Variance Reduction in Policy Gradient

David S. Rosenberg

NYU: CDS

November 3, 2021

Contents

Recap: policy gradient for contextual bandits

[Online] Stochastic k-armed contextual bandit

Stochastic k-armed contextual bandit

1. Environment samples **context** and **rewards vector** jointly, iid, for each round:

$$(X, R), (X_1, R_1), \ldots, (X_T, R_T) \in \mathcal{X} \times \mathbb{R}^k \text{ i.i.d. from } P,$$

   where $R_t = (R_t(1), \ldots, R_t(k)) \in \mathbb{R}^k$.

2. For $t = 1, \ldots, T$,
   1. Our algorithm **selects action** $A_t \in \mathcal{A} = \{1, \ldots, k\}$ based on X_t and history

$$\mathcal{D}_t = \Big((X_1, A_1, R_1(A_1)), \ldots, (X_{t-1}, A_{t-1}, R_{t-1}(A_{t-1})) \Big).$$

   2. Our algorithm **receives reward** $R_t(A_t)$.

- We **never observe** $R_t(a)$ for $a \neq A_t$.

Contextual bandit policies

- A contextual bandit policy at round t
  - gives a conditional distribution over the action A_t to be taken
  - conditioned on the history $\mathcal{D}_t$ and the **current context** X_t.
- In this module, we consider policies parameterized by θ: $\pi_\theta(a \mid x)$, for $\theta \in \mathbb{R}^d$.
- We denote the θ used at round t by θ_t, which will depend on $\mathcal{D}_t$.
- At round t, action $A_t \in \mathcal{A} = \{1, \ldots, k\}$ is chosen according to

$$\mathbb{P}(A_t = a \mid X_t = x, \mathcal{D}_t) = \pi_{\theta_t}(a \mid x).$$

Example: multinomial logistic regression policy

- An example parameterized policy:

$$\pi_\theta(a \mid x) = \frac{\exp\left(\theta^T \phi(x,a)\right)}{\sum_{a'=1}^{k} \exp\left(\theta^T \phi(x,a')\right)},$$

where $\phi(x,a) : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is a joint feature vector.

- And $\theta^T \phi(x,a)$ can be replaced by a more general $g_\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$.
- The differentiability w.r.t. θ is key to a policy gradient method.

How to update the policy?

- Objective function for policy gradient:

$$J(\theta) \quad := \quad \mathbb{E}_\theta \left[R(A)\right].$$

- Idealized policy gradient is to iteratively update θ as:

$$\theta_{t+1} \leftarrow \theta_t + \eta \nabla J(\theta_t).$$

- Policy gradient theorem from last module gives an unbiased estimate of $\nabla J(\theta_t)$.

Unbiased estimate for the gradient

- Consider round t of SGD for optimizing $J(\theta)$.
- We play A_t from $\pi_{\theta_t}(a \mid X_t)$ and record $(X_t, A_t, R_t(A_t))$.
- To update θ_t, we need an unbiased estimate of $\nabla J(\theta_t)$.
- Last time we showed that

$$\mathbb{E}_{\theta_t}\left[R_t(A_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right] = \nabla_\theta J(\theta_t)$$

- Suggests the following iterative update:

$$\theta_{t+1} \leftarrow \theta_t + \eta R_t(A_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t).$$

- This is the basic **policy gradient method**.

Using a baseline

Subtracting a baseline from reward

- Our objective function is

$$J(\theta) = \mathbb{E}_\theta \left[R(A) \right].$$

- Suppose we introduce a new reward vector $R_0 = R - b$, for constant $b \in \mathbb{R}$.
- Then

$$J_b(\theta) = \mathbb{E}_\theta \left(R_0(A) \right) = \mathbb{E}_\theta \left(R(A) \right) - b.$$

- Obviously, $J(\theta)$ and $J_b(\theta)$ have the same maximizer θ^*.
- And $\nabla_\theta J(\theta) = \nabla_\theta J_b(\theta)$.

Policy gradient with a baseline

- If we just plug in the shift to our gradient estimators, we get:

$$\begin{aligned} J(\theta): \quad \theta_{t+1} &\leftarrow \theta_t + \eta R_t(A_t) \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \\ J_b(\theta): \quad \theta_{t+1} &\leftarrow \theta_t + \eta \left(R_t(A_t) - b\right) \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \end{aligned}$$

  where b is called the **baseline**.
- The updates are different, so we'll get different optimization paths.
- Is $\left(R_t(A_t) - b\right) \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$ still unbiased for $\nabla J(\theta)$?
- We'll show that it is, even when we allow a random baseline $B_t = f(\mathcal{D}_t, X_t)$.
- The hope is to find a B_t that reduces the variance of the gradient estimate,
  - getting us to a better policy, faster.

You might remember from the module on policy gradient for bandits that eventually we multiply the baseline by some function of the action A_t to get our control variate. If the control variate can depend on A_t, why can't the baseline B_t also depend on A_t, like $B_t = f(\mathcal{D}_t, X_t, A_t)$? There's nothing that prohibits us from considering such a B_t as a baseline. However, we'd have to be able to compute the expectation of the control variate and show that it's zero, which won't be the case for all such B_t.

The score has zero expectation

- Let $p(a;\theta)$ be a parametric distribution on a finite set $\mathcal{A}$.
- The **score function** is defined as $s(a,\theta) = \nabla_\theta \log p(a;\theta)$.
- Then $\mathbb{E}_{A\sim p(a;\theta)}[s(A,\theta)] = 0$ for any θ.
- **Proof:** (assuming differentiability as needed)

$$
\begin{aligned}
\mathbb{E}_{A\sim p(a;\theta)}[s(A,\theta)] &= \mathbb{E}_{A\sim p(a;\theta)}[\nabla_\theta \log p(a;\theta)] \\
&= \mathbb{E}_{A\sim p(a;\theta)}\left[\frac{\nabla_\theta p(a;\theta)}{p(a;\theta)}\right] \\
&= \sum_{a\in\mathcal{A}} p(a;\theta)\left[\frac{\nabla_\theta p(a;\theta)}{p(a;\theta)}\right] = \sum_{a\in\mathcal{A}} \nabla_\theta p(a;\theta) \\
&= \nabla_\theta\left[\sum_{a\in\mathcal{A}} p(a;\theta)\right] = \nabla_\theta[1] = 0
\end{aligned}
$$

Estimate with baseline is unbiased

- Allow θ_t and the baseline B_t at round t to depend on $\mathcal{D}_t$ and X_t:

$$\begin{aligned} B_t &= f(\mathcal{D}_t, X_t) \quad \text{for some function } f\text{, and let} \\ \theta_t &= g(\mathcal{D}_t) \quad \text{for some function } g. \end{aligned}$$

- So

$$\begin{aligned} &\mathbb{E}\left[B_t \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[B_t \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \mid \mathcal{D}_t, X_t\right]\right] \quad \text{inner expectation over } A_t \sim \pi_{\theta_t}(\cdot \mid X_t) \\ &= \mathbb{E}\left[B_t \mathbb{E}\left[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \mid \mathcal{D}_t, X_t\right]\right] \quad \text{taking out what is known} \\ &= \mathbb{E}\left[B_t 0\right] = 0. \end{aligned}$$

- Therefore $(R_t(A_t) - B_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$ is an unbiased estimate of $\nabla J(\theta)$.
  - for any choice of f and g above.

- Let's show $\mathbb{E}[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \mid \mathcal{D}_t, X_t] = 0$ very explicitly. First, the only thing random in the expectation is $A_t \sim \pi_{\theta_t}(\cdot \mid X_t)$. Note that θ_t is generally random, via its dependence on $\mathcal{D}_t$, but we're conditioning on $\mathcal{D}_t$, so θ_t is constant here.

- Previously, we showed $\mathbb{E}_{A \sim p(a;\theta)}[s(A,\theta)] = 0$ for any θ, where $s(a,\theta) = \nabla_\theta \log p(a;\theta)$. We'll try to put things in these terms...

- Define $p(a;\theta,x) = \pi_\theta(a \mid x)$, which gives a distribution on $\mathcal{A}$ for every $\theta \in \Theta$ and $x \in \mathcal{X}$. Define the corresponding score function as $s(a,\theta;x) = \nabla_\theta \log p(a;\theta,x)$. Then we know $\mathbb{E}_{A \sim p(a;\theta,x)}[s(A,\theta;x)] = 0$ for every θ and x, which we apply in the last step below. Let

$$
\begin{aligned}
r(d,x) &:= \mathbb{E}[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \mid \mathcal{D}_t = d, X_t = x] \\
&= \mathbb{E}[\nabla_\theta \log p(A_t;\theta_t,x) \mid \mathcal{D}_t = d, X_t = x] \\
&= \mathbb{E}[s(A_t,\theta_t;x) \mid \mathcal{D}_t = d, X_t = x] \\
&= \mathbb{E}[s(A_t,g(d);x) \mid \mathcal{D}_t = d, X_t = x] \quad \text{(only } A_t \text{ is random)} \\
&= \mathbb{E}_{A_t \sim p(a;g(d),x)}[s(A_t,g(d);x)] \\
&= 0.
\end{aligned}
$$

So $r(\mathcal{D}_t, X_t) = \mathbb{E}[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \mid \mathcal{D}_t, X_t] = 0$.

What to use for the baseline?

- In round t, our unbiased estimate of $\nabla_\theta J(\theta_t)$ is

$$(R_t(A_t) - B_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t).$$

- We're trying to "reduce the variance" of this estimate.
- But what is the "variance"?
- This expression is generally a **vector** in $\mathbb{R}^d$, since $\theta \in \mathbb{R}^d$.
- There is no scalar "variance" we can just try to minimize.
- We'll revisit this shortly...

Basic approach to the baseline

- The easiest thing to use for a baseline is

$$B_t = \frac{1}{t-1}\sum_{i=1}^{t-1} R_i(A_i).$$

- Think B_t as a **value estimate** for policy $\pi_{\theta_t}(a \mid x)$: $B_t \approx \mathbb{E}_{\theta_t}[R_t(A_t)]$.
- We can think of the baseline as shifting the rewards, making some positive and some negative.
- In practice, it's usually much better than $B_t \equiv 0$.

Input-dependent baseline

- What if rewards R_t are generally smaller for some inputs X_t than others?
- We can try to choose $B_t \approx \mathbb{E}_{\theta_t}[R(A_t) \mid X_t]$.
- Learn $\hat{r}_t(x) \approx \mathbb{E}_{\theta_t}[R_t(A_t) \mid X_t = x]$ from history $\mathcal{D}_t$.
- Use $B_t = \hat{r}_t(X_t)$ as a baseline for round t.
- We can learn $\hat{r}_t(x)$ in an online manner, at the same time as we learn our policy.
  - e.g. in t'th round take a gradient step to reduce $(R_t(A_t) - \hat{r}_t(X_t))^2$.
- This is an approach suggested in Sutton's book [SB18, Sec 13.4].

- If you're concerned that we're trying to estimate $\mathbb{E}_{\theta_t}[R(A_t) \mid X_t]$ with only a single action A_t drawn from θ_t... well that's a reasonable concern!
- Remember, we don't need a perfect estimate of $\mathbb{E}_{\theta_t}[R(A_t) \mid X_t]$ — this is just to reduce the variance and doesn't affect the bias.
- In estimating $\mathbb{E}_{\theta_t}[R(A_t) \mid X_t]$, there are a couple of bias/variance tradeoffs in play. If we use all the historical rewards, then our estimate will be biased, since only the last of those rewards is actually drawn from θ_t. We can importance-weight to get an unbiased objective function, at the cost of increased variance. We can also use a shorter history, where presumably policies from more recent rounds are more similar to θ_t. Thus will also increase the variance but should decrease the bias.

“Optimal” baseline

"Optimal" baseline

- Our gradient estimator is $(R_t(A_t) - B_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$.
- This a vector, so it's not clear what it means to "minimize the variance."
- This random vector has a covariance **matrix**.
- Let's allow a different baseline $B_t(\alpha)$ for each entry of the gradient estimate.
  - (We did this for the multiarmed bandit in the previous module.)
- Now we can attempt to minimize the variance for each entry separately.
- This ignores off-diagonal entries of the covariance matrix.

The entry variance

- Define
$$G_t^j = \left[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right]_j.$$
- That is, G_t^j is the j'th entry of the score at round t.
- Let's consider the variance of the jth entry of our estimator with baseline b:

$$\begin{aligned}
V_j := \operatorname{Var}\left(\left[(R_t(A_t) - b)\,\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right]_j\right) \\
= \operatorname{Var}\left((R_t(A_t) - b)\, G_t^j\right) \\
= \mathbb{E}\left[(R_t(A_t) - b)^2 \left(G_t^j\right)^2\right] - \left[\mathbb{E}\,(R_t(A_t) - b)\, G_t^j\right]^2 \\
= \mathbb{E}\,(R_t(A_t) - b)^2 \left(G_t^j\right)^2 - \left[\mathbb{E}\left[R_t(A_t)\, G_t^j\right]\right]^2
\end{aligned}$$

"Optimal" baselines

- Differentiating V_j w.r.t. b:

$$
\begin{aligned}
V_j &= \mathbb{E}\left(R_t(A_t)-b\right)^2\left(G_t^j\right)^2-\left[\mathbb{E}\left[R_t(A_t)G_t^j\right]\right]^2 \\
\frac{dV_j}{db} &= \frac{d}{db}\left(\mathbb{E}\left[R_t(A_t)^2\left(G_t^j\right)^2\right]+b^2\mathbb{E}\left(G_t^j\right)^2-2b\mathbb{E}R_t(A_t)\left(G_t^j\right)^2\right) \\
&= 2b\mathbb{E}\left(G_t^j\right)^2-2\mathbb{E}R_t(A_t)\left(G_t^j\right)^2
\end{aligned}
$$

- Solving for b in $\frac{dV_j}{db}=0$:

$$
b_t^j := \frac{\mathbb{E}\left[R_t(A_t)\left(G_t^j\right)^2\right]}{\mathbb{E}\left[\left(G_t^j\right)^2\right]}
$$

"Optimal baselines"

- So estimate for the j'th entry should ideally use baseline b_t^j.
- We can try to estimate the expectations from the logs:

$$\mathbb{E}\left[R_t(A_t)\left(G_t^j\right)^2\right] \approx \frac{1}{t-1}\sum_{i=1}^{t-1} R_i(A_i)\left(G_i^j\right)^2$$

$$\mathbb{E}\left[\left(G_t^j\right)^2\right] \approx \frac{1}{t-1}\sum_{i=1}^{t-1}\left(G_i^j\right)^2.$$

- This derivation is based on Berkeley's CS 285: Lecture 5, Slide 19, but their slide is quite vague on specifics. They don't seem to acknowledge that the gradient is a vector or that they'll need a different baseline for each entry. They also don't indicate how to estimate the expectations. Their interpretation of the resulting b_t^j in that slide is that it's "just expected reward, but weighted by gradient magnitudes!". More references are given on the resources slide at the end of this deck.
- If you're after an "optimal" **scalar** baseline, you could try minimizing the trace of the covariance matrix or $\|(R_t(A_t) - B_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\|_2^2$.

"Optimal baselines" putting it together

- Let θ_t^j denote the j'th entry of θ_t.
- Update step at round t with these baselines is

$$\theta_{t+1}^j \leftarrow \theta_t^j + \eta \left(R_t(A_t) - B_t^j \right) \left[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \right]_j ,$$

where

$$\begin{aligned} B_t^j &= \left[\frac{1}{t-1} \sum_{i=1}^{t-1} R_i(A_i) \left(G_i^j \right)^2 \right] / \frac{1}{t-1} \sum_{i=1}^{t-1} \left(G_i^j \right)^2 \\ G_i^j &= \left[\nabla_\theta \log \pi_{\theta_t}(A_i \mid X_i) \right]_j \end{aligned}$$

Actor-Critic methods

Recall the policy gradient derivation

- Recall the following formulation of the value function:

$$\begin{aligned}\mathbb{E}_\theta\left[R(A)\right] &= \mathbb{E}_X\left[\mathbb{E}_{A|X\sim\theta}\left[\mathbb{E}_{R|X}\left[R(A)\mid A,X\right]\mid X\right]\right]\\ &= \mathbb{E}_X\left[\sum_{a=1}^{k}\pi_\theta\left(a\mid X\right)\mathbb{E}_{R|X}\left[R(A)\mid A=a,X\right]\right]\end{aligned}$$

- So

$$\nabla_\theta\mathbb{E}_\theta\left[R(A)\right] = \mathbb{E}_X\left[\sum_{a=1}^{k}\nabla_\theta\left[\pi_\theta\left(a\mid X\right)\right]\mathbb{E}_{R|X}\left[R(A)\mid A=a,X\right]\right]$$

- In PG, we use a “clever trick” to get an unbiased estimate of $\nabla\mathbb{E}_\theta\left[R(A)\right]$ from $(X_t, A_t, R_t(A_t))$.

Plug-in a value estimate

- We have

$$\nabla_\theta \mathbb{E}_\theta [R(A)] \;=\; \mathbb{E}_X \left[\sum_{a=1}^{k} \nabla_\theta [\pi_\theta (a \mid X)] \, \mathbb{E}_{R|X} [R(A) \mid A = a, X] \right]$$

- Suppose we had $\hat{r}(x, a) \approx \mathbb{E}[R(A) \mid A = a, X = x]$.
- Then we get

$$\begin{aligned} \nabla_\theta \mathbb{E}_\theta [R(A)] &\approx \mathbb{E}_X \left[\sum_{a=1}^{k} \nabla_\theta [\pi_\theta (a \mid X)] \, \hat{r}(X, a) \right] \\ &\approx \sum_{a=1}^{k} \nabla_\theta [\pi_\theta (a \mid X_t)] \, \hat{r}(X_t, a) \end{aligned}$$

- The last step is a one-sample Monte Carlo estimate for $\mathbb{E}_X$.

Online update of value estimator

- Parametrize value estimator: $\hat{r}_w(x, a)$.
- We'll fit w by SGD on square loss:

$$\nabla_w (\hat{r}_w(X,A) - R(A))^2 \;=\; 2(\hat{r}_w(X,A) - R(A))\nabla_w \hat{r}_w(X,A).$$

- This is the step direction, and we can absorb the 2 into the step size multiplier.
- So value estimator update is

$$w_{t+1} \leftarrow w_t - \eta_w (\hat{r}_w(X,A) - R(A))\nabla_w \hat{r}_w(X,A)$$

- Setting the step size can be done with the usual approaches.

Actor-critic method

Definition (Actor-critic method, [SB18, p. 321])

Methods that learn approximations to both policy and value functions are often called **actor-critic** methods, where **actor** is a reference to the learned policy, and **critic** is a reference to the learned value function.

- Initialize θ_1 and w_1 (learning rates η_θ and η_w).
- For each round t:
  - Observe X_t, choose action $A_t \sim \pi_{\theta_t}(a \mid X_t)$, receive $R_t(A_t)$.
  - **[Update actor]** $\theta_{t+1} \leftarrow \theta_t + \eta_\theta \left[\sum_{a=1}^{k} \nabla_\theta \left[\pi_\theta\left(a \mid X_t\right)\right] \hat{r}_{w_t}(X_t, a)\right]$
  - **[Update critic]** $w_{t+1} \leftarrow w_t - \eta_w \left(\hat{r}_w(X_t, A_t) - R_t(A_t)\right) \nabla_w \hat{r}_w(X_t, A_t)$

A **slow** direct method: we're slowly adjusting our policy towards larger [estimated] value.

Compare to policy gradient

- The estimate of $\nabla_\theta \mathbb{E}[R(A)]$ in policy gradient is

$$(R_t(A_t) - B_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t).$$

- It's unbiased, but it has variance coming from R_t, A_t, and X_t.
- The actor-critic estimate of $\nabla_\theta \mathbb{E}[R(A)]$ is

$$\sum_{a=1}^{k} \nabla_\theta \left[\pi_\theta(a \mid X_t)\right] \hat{r}(X_t, a).$$

- Variance comes from X_t and from $\hat{r}$, but the variance of $\hat{r}$ decreases as we get more data.
- The actor-critic estimate is **biased** by $\hat{r}$, in general, but we expect it to have **less variance**.

References

Resources

- In this module and the previous module, we present approaches to the online contextual bandit problem. The policy gradient and actor-critic methods are usually presented in the more general setting of reinforcement learning. The standard textbook reference is [SB18, Ch 13] and [Wil92] is the original paper for "REINFORCE", which is policy gradient in the reinforcement learning setting.
- In [GBB04] they approach the "optimal baseline" problem in a more general setting, but they define optimality in terms of the trace of the covariance matrix of the gradient estimate. This ignores correlations between components, as we do here. The same approach is taken in [WRD+18, Appendix A].
- One can find something similar to our "optimal" baseline approach (with a different baseline for each component of the gradient estimate) in [PS08, Sec 3.2], though they're in the full reinforcement learning setting.

References I

[GBB04] Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter, *Variance reduction techniques for gradient estimates in reinforcement learning*, J. Mach. Learn. Res. **5** (2004), 1471–1530.

[PS08] Jan Peters and Stefan Schaal, *Reinforcement learning of motor skills with policy gradients*, Neural Networks **21** (2008), no. 4, 682–697.

[SB18] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: An introduction*, A Bradford Book, Cambridge, MA, USA, 2018.

[Wil92] Ronald J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, Machine Learning **8** (1992), no. 3-4, 229–256.

References II

[WRD+18] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M. Bayen, Sham M. Kakade, Igor Mordatch, and Pieter Abbeel, *Variance reduction for policy gradient with action-dependent factorized baselines*, 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.