Policy Gradient Estimation without Back-Looking Terms

Mathias Winther Madsen
mathias@micropsi-industries.com

May 25, 2016

In order to improve an action policy that produced a trajectory of T actions and T rewards, we can estimate the gradient of the total reward as a sum of T^2 terms. This note explains why, under reasonable assumptions, we can throw almost half of these terms away without biasing the estimate of the gradient.

In notation that will be defined in more detail below, this means that we can replace the gradient estimator

$$\nabla_\theta \, E_\theta[R] \;\approx\; \sum_{i=1}^{T} \sum_{j=1}^{T} \, E_\theta \left[r_i(a_{1:i}) \, \nabla_\theta \ln p_\theta(a_j \mid a_{1:j-1}) \right]$$

by the more accurate gradient estimator

$$\nabla_\theta \, E_\theta[R] \;\approx\; \sum_{i=1}^{T} \sum_{j=i}^{T} \, E_\theta \left[r_i(a_{1:i}) \, \nabla_\theta \ln p_\theta(a_j \mid a_{1:j-1}) \right].$$

The difference between the two is the bounds on the summation over j, and therefore the number of terms.

Since the terms that are discarded contribute nothing but noise to the gradient estimate, cutting them out will often have statistical as well as computational effects. In addition to explaining the reasoning behind the trick, this note also illustrates what these effects are.

1 Stepwise Actions

Suppose an agent chooses a random action a according to some probability distribution p_θ, and then receives a reward $R(a)$ in return. The agent can improve the expectation of this reward by climbing up the gradient

$$\frac{\partial}{\partial\theta} \, E_\theta \left[R \right] \;=\; \int R \, \frac{\partial}{\partial\theta} \, p_\theta \;=\; E_\theta \left[R \, \frac{\partial}{\partial\theta} \ln p_\theta \right].$$

Typically, the action a is most naturally described as a sequence of moves, $a_{1:T} = (a_1, a_2, \ldots, a_T)$. We can then factorize the probability density p_θ into conditional density factors of the form $p_\theta(a_t \mid a_{1:t-1})$. This results in the gradient estimate

$$E_\theta\left[R(a_{1:T})\,\frac{\partial}{\partial\theta}\,\ln p_\theta(a_{1:T})\right] \;=\; \sum_{t=1}^{T} E_\theta\left[R(a_{1:T})\,\frac{\partial}{\partial\theta}\,\ln p_\theta(a_t \mid a_{1:t-1})\right].$$

For a fixed value of the action sequence $a_{1:T}$, the Fisher score

$$S(a_{1:T}) \;=\; \frac{\partial}{\partial\theta}\,\ln p_\theta(a_{1:T}) \;=\; \sum_{t=1}^{T}\frac{\partial}{\partial\theta}\,\ln p_\theta(a_t \mid a_{1:t-1})$$

measures how sensitive the probability density $p_\theta(a_{1:T})$ is with respect to changes in the parameter θ. A positive value of $S(a_{1:T})$ means that the reward $R(a_{1:T})$ will show up more frequently if we increase θ, while a negative means that it will show up less frequently.

By multiplying this score of $a_{1:T}$ with the reward $R(a_{1:T})$, we translate these changes in freqeuncy into changes in reward. Thus, if

$$R(a_{1:T})\,S(a_{1:T}) \;>\; 0,$$

it will be a good idea to increase θ, either because $R(a_{1:T}) > 0$ and increasing θ makes this reward more common, or because $R(a_{1:T}) < 0$ and increasing θ makes it more rare.

For each sample path $a_{1:T}$, we thus get a piece of advice from the random variable $R(a_{1:T})\,S(a_{1:T})$. The log-likelihood formulas above tell us that we can compute the derivative of $E_\theta[R]$ by taking the mean of these pieces of advice, averaged over all values of $a_{1:T}$. The expected advice thus tells us how to increase our expected reward.

2 Stepwise Rewards

Suppose that the reward $R(a_{1:T})$ can be written as a sum of partial rewards,

$$R(a_{1:T}) \;=\; \sum_{t=1}^{T} r_t(a_{1:t}).$$

We assume that the tth partial reward depends only on the first t terms of $a_{1:T}$, as shown in Figure 1.

By the linearity of expectations, the expectation of this reward is

$$E_\theta\left[R(a_{1:T})\right] \;=\; \sum_{t=1}^{T} E_\theta\left[r_t(a_{1:t})\right],$$

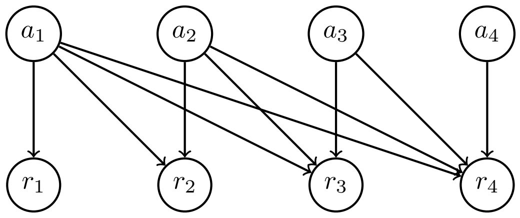

Figure 1: The statistical dependencies between actions and rewards: conditional on the early actions, the early rewards are independent on the later actions.

where all expectations are over the values of $a_{1:T}$. Note that this equality holds even if the partial rewards are statistically dependent (as they typically are). The sum of the marginal expectations is still equal to the expected sum.

By assumption, $r_t(a_{1:t})$ is conditionally independent $a_{t+1:T}$ given $a_{1:t}$. This means that we can compute $E_\theta\left[r_t(a_{1:t})\right]$ as an expectation over $a_{1:t}$ as opposed over $a_{1:T}$ without changing its value.

This follows from the fact that we can marginalize out dimensions that are irrelevant to an expectation,

$$E_{X,Y}[f(X)] \;=\; E_X[E_Y[f(X)]] \;=\; E_X[f(X)],$$

when $f = f(X)$ is conditionally independent of Y given X.

In other words, instead of computing the expectation of $r_t(a_{1:t})$ over all values of $a_{1:T}$, we can focus on a lower-dimensional slice of this trajectory space and still get the same result:

$$\frac{\partial}{\partial\theta}\,E_\theta\left[r_t(a_{1:t})\right] \;=\; \int r(a_{1:t})\,p(a_{1:T})\,da_{1:T} \;=\; \int r(a_{1:t})\,p(a_{1:t})\,da_{1:t}.$$

This allows us to use the log-likelihood trick on the more low-dimensional form of this expectation:

$$\frac{\partial}{\partial\theta}\,E_\theta\left[r_t(a_{1:t})\right] \;=\; \int r(a_{1:t})\,\frac{\partial}{\partial\theta}\,p(a_{1:t})\,da_{1:t} \;=\; E_\theta\left[r_t(a_{1:t})\,\frac{\partial}{\partial\theta}\,\ln p_\theta(a_{1:t})\right].$$

Substituting in the factorized form of $p_\theta(a_{1:t})$, this can be rewritten as

$$E_\theta\left[r_t(a_{1:t})\,\frac{\partial}{\partial\theta}\,\sum_{u=1}^{t}\,\ln p_\theta(a_u \mid a_{1:u-1})\right].$$

This gives us the gradient of the tth reward term, that is, the incentive we would have to change our policy if we only cared about what happened at time t. By adding up all of these terms, we get an estimate of the gradient of the

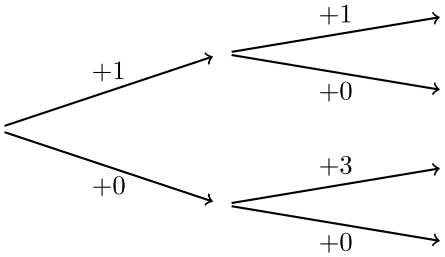

Figure 2: The microrewards triggered by the agent's actions in the game described in the text. Upper branches are 0s, and lower branches are 1s.

entire reward:

$$\begin{aligned} \frac{\partial}{\partial\theta}\, E_\theta\left[R(a_{1:T})\right] &= \sum_{t=1}^{T}\sum_{u=1}^{t}\, E_\theta\left[r_t(a_{1:t})\,\frac{\partial}{\partial\theta}\,\ln p_\theta(a_u \mid a_{1:u-1})\right] \\ & \sum_{i=1}^{T}\sum_{j=i}^{T}\, E_\theta\left[r_i(a_{1:i})\,\frac{\partial}{\partial\theta}\,\ln p_\theta(a_j \mid a_{1:j-1})\right]. \end{aligned}$$

This sum corresponds to the lower triangle of a table of terms, so we have discarded about half the terms. If the system gave rewards in a completely memoryless way, this would also cut down the variance of our gradient estimates by a factor of two. However, in a more realistic model, the late rewards can be both smaller and larger than the early rewards, and no such generalizations can be made.

3 An Example Game

Suppose the agent plays a game in which a trajectory consists of two binary moves, $a = (a_1, a_2) \in \{0,1\}^2$.

When walking through this space, the agent is given the rewards shown in Figure 2. Note that the representation of the game as a tree implicitly encodes the assumption that future actions cannot affect past rewards.

The gradient estimate associated with this game is the sum

$$\begin{aligned} \frac{\partial}{\partial\theta}\, E_\theta\left[R(a_{1:T})\right] &= E_\theta\left[r_1(a)\frac{\partial}{\partial\theta}\,\ln p_\theta(a_1)\right] + E_\theta\left[r_1(a)\frac{\partial}{\partial\theta}\,\ln p_\theta(a_2 \mid a_1)\right] + \\ & E_\theta\left[r_2(a)\frac{\partial}{\partial\theta}\,\ln p_\theta(a_1)\right] + E_\theta\left[r_2(a)\frac{\partial}{\partial\theta}\,\ln p_\theta(a_2 \mid a_1)\right], \end{aligned}$$

where the probability densities $p_\theta(a_t \mid a_{1:t-1})$ are given by the agent's parametric strategy (whatever it is).

However, by the argument in the previous section, we have

$$E_\theta \left[r_1(a) \frac{\partial}{\partial \theta} \, \ln p_\theta(a_2 \mid a_1) \right] \;=\; 0,$$

since a change in the distribution of the second action does affect the distribution of the first reward. It follows that we can drop one of these terms, leaving us with the gradient estimate

$$\begin{aligned} \frac{\partial}{\partial \theta} \, E_\theta \left[R(a_{1:T}) \right] \;&=\; E_\theta \left[r_1(a) \frac{\partial}{\partial \theta} \, \ln p_\theta(a_1) \right] + \\ &\qquad E_\theta \left[r_2(a) \frac{\partial}{\partial \theta} \, \ln p_\theta(a_1) \right] + E_\theta \left[r_2(a) \frac{\partial}{\partial \theta} \, \ln p_\theta(a_2 \mid a_1) \right]. \end{aligned}$$

The two first terms in this sum are expectations over the values of $a_{1:1}$, that is, over the set $\{0,1\}^1$. The second is an expectation over the values of $a_{1:2}$, that is, over the set $\{0,1\}^2$.

4 An Example Policy

To make things more concrete, let's suppose that the agent chooses its first action by flipping a bent coin with parameter q_1, and that it chooses its section action by using one of two coins with parameters q_0 and q_1, depending on how the first coin flip fell out.

The derivatives in the previous section can then be replaced with a gradient with respect to the parameter vector $\theta = (q, q_0, q_1)$. Specifically, we get the following table of score vectors, rewards, and probabilities:

a_1	a_2	$\nabla_\theta \, \ln p_\theta(a_1)$	$\nabla_\theta \, \ln p_\theta(a_2 \mid a_1)$	r_1	r_2	$p_\theta(a)$
0	0	$(-1/(1-q), 0, 0)^T$	$(0, -1/(1-q_0), 0)^T$	+1	+1	$(1-q)(1-q_0)$
0	1	$(-1/(1-q), 0, 0)^T$	$(0, 1/q_0, 0)^T$	+1	+0	$(1-q)q_0$
1	0	$(1/q, 0, 0)^T$	$(0, 0, -1/(1-q_1))^T$	+0	+3	$q(1-q_1)$
1	1	$(1/q, 0, 0)^T$	$(0, 0, 1/q_1)^T$	+0	+0	qq_1

Using this table, we can compute the gradient estimates $R(a)\,S(a)$ for various trajectories a. For instance, the trajectory $a = (1,0)$ gives rise to the estimate

$$R(1,0)\,S(1,0) \;=\; 0 \begin{pmatrix} 1/q \\ 0 \\ 0 \end{pmatrix} + 3 \begin{pmatrix} 0 \\ 0 \\ -1/(1-q_1) \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 0 \\ -3/(1-q_1) \end{pmatrix}.$$

5 A Sanity Check

Let's use this table to confirm by direct computation that the mean of the first reward term times the second score is zero. Using the notation $s_2(a_1, a_2) = \ln p_\theta(a_2 \mid a_1)$ for the second score term, we have

$$r_1(0,0)\, s_2(0,0)\, p_\theta(0,0) \;=\; (+1) \begin{pmatrix} 0 \\ -1/(1-q_0) \\ 0 \end{pmatrix} (1-q)(1-q_0) \;=\; \begin{pmatrix} 0 \\ q-1 \\ 0 \end{pmatrix};$$

$$r_1(0,1)\, s_2(0,1)\, p_\theta(0,1) \;=\; (+1) \begin{pmatrix} 0 \\ 1/q_0 \\ 0 \end{pmatrix} (1-q)q_0 \;=\; \begin{pmatrix} 0 \\ 1-q \\ 0 \end{pmatrix};$$

$$r_1(1,0)\, s_2(1,0)\, p_\theta(1,0) \;=\; (+0) \begin{pmatrix} 0 \\ 0 \\ -1/(1-q_1) \end{pmatrix} q(1-q_1) \;=\; 0;$$

$$r_1(1,1)\, s_2(1,1)\, p_\theta(1,1) \;=\; (+0) \begin{pmatrix} 0 \\ 0 \\ 1/q_1 \end{pmatrix} qq_1 \;=\; 0.$$

It follows that

$$E_\theta\left[r_1 s_2\right] \;=\; \begin{pmatrix} 0 \\ q-1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1-q \\ 0 \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

so this backward-looking term does indeed not contribute to the gradient.

Do note, however, that the random vector $r_1 s_2$ is not a constant: its elements take both positive and negative values, depending on what the agent did in the first step. These values just cancel out in expectation, because the expected reward at time 1 is completely insensitive to what happens at later stages.

6 Estimate Distributions

We've seen above that the random vector

$$\hat{\nabla}_1\, E_\theta[R] \;=\; r_1 s_1 + r_1 s_2 + r_2 s_1 + r_2 s_2$$

has the same mean as the random vector

$$\hat{\nabla}_2\, E_\theta[R] \;=\; r_1 s_1 + r_2 s_1 + r_2 s_2,$$

both of them being unbasied estimators of the reward gradient. Unless the vector $r_1 s_2$ is anit-correlated with the other terms, the latter is also a better estimator, since it will have lower variance. For this particular example, we can verify this observation empirically.

Throughout this section, I assume that $\theta = (q, q_0, q_1) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. Under this assumption, the table of score vectors and rewards reduces to

a_1	a_2	s_1	s_2	r_1	r_2	$p_\theta(a)$
0	0	$(-2, 0, 0)^T$	$(0, -2, 0)^T$	$+1$	$+1$	1/4
0	1	$(-2, 0, 0)^T$	$(0, 2, 0)^T$	$+1$	$+0$	1/4
1	0	$(2, 0, 0)^T$	$(0, 0, -2)^T$	$+0$	$+3$	1/4
1	1	$(2, 0, 0)^T$	$(0, 0, 2)^T$	$+0$	$+0$	1/4

By means of this table, we can compute the expectations

$$E_\theta[r_1 s_1] = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \qquad E_\theta[r_1 s_2] = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$E_\theta[r_2 s_1] = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad E_\theta[r_2 s_2] = \begin{pmatrix} 0 \\ -1/2 \\ -3/2 \end{pmatrix}$$

The gradient of the expected reward is thus the vector

$$\nabla_\theta \, E_\theta[r_1 + r_2] \;=\; \begin{pmatrix} -1+0+1+0 \\ 0+0+0-1/2 \\ 0+0+0-3/2 \end{pmatrix} \;=\; \begin{pmatrix} 0 \\ -1/2 \\ -3/2 \end{pmatrix},$$

and this is the vector that both $\hat{\nabla}_1 \, E_\theta[R]$ and $\hat{\nabla}_2 \, E_\theta[R]$ approximates.

The scatterplot in Figure 3 gives a visual sense of how well the two estimators do, and how they compare to each other. Figure 4 visualizes the same information in a different way, and based on a larger data set.

7 Mean Error

As the figures suggests, the main effect of throwing away $r_1 s_2$ has been to cut down the variance of the estimator around the second gradient component,

$$\frac{\partial}{\partial q_0} E_\theta[R] \;=\; -\frac{1}{2}.$$

This makes sense in terms of the values that the deleted term $r_1 s_2$ takes, to wit,

$$\begin{pmatrix} 0 \\ -2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

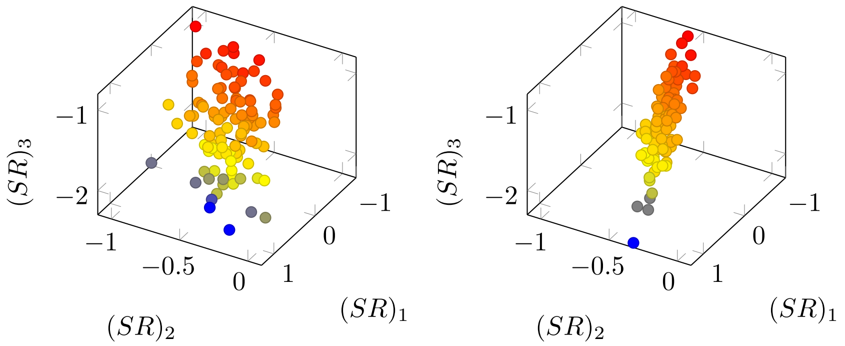

Figure 3: A sample of gradient estimates with and without the term of mean 0. Each of the 100 points represents the average of 100 samples from RS.

Since these four values all occur with equal probability, the variance of the random vector $r_1 s_2$ is

$$E_\theta[\|r_1 s_2 - \mathbf{0}\|^2] \;=\; \frac{1}{4}2^2 + \frac{1}{4}2^2 + \frac{1}{4}0 + \frac{1}{4}0 \;=\; 2.$$

If this term were completely uncorrelated with any of the other terms, deleting it would increase the precision of the estimator RS by $1/2$ unit of squared distance per sample. When the gradient estimate is computed from 100 samples as in Figure 3, this would correspond to a drop in variance (and therefore mean squared error) by 0.02 units of squared distance.

However, $r_1 s_2$ is not completely uncorrelated with $r_1 s_1 + r_2 s_1 + r_2 s_2$, and the actual gain is bit larger. I have found empirically that the mean squared error of $\hat{\nabla}_1\, E_\theta[R]$ is about 25.5, while the mean squared error of $\hat{\nabla}_2\, E_\theta[R]$ is about 21.5. This suggests that the real drop in mean loss is about 4.0 units of squared distance rather than 2.

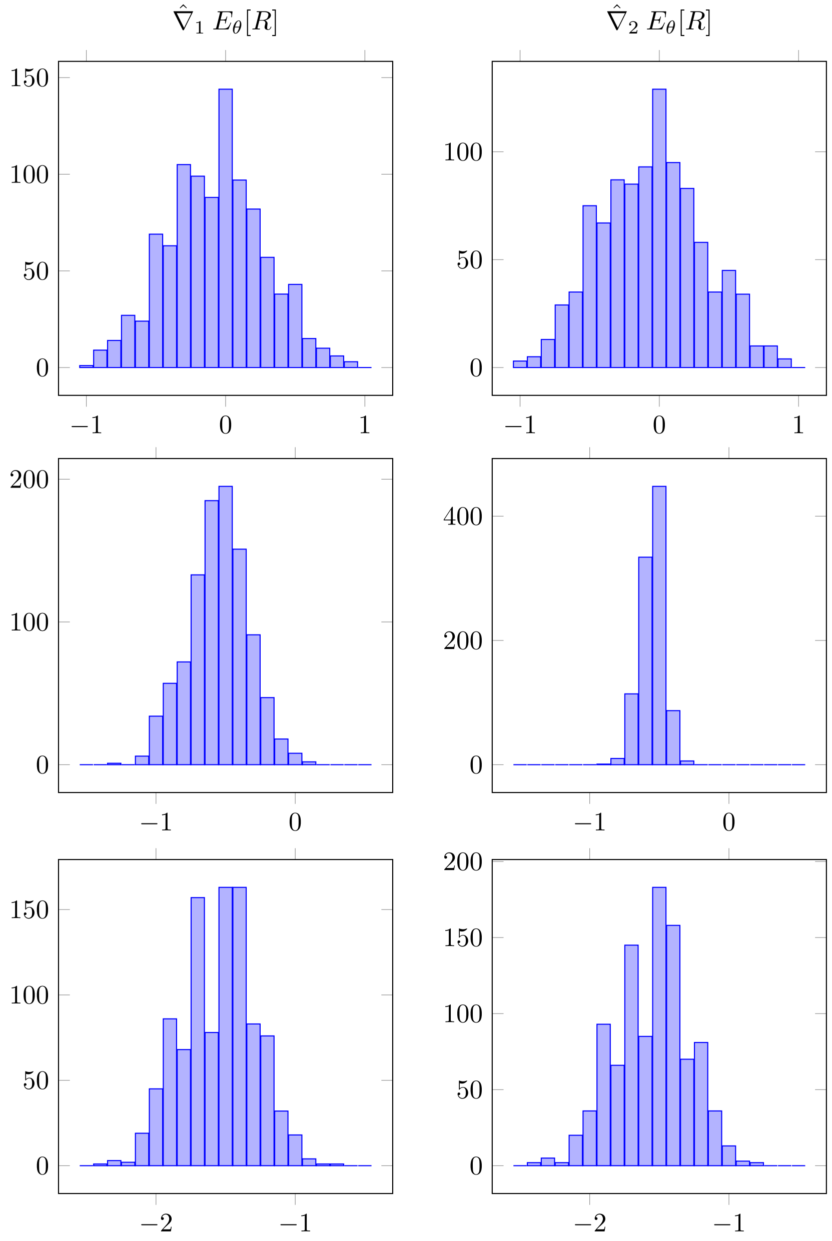

Figure 4: The distributions of the coordinates of the two estimators of $\nabla E_\theta[R]$. The lefthand histograms show the distribution of the four-term sum, while the righthand ones show the distribution of the three-term sum. Each estimate is an average of 100 samples, and each histogram contains 1000 estimates.