Hunger Games Competition: Methodology behind the algorithm

David Shin, Remy Lee, and Alex Tairbekov

August 18, 2013

Maximizing the expectation for each round

In order to maximize expectation, we need to express it in some manageable form. Let β be a binary variable representing our player's decision to hunt or slack (such that $\beta = 1$ represents a decision to hunt and $\beta = 0$ represents a decision to slack.) Let H be the event that the player's partner hunts. Thus, the expectation of each round can be expressed as:

$$E(T_k) = \sum_{i=1}^{p-1} f(\beta_i)\mathbb{P}(H_i) + \sum_{i=1}^{p-1} g(\beta_i)[1 - \mathbb{P}(H_i)] + 2(p-1)\mathbb{P}(\lambda n \geq m) \quad (1)$$

where f represents the value of the decision to hunt or slack given your partner hunts and g represents the value of the same decision given your partner slacks. Numerically, $f(0) = 1$, $f(1) = 0$, $g(0) = -2$, and $g(1) = -3$. In addition, $\mathbb{P}(\lambda n \geq m)$ is the probability of obtaining the bonus, which is awarded if the total number of hunts is greater than or equal to m. We will express the total number of hunts N as a product of n, the number of hunts your player participates in, and λ, a scaling parameter.

We can rearrange $E(T_k)$ as:

$$E(T_k) = \sum_{i=1}^{p-1} g(\beta_i) + \sum_{i=1}^{p-1} [f(\beta_i) - g(\beta_i)]\mathbb{P}(H_i) + 2(p-1)\mathbb{P}(\lambda n \geq m) \quad (2)$$

Since $\sum_{i=1}^{p-1} g(\beta_i) = -3n - 2(p-1-n) = 2 - 2p - n$, and $f(\beta_i) - g(\beta_i) = 3$ for all β_i (for both $\beta_i = 0$ and $\beta_i = 1$), we can simplify the expectation even further:

$$E(T_k) = 2 - 2p - n + 3\sum_{i=1}^{p-1} \mathbb{P}(H_i) + 2(p-1)\mathbb{P}(\lambda n \geq m) \tag{3}$$

Now, let's take a step back from the expectation and focus on $\mathbb{P}(\lambda n \geq m)$. We know that m is given and we choose n, so λ is the only random variable. Thus, $\mathbb{P}(\lambda n \geq m) = \mathbb{P}(\lambda \geq \frac{m}{n})$. Assume that the reputations don't change significantly this round ($\Delta R \approx 0$.) Then $n_i \approx kR_i$, so $N = \sum n_i \approx k \sum R_i$. Since $N = \lambda n$, $N \approx \lambda kr$, so $\lambda kr \approx k \sum R_i$. Hence,

$$\lambda \approx \frac{\sum R_i}{r} = \frac{1}{p}\sum \frac{pR_i}{r} \tag{4}$$

In this form, we can approximate λ as a random variable with a Normal distribution (using the Central Limit Theorem). Let μ_R be the mean of R_i and σ_R^2 be the variance. Then,

$$\mu_\lambda = \frac{\mu_R p}{r} \text{ and } \sigma_\lambda^2 = \frac{\sigma_R^2 p}{r^2} \tag{5}$$

Thus, we can approximately describe λ as:

$$\lambda \approx N\left(\frac{\mu_R p}{r}, \frac{\sigma_R^2 p}{r^2}\right) \tag{6}$$

We can now approximate $\mathbb{P}(\lambda \geq \frac{m}{n})$ with the cumulative density function (CDF) of the Normal distribution. To do this, we use the error function which is in R and MATLAB and can easily be implemented in Python to a high precision. The probability is:

$$\mathbb{P}\left(\lambda \geq \frac{m}{n}\right) \approx \frac{1}{2}\left(1 + \text{erf}\left(\frac{r}{\sigma_R\sqrt{2p}}\left(\frac{\mu_R p}{r} - \frac{m}{n}\right)\right)\right) \tag{7}$$

We still need a function for $\mathbb{P}(H_i)$, so we will make some assumptions. For a given player, the total number of hints is:

$$N = \lambda n_i \approx n\frac{\sum R}{R_i} \tag{8}$$

Thus, the number of hunts with that specific player should be approximately:

$$\frac{N}{p} \approx \frac{n}{p}\frac{\sum R}{R_i} \tag{9}$$

However, n is decided by the player, so we will ignore n and let the indexing value, ω be:

$$\omega = \frac{\sum R}{pR_i} \tag{10}$$

We can think of ω as the average of the reputations inversely weighted by R_i. Note that the range of ω is $(0, \infty)$. We need $\mathbb{P}(H_i) \rightarrow 0$ as $r \rightarrow 0$ and $\mathbb{P}(H_i) \rightarrow \max[\mathbb{P}(H_i)]$ as $r \rightarrow 1$. One intuitive approach is to use $\frac{1-r}{r}$, which also has domain $(0, \infty)$. Thus, if $r \rightarrow 0$, $\frac{1-r}{r} \rightarrow \infty$, and $\mathbb{P}(\omega \geq \frac{1-r}{r}) \rightarrow 0$. Similarly, if $r \rightarrow 1$, $\frac{1-r}{r} \rightarrow 0$, and $\mathbb{P}(\omega \geq \frac{1-r}{r}) \rightarrow \alpha$, where α is the maximum probability.

Thus,

$$\mathbb{P}(H_i) \approx \mathbb{P}\left(\frac{\sum R}{pR_i} \geq \frac{1-r}{r}\right) = \mathbb{P}\left(\frac{\sum R}{r} \geq \frac{pR_i(1-r)}{r^2}\right) \tag{11}$$

Since $\lambda \approx \frac{\sum R}{r}$, we can also approximate this probability:

$$\mathbb{P}(H_i) \approx \frac{\alpha}{2}\left(1 + \operatorname{erf}\left(\frac{r}{\sigma_R\sqrt{2p}}\left(\frac{\mu_R p}{r} - \frac{pR_i(1-r)}{r^2}\right)\right)\right) \tag{12}$$

Now we have the expectation as a function of n which means we can find the value of n that maximizes the expectation. Note that this relies on the assumption that each player will attempt to optimize their reputation and that the higher our reputation is, the higher the probability that they will hunt with us. We should also try to find r that maximizes the expectation so we can adjust the previous n by adding or subtracting ϵ to get that r. To do this, we set up the expectation equation and maximize r, from which we find ϵ by:

$$\epsilon = (k+1)(p-1)\left[r_{\max} - \frac{k}{k+1}r_0\right] - n_{\max} \tag{13}$$

where k is the current round number, $r_{\max}$ is the maximizing r for the next round, r_0 is the value of r for the current round, and $n_{\max}$ is the maximizing n for the current round.

In practice

While this algorithm is running, it should collect approximate values for α and λ. The mean and variance of λ should be merged with the approximated $\lambda \approx \frac{\sum R}{r}$. The value of α is solved for using the empirical data in `food_earnings` and the approximated values of the hunting probabilities.

To apply this data, we have four stages: in the first we are naively approximating the value that maximizes expectation, in the second we include the parameter ϵ, in the third we no longer hunt with reputations below a certain threshold, and in the final stage, we calculate the optimal r rather than n. The ranges for these stages were estimated and calculated empirically.