A Minimal Working Example of Empirical Gradient Ascent

Mathias Winther Madsen
mathias@micropsi-industries.com

May 17, 2016

This note describes a very simple discrete-state game that can be played by artificial agents. The game has an explicit solution, which we can compare our intuitions and approximation methods against.

1 States, Actions, and Random Policies

The game discussed in this note is played on a board containing three fields, $\Omega = \{0, 1, 2\}$, by an agent capable of two different actions, $A = \{+1, -1\}$.

Movement is equivalent to modulo 2 addition; that is, the player can move around on a triangle in a clockwise or counterclockwise fashion. We will assume that the game always begins with the player being randomly placed on either $\omega = 0$ or $\omega = 1$. The game halts when the agent reaches the field $\omega = 2$ (which is thus a terminal state).

Since the player is not permitted to stand still across rounds, all histories of this game has one of the two forms

$$0101\ldots012 \qquad \text{or} \qquad 1010\ldots102.$$

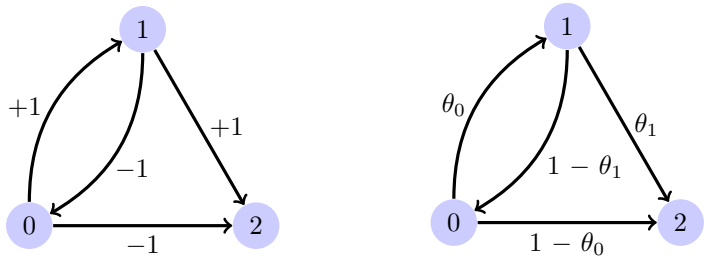

Figure 1: The possible moves in the game defined in the text, first as actions (left), and then as transition probabilities (right).

θ_0	θ_1	Description	p_0	p_1	$E_\theta[R]$
1	1	Always clockwise	1	1	1
0	0	Always counterclockwise	0	0	0
1/2	1/2	Random walk	1/3	2/3	1/2
0	1	Terminate immediately	0	1	1/2
1	0	Never terminate	0/0	0/0	N/A

Table 1: Some examples of expected rewards under different action policies.

The rules of movement are illustrated as transition diagrams in Figure 1.

Conditional on the choices of the player, the rules of the game define a deterministic trajectory $(\omega_1, \omega_2, \ldots, \omega_N)$. In other words, the world defined by this game contains no other randomness than what the agent brings to the table.

An agent's strategy for playing this game can be described in terms of three transition probabilities $\theta = (\theta_0, \theta_1, \theta_2)$, which determine how likely the agent is to walk clockwise when standing in state $\omega = 0, 1, 2$. However, since state $\omega = 2$ is a terminal state, so no decision made in this situation will ever affect the outcome of the game. We will therefore consider the agent's policy as parametrized by the pair $(\theta_0, \theta_1) \in [0, 1]^2$.

2 The Exact Value Gradient

The goal of the game is to walk from field $\omega = 1$ to field $\omega = 2$. When this happens, the player is paid a reward of 1 cookie, and the game ends (since $\omega = 2$ is a terminal state). Trajectories that terminate the game by taking the agent from field $\omega = 0$ to field $\omega = 2$ yield a payoff of 0 cookies.

Let's use the names p_0 and p_1 for the probabilities that an agent will eventually earn a cookie when, at some point in a trajectory, its finds itself in state 0 and 1, respectively. Given a fixed policy $\theta = (\theta_0, \theta_1)$, these two probabilities must satisfy

$$\begin{aligned} p_0 &= \theta_0 p_1 \\ p_1 &= \theta_1 + (1 - \theta_0) p_0 \end{aligned}$$

This implies that

$$\begin{aligned} p_0 &= \frac{\theta_0 \theta_1}{1 - \theta_0(1 - \theta_1)} \\ p_1 &= \frac{\theta_1}{1 - \theta_0(1 - \theta_1)} \end{aligned}$$

Since the agent is initially placed randomly $\omega = 0$ or $\omega = 1$ with equal probability, this implies that expected reward given the policy θ is

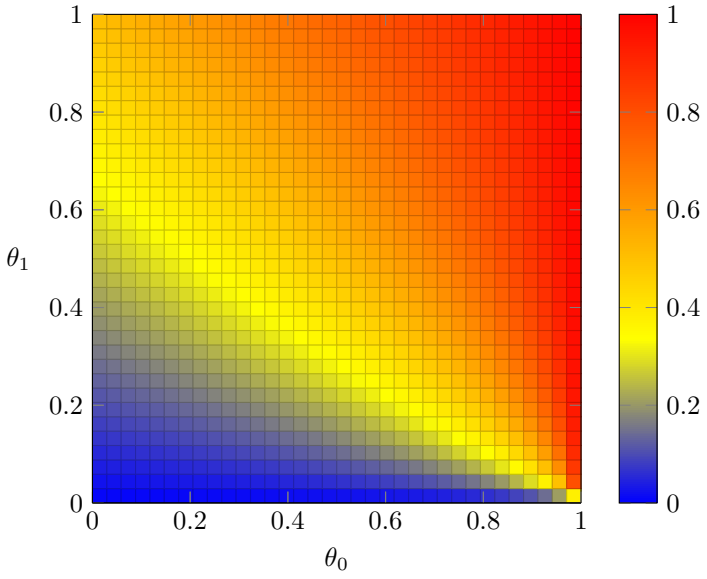

Figure 2: The expected reward as a function of the two policy parameters.

$$E_\theta[R] = \frac{p_0}{2} + \frac{p_1}{2} = \frac{1}{2}\frac{(1+\theta_0)\theta_1}{1-\theta_0(1-\theta_1)}.$$

The example values in Table 1 confirm the sanity of these numbers, and Figure 2 provides a heatmap of the expected reward on the policy parameter space.

Differentiating the expected derivative with respect to the two policy parameters, we find

$$\begin{aligned}\frac{\partial}{\partial\theta_0}E_\theta[R] &= \frac{1}{2}\frac{(2-\theta_1)\theta_1}{(1-\theta_0(1-\theta_1))^2}\\ \frac{\partial}{\partial\theta_1}E_\theta[R] &= \frac{1}{2}\frac{1-\theta_0^2}{(1-\theta_0(1-\theta_1))^2}\end{aligned}$$

The gradient of the expected reward function with respect to θ is therefore

$$\nabla E_\theta[R] = \frac{1}{2}\frac{1}{(1-\theta_0(1-\theta_1))^2}\begin{pmatrix}(2-\theta_1)\theta_1\\ 1-\theta_0^2\end{pmatrix}.$$

Figure 3 provides a plot of this gradient as a quiver plot, showing agent's incentive to change policy as a function of the current choice of policy.

Note that any strategy with $\theta_0 = 1$ is a "solution" to the game. Such a strategy is blocked from exiting the game through the wrong door and therefore guaranteed to eventually earn a point. This may happen only after a long period of bouncing back and forth between states $\omega = 0$ and $\omega = 1$, but since we do not penalize long trajectories, a strategy with $\theta_0 = 1$ and $\theta_1 \approx 0$ does not, by our rules, have any preference over one with $\theta_0 = 1$ and $\theta_1 \approx 1$.

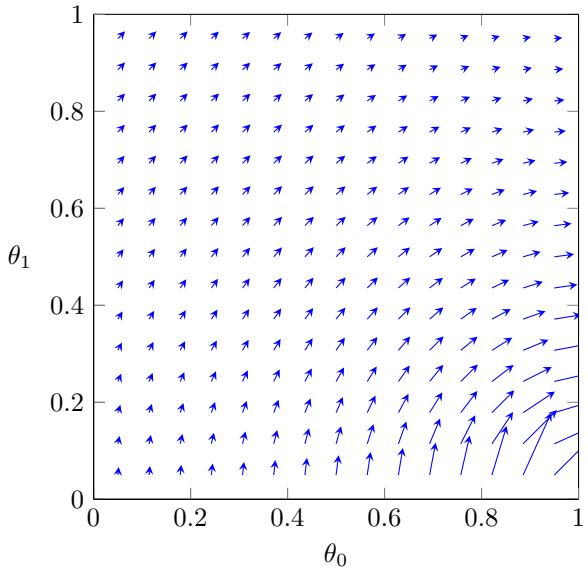

Figure 3: The gradient of the expected reward function.

3 Approximate Value Gradients

We could like to estimate the gradient of the expected reward function at θ using the approximation

$$\frac{\partial}{\partial \theta_k} E_\theta [R] \approx E_\theta \left[R(\tau) \frac{\partial}{\partial \theta_k} \ln p_\theta(\tau) \right],$$

where τ is a random trajectory. We can approximate the expectation on the left-hand side by generating a sample of trajectories $\tau_1, \tau_2, \ldots, \tau_N$ according to the correct distribution and then computing an empirical average. For this purpose, we need to know how to compute the derivatives

$$\frac{\partial}{\partial \theta_k} \ln p_\theta(\tau)$$

for a given trajectory τ.

Such a trajectory consists of world states $\omega_1, \omega_2, \ldots, \omega_N$ and actions $c_1, c_2, \ldots, c_N$. Since we are forcing the agent to select c_n based only on ω_n, the logarithmic probability $\ln p_\theta(\tau)$ can be expanded into a sum

$$\ln p_\theta(c_1 \mid \omega_1) + \ln p_\theta(c_2 \mid \omega_2) + \cdots + \ln p_\theta(c_N \mid \omega_N),$$

plus some additional terms that do not vary with θ.

In our three-state game, the logarithmic probability of, say, the decision history $c = (+1, -1, +1, +1)$ given the state history $\omega = (0, 1, 0, 1, 2)$ is

$$2 \ln \theta_0 + 1 \ln \theta_1 + 1 \ln(1 - \theta_1).$$

The gradient with respect to the parameter $\theta = (\theta_0, \theta_1)$ is therefore

$$\begin{aligned}\frac{\partial}{\partial\theta_0}\ln p_\theta(c\,|\,\omega) &= \frac{2}{\theta_0}\\ \frac{\partial}{\partial\theta_1}\ln p_\theta(c\,|\,\omega) &= \frac{1}{\theta_1} - \frac{1}{1-\theta_1}\end{aligned}$$

More generally, the gradient of the logarithmic derivatives of the probability of $c = (c_1, c_2, \ldots, c_N)$ given $\omega = (\omega_1, \omega_2, \ldots, \omega_N)$ are given by the counting formulas

$$\begin{aligned}\frac{\partial}{\partial\theta_0}\ln p_\theta(c\,|\,\omega) &= \frac{\#\{c_i = +1, \omega_i = 0\}}{\theta_0} - \frac{\#\{c_i = -1, \omega_i = 0\}}{1-\theta_0}\\ \frac{\partial}{\partial\theta_1}\ln p_\theta(c\,|\,\omega) &= \frac{\#\{c_i = +1, \omega_i = 1\}}{\theta_1} - \frac{\#\{c_i = -1, \omega_i = 1\}}{1-\theta_1}\end{aligned}$$

No other factors in $p_\theta(\tau)$ depend on θ, and they consequently vanish under differentiation with θ_0 or θ_1.

4 Appproximate Gradient Ascent

Using the formulas in the previous section, we can now estimate the two partial derivatives of the expected reward by means of an empirical average.

Consider for instance an agent using the policy parameters $\theta = (0.8, 0.3)$. By interacting with the world under this action policy, such an agent might produce the trajectory

$$\begin{aligned}\omega &= (1, 0, 1, 0, 1, 2)\\ c &= (-1, +1, -1, +1, +1, -1)\end{aligned}$$

from which we can extract the frequencies

#	$c_i = +1$	$c_i = -1$
$\omega_i = 0$	2	0
$\omega_i = 1$	1	2
$\omega_i = 2$	0	1

This means that the score — the gradient of the logarithmic probability of the trajectory — is

$$\nabla \ln p_\theta = \begin{pmatrix} 2/0.8 - 0/0.2 \\ 1/0.3 - 2/0.7 \end{pmatrix} \approx \begin{pmatrix} 2.50 \\ 0.48 \end{pmatrix}.$$

Since this trajectory payed a reward of $R = 1$, our estimate of the reward gradient is

$$E_\theta[R] \approx 1\,\nabla \ln p_\theta \approx \begin{pmatrix} 2.50 \\ 0.48 \end{pmatrix}.$$

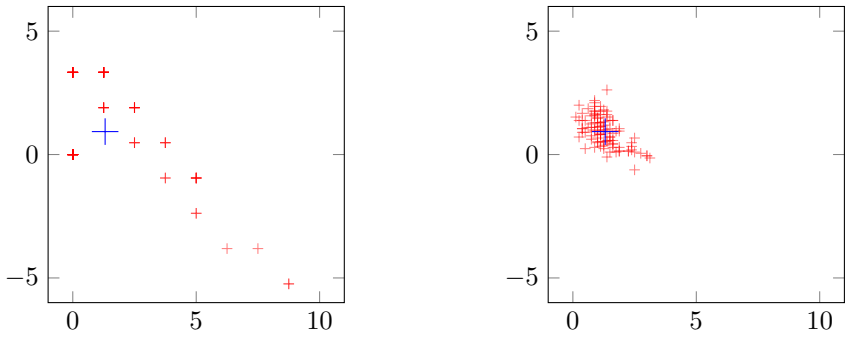

Figure 4: Reward gradient estimates based on 1 trajectory (left) and 10 trajectories (right).

The exact value is

$$\nabla E_\theta[R] \;=\; \frac{0.5}{(1-0.8(1-0.3))^2}\begin{pmatrix}(2-0.3)0.3\\ 1-0.8^2\end{pmatrix} \approx \begin{pmatrix}1.31\\ 0.93\end{pmatrix},$$

so this particular estimate is not very good. Here are a few more reward gradients estimated from a single sample:

$$\begin{pmatrix}0.0\\ 0.0\end{pmatrix}, \begin{pmatrix}3.75\\ -0.95\end{pmatrix}, \begin{pmatrix}0.0\\ 0.0\end{pmatrix}, \begin{pmatrix}1.25\\ 3.33\end{pmatrix}, \begin{pmatrix}0.0\\ 0.0\end{pmatrix}, \begin{pmatrix}2.50\\ 1.90\end{pmatrix}, \ldots$$

(The frequent occurrence of the zero vector on this list is due to the fact that all trajectories with $R = 0$ lead to a null estimate.) Figure 4 shows graphically how these estimates are distributed in the Cartesian plane, along with distribution of the average of 10 such estimates. Obviously, the pooled estimates have a much lower variance and are far superior to the pointwise ones.

Figure 5 illustrates how estimates of the reward gradient can be used to slowly climb towards a maximum expected reward: Four agents with different initial conditions play for a while, use the information they collect to estimate a gradient, and then adjust their policies according to that gradient.

5 Optimizing the Baseline

In the previous section, we used the reward gradient estimator

$$\frac{\partial}{\partial\theta_k}E_\theta[R] \;\approx\; \hat{E}_\theta\left[RS_k\right],$$

where S_k is the derivative of $\ln p_\theta(\tau)$ with respect to the kth control parameter,

$$S_k(\tau) \;=\; \frac{\partial}{\partial\theta_k}\ln p_\theta(\tau).$$

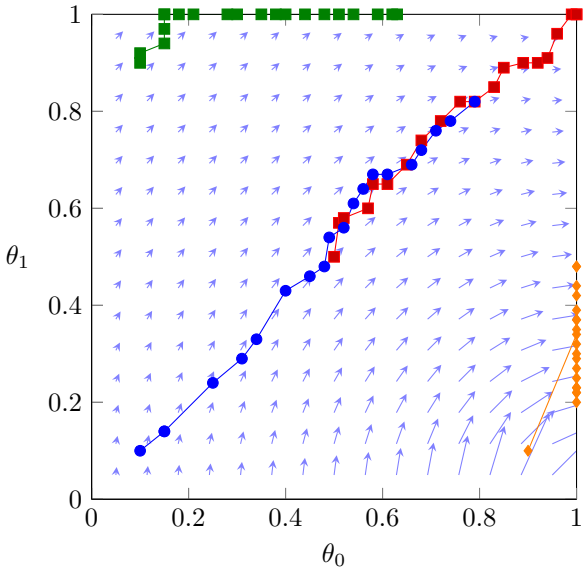

Figure 5: Four different gradient-climbing walks with different initial conditions. Each path contains 20 steps, and each gradient is estimated from 10 episodes. The step size is $\delta = 0.05$ times the (estimated) gradient.

This estimator has the right mean, but not always the smallest possible variance. To decrease its noise-sensitivity, we would therefore like to replace it with its more robust cousin,

$$\frac{\partial}{\partial\theta_k} E_\theta[R] \approx \hat{E}_\theta \left[\left(R - \frac{\hat{E}_\theta[{S_k}^2 R]}{\hat{E}_\theta[{S_k}^2]} \right) S_k \right].$$

For a data set $\tau_1, \tau_2, \ldots, \tau_N$, this amounts to computing

$$R_k^* = \frac{\frac{1}{N}\sum_{n=1}^{N} S_k(\tau_n)^2 R(\tau_n)}{\frac{1}{N}\sum_{n=1}^{N} S_k(\tau_n)^2}$$

and then

$$\frac{\partial}{\partial\theta_k} E_\theta[R] \approx \frac{1}{N}\sum_{n=1}^{N} (R(\tau_n) - R_k^*) S_k(\tau_n).$$

This computation is repeated, with different values of R_k^*, one for every control parameter θ_k.

In our small three-state example, it is possible to explicitly evaluate the expectations $E_\theta[S_k^2 R]$ and $E_\theta[S_k^2]$ for a given $\theta = (\theta_0, \theta_1)$. This would give us two numbers R_0^* and R_1^* that could be fed into the subsequent estimate $\nabla E_\theta[R]$. However, since this computation is a bit involved, and since we will not be able to perform it in the general case, I skip this problem here.

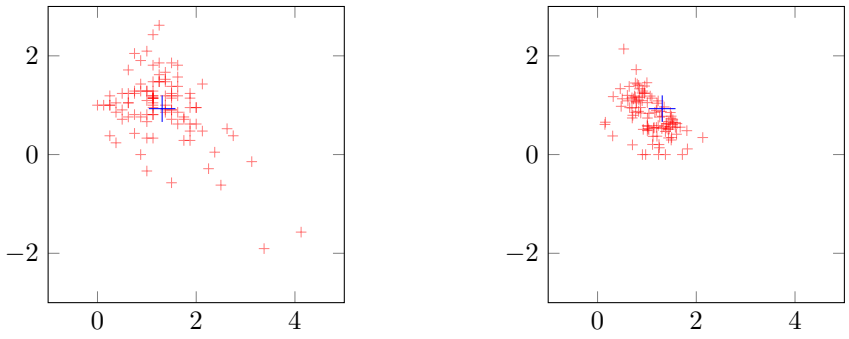

Figure 6: The distribution of the reward gradient estimator with 10 trajectories, using either the fixed baseline $R_0^* = R_1^* = 0$ (left), or two adaptable baselines R_0^* and R_1^* (right).

However, it is clear from experiment that the baseline-optimized estimator does achieve a better performance than the one that simply stretches S by the same constant R in all directions. Figure 6 contains a plot that illustrates how the variance of the estimates clearly decrease when the sample of trajectories is used to estimate a good baseline before it is used to estimate a good reward gradient.