Lava: Parameter Optimization

Kenny Peluso

August 2018

1 Problem Statement

Lava facilitates a game between *randers* - players who submit random numbers - and *preders* - players who predict those random numbers - funded by *consumers* - external parties who request random numbers. As this game is played, consumers purchase samples from a discrete distribution that is likely to be uniform. Hence, Lava is a means of decentralized random number generation.

From here onward, it is assumed that the reader has already read the `README.md` in the GitHub repo. See the end of this paper for a link to that repo.

Let a *round* be the time between one random number request (made by a consumer), exclusive, and the next random number request, inclusive. Let $\mathbb{E}[P_i], \mathbb{E}[R_j]$ be the expected value of preder i and rander j, respectively, for a single round of play. We seek to find parameters such that $0 < \mathbb{E}[P_i] = \mathbb{E}[R_j]$. In other words, we intend for Lava to be a fair game that's profitable for all players involved. We'll achieve this by tuning *parameters* while making assumptions about *hyperparameters*:

Parameters:

- B - Cost per random number paid by a consumer
- C - Deposit per random number paid by randers
- D - Stake per prediction paid by preders

Hyperparameters:

- A - Number of random numbers in the cyclical array
- $\hat{N}_P$ - Anticipated number of preders
- $\hat{N}_R$ - Anticipated number of randers
- X - Size of the uniform distribution's support

Furthermore, we will make the following assumptions:

- The pot is empty.
- *Long-run behavior* is dominant (randers and preders actually submit random numbers and predictions from the discrete uniform distribution).
- $\hat{N}_P$ preders are always participating.
- $\hat{N}_R$ randers are always participating.

- Randers participate once per round in a random order.
- Preders participate once per round.

2 Setup

First, define $Q(N) = \frac{(N-1)!}{N!}$. $Q(N)$ is the probability that one thing will be found in any particular spot if we were to randomly choose an ordering of N things. For example, if N people were to randomly line up for a concert, the probability that any single one of them is fourth in line is $Q(N)$.

Let's start with the preders, in words:

$\mathbb{E}[P_i] =$ [chance of guessing correctly] $\times \mathbb{E}$[winnings given preder submitted accurate prediction first] + [chance of guessing incorrectly] $\times(-D)$

$\mathbb{E}[P_i] = \frac{1}{X}\mathbb{E}$[winnings given preder submitted accurate prediction first] $- (1 - \frac{1}{X})D$

$$\mathbb{E}[P_i] = \frac{1}{X}\left[\left(1 - Q\left(\left\lfloor \frac{\hat{N}_P}{X} \right\rfloor\right)\right)\left(\frac{B}{\hat{N}_P/X}\right) + Q\left(\left\lfloor \frac{\hat{N}_P}{X} \right\rfloor\right)\left(AC + \frac{B}{\hat{N}_P/X}\right)\right] - (1 - \frac{1}{X})D$$

Why does this make sense?

$\frac{\hat{N}_P}{X}$ is the number of accurate preders ($\hat{N}_P$ independent trials from a discrete uniform distribution with support $1, \ldots, X$). $Q\left(\left\lfloor \frac{\hat{N}_P}{X} \right\rfloor\right)$ is defined above. This is equal to the probability of preder i being first to submit a prediction among all preders.

A winning preder always wins $\frac{B}{\hat{N}_P/X}$. If they're the first accurate preder, then they also win AC, or all of the randers' deposits (among those still in the cyclical array). A winning preder also makes make their staked amount D. They lose this stake if they're incorrect.

Let's now turn to the randers, in words:

$\mathbb{E}[R_j] =$ [chance of no preder guessing correctly] $\times \mathbb{E}$[winnings given rander submitted an unpredicted random number] + [chance of at least 1 preder guessing correctly] $\times \mathbb{E}$[losses given rander submitted a predicted random number]

$$\mathbb{E}[R_j] = (1 - \tfrac{1}{X})^{\hat{N}_P}\left[Q\big(\hat{N}_R\big)\big(\tfrac{B}{2}\big) + \left(1 - Q\big(\hat{N}_R\big)\right)\left(\sum_{k=3}^{A+1} \tfrac{B}{k^2}\right)\right] + \big(1 - (1 - \tfrac{1}{X})^{\hat{N}_P}\big)\left(\tfrac{A}{\hat{N}_R}(-C)\right)$$

Why does this make sense?

$(1 - \frac{1}{X})^{\hat{N}_P}$ is the binomial distribution for all failures in $\hat{N}_P$ trials. $Q\big(\hat{N}_R\big)$ is the probability that any rander is first (was the last person to submit a random number before a request was made). Again, this is equal to the probability of a rander being in any particular place. If a rander is first, they receive $\frac{B}{2}$ in prize money. Otherwise, they earn $\frac{B}{k^2}$ where k is their "place" ($k = 1$ if rander was first, $k = 2$ rander was second, etc.).

If rander j is part of the cyclical array and the first rander's random number is successfully predicted, then rander j loses their deposit C. If rander j is *not* part of the cyclical array, then their deposit is returned and

they are safe from a preder's success. Since random number submissions occur randomly, the probability that any rander will land in the length-A cyclical array is $\frac{A}{\hat{N}_R}$.

Furthermore, it can be shown that:

$$\sum_{k=3}^{A+1} \frac{1}{k^2} = \frac{\pi^2}{6} - \psi^{(1)}(A) - \frac{3}{4}$$

where $\psi^{(i)}(n)$ is the ith derivative of the digamma function evaluated at n. Note that the above series is a finite 2-series (or a finite "overharmonic series" with $p = 2$).

Let's simplify both formulae further:

$$\mathbb{E}[P_i] = B\frac{1}{\hat{N}_P} + C\left(\frac{AQ\left(\left\lfloor \frac{\hat{N}_P}{X} \right\rfloor\right)}{X}\right) + D\left(\frac{1}{X} - 1\right)$$

$$\mathbb{E}[R_j] = B\left[Q\big(\hat{N}_R\big)(1 - \frac{1}{X})^{\hat{N}_P}\left(\frac{\pi^2}{6} - \psi^{(1)}(A) - \frac{3}{4}\right)\right] + C\left[\frac{A}{\hat{N}_R}\left(\left(1 - \frac{1}{X}\right)^{\hat{N}_P} - 1\right)\right]$$

Given our hyperparameters, B, C, D exist as variables in an underdetermined system of linear equations. We can make this system determined by imposing an *Illusory Fairness*, whereby $C = D$. This is illusory because equating a preder's stake to a rander's deposit (equating the cost of entry of all players) does not make a game fair; Equating the *expected values* of all players' net winnings makes a game fair. We aim to attain true fairness anyway, so any success we find in this regard will not be illusory, but real.

By imposing Illusory Fairness, we have:

$$\begin{bmatrix} \alpha & (\beta + \gamma) \\ \delta & \epsilon \end{bmatrix} \begin{bmatrix} B \\ C \end{bmatrix} \frac{1}{\omega} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

or, more succinctly:

$$Mx = b$$

where:

$$\omega = \mathbb{E}[P_i] = \mathbb{E}[R_j] > 0$$

$$\alpha = \frac{1}{\hat{N}_P}$$

$$\beta = \frac{AQ\left(\left\lfloor \frac{\hat{N}_P}{X} \right\rfloor\right)}{X}$$

$$\gamma = \left(\frac{1}{X} - 1\right)$$

$$\delta = Q\big(\hat{N}_R\big)(1 - \frac{1}{X})^{\hat{N}_P}\left(\frac{\pi^2}{6} - \psi^{(1)}(A) - \frac{3}{4}\right)$$

$$\epsilon = \frac{A}{\hat{N}_R}\left(\left(1 - \frac{1}{X}\right)^{\hat{N}_P} - 1\right)$$

3 Solve

Note that M is a function of the hyperparameters. We easily solve for B, C, D up to a constant scalar factor (up to phase) by computing $x = M^{-1}b$ for various hyperparameter values within reason. This work was done already and can be found in the linked repo under `docs > params.py`.

There were only a few permutations of hyperparameter values that returned positive values for ω, B, C, D. One such permutation was selected to be used in Lava, namely:

- $A = 100$
- $\hat{N}_P = 100$
- $\hat{N}_R = 100$
- $X = 65536$
- $B = 113.20453437$
- $C = D = 0.13204736$

Again, we know B, C, D up to a constant scalar factor. We can normalize B, C, D with respect to C and D and then round to the nearest integer, thus yielding the following parameter values:

- $B = 857$
- $C = D = 1$

Recall that the constant we chose for normalization, ω, is the expected return for each player.

Note that, given the small values of $\hat{N}_P$ and $\hat{N}_R$, copies of Lava can be launched and used simultaneously whenever any deployment of Lava becomes too crowded to yield substantial positive returns for players.

4 Conclusion and Future Work

The biggest risk undermining the promises of Lava with this set of parameters is the strength of our first and last assumptions. The conjunction of these assumptions is blind to the following scenario: If the pot were to grow past some threshold, then it may be advantageous for preders to stake their bets on *all* possible values in the support (the integers 1 through X), making their payout inevitable. However, if the timing of the next round is inaccurately predicted or if too many preders adopt the same strategy (lowering the payout to each correct preder), then the opportunistic preder's windfall may be stifled. Future work will address this risk anyway by actually including historical winnings and pot growth into the expected winnings models formulated above.

5 Supplementary Resources

This paper and its parent repo can be accessed on GitHub at:

`https://github.com/kpeluso/lava`