right]\mathrm{P}(dX) = -\mathrm{E}\left[\left(Z(X)-(\theta_1+\theta_2 X)\right)\right]$$

$$\frac{1}{2}\int_\Omega \left[\frac{\partial}{\partial \theta_2}\left(Z(X)-(\theta_1+\theta_2 X)^2\right)\right]\mathrm{P}(dX) = \frac{1}{2}\int_\Omega \left[-2X\left(Z(X)-(\theta_1+\theta_2 X)\right)\right]\mathrm{P}(dX) = -\mathrm{E}\left[\left(Z(X)X-\left(\theta_1 X+\theta_2 X^2\right)\right)\right]$$

Where:

$$\nabla J(\theta) = \begin{pmatrix} -\mathrm{E}\left[\left(Z(X)-\theta_1-\theta_2 X\right)\right] \\ -\mathrm{E}\left[\left(Z(X)X-\theta_1 X-\theta_2 X^2\right)\right] \end{pmatrix}$$

Now let take expectation value of the feedback function $Y_n = \begin{pmatrix} \xi_n - \theta_{n,1} - \theta_{n,2}x_n \\ \xi_n x_n - \theta_{n,1}x_n - \theta_{n,2}x_n^2 \end{pmatrix}$

$$\mathrm{E}\left[Y_n \mid \theta_n\right] = \begin{pmatrix} \mathrm{E}\left[\xi_n - \theta_{n,1} - \theta_{n,2}x_n\right] \\ \mathrm{E}\left[\xi_n x_n - \theta_{n,1}x_n - \theta_{n,2}x_n^2\right] \end{pmatrix} = \begin{pmatrix} \mathrm{E}\left[Z(x_n) - \theta_{n,1} - \theta_{n,2}x_n\right] \\ \mathrm{E}\left[Z(x_n)x_n - \theta_{n,1}x_n - \theta_{n,2}x_n^2\right] \end{pmatrix} = -\nabla_\theta J(\theta_n)$$

EXERCISE 4.7. Consider the following well known Nash equilibrium problem in transportation. The transportation time along an arc i is denoted by t_i and it depends on the vector of normalized traffic flow x, where $x_{(k,l)}$ is the total traffic on the arc (k,l). In Figure 4.1 there are three possible routes from origin to destination, $\{(OAD),(OABD),(OBD)\}$ and the corresponding travel times per arc are shown. The fraction of traffic on route r is denoted by θ_i, so that $\sum_{i=1}^{3}\theta_i = 1.0$. Therefore the fraction of traffic on the arc (O,A) is given by $x_{(O,A)} = \theta_1 + \theta_2$ and similarly for other arcs. The total time on route i is the sum of the travel times along the arcs that form the route and is denoted by $T_i, i = 1,2,3$.

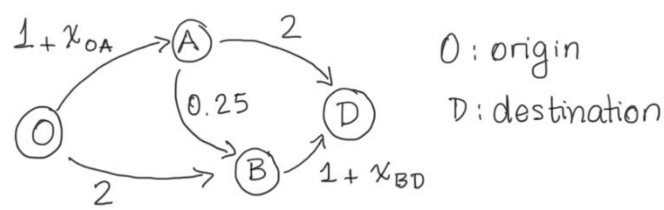

Figure 4.1: Traffic flow equilibrium

A Nash equilibrium will be an allocation such that if a driver on route i decides to change his/her path, then he/she will experience a larger delay that staying at equilibrium. The vector of travel times per route is given by:

$$T(\theta) = \begin{pmatrix} 3 + \theta_1 + \theta_2 \\ 2.25 + \theta_1 + 2\,\theta_2 + \theta_3 \\ 3 + \theta_2 + \theta_3 \end{pmatrix}$$

(a) Show that there is a unique Nash equilibrium, by showing that there is a unique value θ^* such that $T_i(\theta^*) =$ constant, is independent of i.

(b) In order to attain equilibrium, the flow on path i should aim to "equalise" the delay. Suppose now that we do not know the various constants in the delay function, but can only just estimate the delays by running a simulation, which yields an unbiased estimator $\widehat{T(\theta)}$.

$$\theta_{n+1,i} = \theta_{n,i} - \epsilon_n \left(\widehat{T_i(\theta_n)} - \frac{1}{3} \sum_k \widehat{T_k(\theta_n)} \right). \tag{4.20}$$

Show that the algorithm is mass preserving, that is, $\sum_i \theta_{0,i} = 1$ then $\sum_i \theta_{n,i} = 1$.

(c) Characterise the behaviour of the stochastic approximation (4.20) and specify your assumptions. In particular, show that the target vector field is coercive for the equilibrium problem.

By the way, this example is classic to show that the Nash equilibrium does not minimise overall travel time. For the specific model, $\theta^* = (0.25, 0.50, 0.25)^T$, and $T_i(\theta^*) = 3.75$ for all i. However, for $\theta = (0.50, 0.0, 0.50)$, then $T_1(\theta) = T_3(\theta) = 3.5$.

$\boxed{4.7}$ t_i = transportation time along arc i

fraction of traffic on route = θ_i where $\sum_{i=1}^{3} \theta_i = 1$

a) vector of travel times is given in the problem statement

$$T(\theta) = \begin{pmatrix} 3 + \theta_1 + \theta_2 \\ 2.25 + \theta_1 + 2\theta_2 + \theta_3 \\ 3 + \theta_2 + \theta_3 \end{pmatrix}$$

regardless of i, we want constant θ^*

show that $T(\theta_1) = T(\theta_2) = T(\theta_3)$:

- $T(\theta_1) = T(\theta_3)$

$$= 3 + \theta_1 + \theta_2 = 3 + \theta_2 + \theta_3$$

$$= \quad \theta_1 = \theta_3$$

- $T(\theta_1) = T(\theta_3)$

$$= 3 + \theta_1 + \theta_2 = 2.25 + \theta_1 + 2\theta_2 + \theta_3$$

$$= \quad .75 = \theta_2 + \theta_3$$

$$\theta_1 + \theta_2 + \theta_3 = 1$$

$$\theta_2 + \theta_3 = .75$$

So $\theta_1 = .25$

$\theta_1 = \theta_3 = .25$

So $\theta_2 = .5$

b) we want to equalise the delay, we only have an estimator.

$$\theta_{n+1,i} = \theta_{n,i} - \epsilon_n \left(\widehat{T_i(\theta_n)} - \frac{1}{3} \sum_K \widehat{T_K(\theta_n)} \right)$$

estimate of current route time

average of estimate of total route time.

Show that this algorithm is mass preserving, meaning at any itteration, the routes still sum to 1:

Although individual $T_i(\theta)$ might change, $\Sigma_i \theta_i$ is always 1. so, sum of average $\theta_i = 1$

$1-1=0$ so the average change is always 0.

So $\sum_{i=1}^{3} \left(\widehat{T_i(\theta_n)} - \frac{1}{3} \sum_K \widehat{T_K(\theta_n)} \right) = 0$

and $\sum_{i=1}^{3} \widehat{T_i(\theta_n)} - 3 * \frac{1}{3} \sum_K \widehat{T_K(\theta_n)} = 0$

$\epsilon_n * 0 = 0$ so θ_n isn't changing

c) Behaviour of the above stochastic optimization:

If $T_i(\theta) >$ average $T_i(\theta)$

then $\theta_{n+1} < \theta_n$: moving towards less traffic.

and vise versa, equalizing the delays!

Our assumptions: $\epsilon \to 0$ and $\Sigma \epsilon_i = \infty$

This is because as we get closer to equalized delay we need more accuracy, but we many never actually reach it. we'll just keep getting closer with infinate iterations.

we also assume the solution is a stationary point because once we reach zero, θ stays the same.

Coercive: $\sum_{i=1}^{3} \theta_i = 1$ so θ is bound,

for each θ: $0 < \theta < 1$

Since θ is bound, the target vector field doesn't blow up in finite time, rather it keeps getting closer to the right answer.

The target vector field is locally Lipschitz since θ is bound and cannot blow up, it moves in very small steps which tend to zero

we reach an asymptotically point when all delays are equalized. this is the solution.

$$\widehat{T_i(\theta_i)} - \frac{1}{3}\sum_{K} \widehat{T_K(\theta_n)} = 0$$

so $\theta_{n+1} = \theta_n$ = stationary

solution vector θ is within the bounds of $0 < \theta < 1$

so its a KKT point.