AMS 514 Fall 2024 Second Project

A. P. Mullhaupt

November 19, 2024

1. The Second Project

The Second Project consists of the two Exercises in the following description.

2. Thompson Sampling

The multi-armed bandit problem is not one problem, it is a family of problems with many different variations. But for the purposes of this assignment, the multi-armed bandit problem is a sequential decision problem where the player has at each time t the possibility of choosing any of the actions $\{a_1, ..., a_m\}$; having chosen say, action a_k, then the 'bandit' chooses a payout $y(t, a_k)$. We consider the case of *partial information*, where the player is not informed what payout would have occurred for choices that were not made at time t.

Thompson sampling is a simple but effective strategy which can be applied to the multi-armed bandit problem. It consists of maintaining a distribution approximating that of the payout of each arm, and at each round, sampling from each distribution, and choosing the arm with the highest sample.

Example 1. *A very simple example is Poisson trials, that is a sequence of i.i.d. Bernoulli random variables, where the actions are $\{H, T\}$ (for 'heads' and 'tails' respectively) and at each time t the probability of 'heads' is q_t; and this sequence of probabilities is unknown to the player. The payout when the player correctly matches the outcome is 1 and 0 otherwise. In other words, at each time, the total payout the player has received is the number of correct action choices so far. The object of the player is to maximize this number in some sense. The strategy of simple Thompson sampling is for the player to keep track of the (sample) mean payout for each action. In other words, for action k that was chosen at time t*

$$N_k(t) = N_k(t-1) + 1 \tag{1}$$

and

$$\begin{aligned} Q_k(t) &= \frac{N_k(t-1)\,Q_k(t-1) + y(t, a_k)}{N_k(t)} && (2) \\ &= \frac{N_k(t-1)}{N_k(t)} Q_k(t-1) + \frac{1}{N_k(t)}\left(y(t, a_k) - Q_k(t-1)\right) && (3) \end{aligned}$$

and for the actions that were not chose, $Q_j(t) = Q_j(t-1)$. Note that since action k was chosen at time t that $N_k(t)$, the number of times action k has been chosen up until time t, we are not dividing by zero.

At time $t+1$ the Thompson sampler draws a sample from each of $Bern(Q_k)$ distributions approximating the payout of each arm, and chooses the arm which had the highest sample. It is necessary to 'initialize' the p.m.f. with a 'prior' in order to make a choice at time $t = 1$. It is important that this prior have a positive probability of choosing every one of the actions, otherwise that action will never be chosen.

Exercise 2. *Write code to simulate this Poisson trial example, for the following sequences of success probabilities $P(H|t)$:*

$$\begin{aligned} P(H|t) &= \frac{3}{5} && (4) \\ P(H|t) &= t \bmod 2 && (5) \\ P(H|t) &= \left\lfloor \frac{\log t}{\log 2} \right\rfloor \bmod 2 && (6) \\ &= \{0,1,1,0,0,0,0,1,1,1,1,\cdots\}. && (7) \end{aligned}$$

Show equity curves and learning curves. Explain what you did to resolve the situation where the samples from the H action and the T action were equal for a particular round.

3. Thompson Controlled Mixture Sampling for Global Noisy Nonconvex Maximization

Here we consider the possibility of using Thompson sampling to find a global maximum of a nonconvex function ϕ for which we only have noisy estimates $\phi(x) + \eta$. The simple algorithm for this project is to define a mixture distribution on the seach space, and then use Thompson sampling, where at each round, the Thompson sampler chooses which mixture component to sample from. In the case of Gaussian noise η this means the Thompson sampler chooses an arm according to the probability that arm would have the maximum sample from the estimated Gaussians for each arm.

With x as the sample from that mixture component, then we obtain payout $\phi(x) + \eta$. We hope that the Thompson sampler proposal distribution has entropy which decreases (indicating learning).

When the proposal distribution has low enough entropy, then we can either terminate, or we can find another mixture which has higher entropy, and resume Thompson sampling.

In this project, the search space is simply the unit interval $[0,1]$ and the partition is simply a partition into m parts

$$x_0 = 0 < x_1 < \cdots < x_m = 1 \tag{8}$$

and the mixture distribution has k^{th} component $U(x_{k-1}, x_k)$, and proposal probability $\tilde{P}_k(t)$. Sampling x from this mixture corresponds to choosing the component k with probability $\tilde{P}_k(t)$ and then choosing $x \sim U(x_{k-1}, x_k)$.

This has the interpretation that action k is deciding to look in (x_{k-1}, x_k) for the maximum, and then the payout $\phi(x) + \eta$ will be high or low depending on the noise η and the values of $\phi(x)$ in that interval.

The (Shannon) entropy of the proposal distribution is

$$H(t) = \sum_{k=1}^{m} \tilde{P}_k(t) \log \frac{1}{\tilde{P}_k(t)} \tag{9}$$

and if each interval is equally likely to be proposed, then then $\tilde{P}_k(t) = \frac{1}{m}$ and the entropy is

$$\sum_{k=1}^{m} \frac{1}{m} \log m = \log m \tag{10}$$

so a reasonable condition to stop learning the partition might be that $H(t) << \log m$, but in this application, there might be two intervals adjacent to the maximum of $\phi(x)$, so the

Thompson sampler might learn a proposal distribution with two nearly equal components, which will have entropy approximated by $\log 2$. So it is not always possible to wait until $H(t)$ is absolutely small.

In this project we will stop learning when $H(t) \leq \frac{1}{2} H(0)$, corresponding roughly to reducing the uncertainty of m equal choices to $\sqrt{m}$ equal choices.

Finally, we address the question of repartitioning. A simple method of repartitioning the interval $(0,1)$ is to choose the new partition points x'_k to be the points at which the linear interpolation of the proposal cdf are equal to k/m.

Example 3. *Suppose we are maximizing the function $\phi(x) = x(1-x)$ with no noise (that is $\eta = 0$). If we have $m = 3$ and start with the partition $x = \left\{0, \frac{1}{3}, \frac{2}{3}, 1\right\}$ and run the Thompson sampler for a long time, we will find the proposal distribution has p.m.f. very close to $(0,1,0)$, indicating (correctly) that the maximum is in the second interval. If we then repartition by choosing the points x'_k to be where the linearly interpolated cdf equals k/m then we will obtain*

$$x' = \left\{0, \frac{4}{9}, \frac{5}{9}, 1\right\} \tag{11}$$

and the new proposal distribution has an action which still contains the maximum but in a smaller part of the search space.

Exercise 4. *Write code for the objective functions $\phi(x) = x(1-x)$ and $\phi(x) = 2 + 2x(1-2x) + \frac{1}{50}\sin(52\pi x)$ and $\eta(x)$ given by Gaussian noise with mean 0 and variance $(1-x)^2$. Start with the unit interval partitioned into $m = 16$ equal parts, and repartition when $H(t) = \frac{1}{2} H(0)$, and repeat the repartitioning three times. Plot the objective functions against the histogram of the proposal distribution at the end of each of the three partitionings. Show how the entropy of the proposal distribution declined with t.*

4. Optimization of an Optimization-resistant function

The third part of the exercise is to find the minimum of a continous, but nonsmooth, hightly nonconvex function. This is actually impossible, so we will settle for something that is not impossible: to find a set which is in one respect small, but which has a high probability of containing the minimum. You can think of this as 'localizing' the minimum of this function. We say 'the' minimum because the function in question will have only one global minimum, but infinitely many local minima.

The approach by Thompson sampling is essentially the same as for the previous example, which was a smooth function with added noise. The only difference is that this objective is a function that has no noise, but a lot of wrinkles than seem like noise in the beginning. The function is a realization of the symmetric Brownian bridge from (0,0) to (1,0). Almost surely, such a realization is continuous, has a unique minimum, and is nowhere differentiable. (The module on 'Online Sampling the Brownian Bridge' explains this in more detail, and with code.)

Exercise 5. *Use Thompson sampling to find a partition of the interval (0,1) and p.m.f. for that partition that gives the probability that the realization of the Brownian bridge being sampled is in each interval, where the p.m.f. is concentrated on intervals that cover a small fraction of the overall interval (0,1).*

The only significant modification of the code from the last section is that the function called at each round needs to keep track of all the values that have already been returned from previous calls. As given in the 'Online Sampling the Brownian Bridge' module, one simple way to write that function is:

```
%
%   incremental_brownian_bridge: samples a Brownian Bridge in increments
%   and retains the previous samples
%

function [t, B] = incremental_brownian_bridge(x, t, B)

    if nargin == 0
        % we initialize the bridge with B(0) = B(1) = 0;
        t = [0;1];
        B = [0;0];
    else

        rind = find(t < x, 1, 'last'); r = t(rind);
        sind = find(t > x, 1, 'first'); s = t(sind);

        r = t(rind); s = t(sind);

        % Brownian bridge sample:

        mu = ((s - x)*B(rind) + (x - r)*B(sind)) / (s - r);
        sigma = sqrt((s - x)*(x - r)/(s - r));
        y = mu + sigma * randn;

        B = [B; y];
        t = [t; x];

    end

end
```