Variance Reduction in Policy Gradient

David S. Rosenberg

NYU: CDS

March 31, 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 whole conditional distribution $\pi_\theta(a \mid x)$ can also be represented as a neural network with a softmax output.
- 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.
- Then
$$J_b(\theta) = \mathbb{E}_\theta (R_0(A)) = \mathbb{E}_\theta \left(R(A) \right) - b.$$
- 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:

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

  where b is called the **baseline**.
- The updates are different, so we'll get different optimization paths.
- Is $(R_t(A_t) - b) \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$ still unbiased for $\nabla J(\theta)$?
- Doesn't really look like it.
- But we'll show that it is.
- Then we'll discuss choices for b.

The score has zero expectation

- The **score** is the gradient of the likelihood w.r.t. the parameter.
- Let $p_\theta(a)$ be a parametric distribution on finite set $\mathcal{A}$.
- Then $\mathbb{E}_{A\sim p_\theta(a)}\left[\nabla_\theta \log p_\theta(A)\right] = 0$.
- **Proof:** (assuming differentiability as needed)

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

Estimate with baseline is unbiased

- Since the score has expectation 0,

$$\begin{aligned}\mathbb{E}\left[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right] &= \mathbb{E}_{X_t}\left[\mathbb{E}_{A_t \mid X_t}\left[\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t) \mid X_t\right]\right] \\ &= \mathbb{E}_{X_t}[0] = 0.\end{aligned}$$

- So

$$\mathbb{E}\left[(R_t(A_t) - b)\,\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right] = \mathbb{E}\left[R_t(A_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right].$$

- Therefore, $(R_t(A_t) - b)\,\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$ is an unbiased estimate of $\nabla J(\theta)$.
- We can also think of this as a control variate estimator – what's the control variate? [Homework]

What to use for the baseline?

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

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

- We're trying to "reduce the variance" of this estimate.
- But what is the "variance"?
- This expression is generally a **vector**.
- There is no scalar "variance" we can just try to minimize.
- So think very carefully if you see somebody claim that a particular b gives "minimal variance."

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 of this as an estimate of the value function: $b_t \approx \mathbb{E}_{\theta_t}[R_t(A_t)]$.
- So b_t is a **value estimate** for policy $\pi_{\theta_t}(a \mid x)$.
- This choice seems reasonable.
- It should make some rewards positive and some rewards negative.
- I don't know a great mathematical justification for this choice
- In practice, it's usually much better than $b_t = 0$.

Input-dependent baseline

- What if we generally get lower rewards R_i for some inputs X_i than others?
- Can we have the baseline b_i depend on the input X_i?
- Yes!
- But how to choose $b_t(X_t)$?
- We can think of having $b_t(x) \approx \mathbb{E}_{\theta_t}[R(A_t) \mid X = x]$.

Learning the baseline

- Learn function $\hat{r}_t(x)$ to predict the reward for a given input x.
- That is, find $\hat{r}_t(x) \approx \mathbb{E}_{\theta_t}[R_t(A_t) \mid X_t = x]$.
- So $\hat{r}_t(x)$ is a context-conditional value estimate for policy $\pi_{\theta_t}(a \mid x)$.
- Use $\hat{r}_t(X_t)$ as the 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].

“Optimal” baseline

"Optimal" baseline

- Notice that we're estimating a gradient, which is a vector.
- Let's allow a different baseline $b(\alpha)$ for the estimate of each entry of the gradient.
  - (We did this for the multiarmed bandit as well in the previous module.)
- Could use the general result from our covariate module, but seems easier to repeat the analysis.
- Define

$$g(a,x) = \nabla_\theta \log \pi_{\theta_t}(a \mid x).$$

- And define

$$G_t^j = [g(A_t, X_t)]_j \,.$$

- That is, G_t^j is the j'th entry of the score at round t.

"Optimal" baselines

- Let's consider the variance of the jth entry of our estimator:

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

- And

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

"Optimal baselines"

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

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

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

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

- Warning: I haven't seen this derivation in the literature. It's 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.
- The interpretation of the resulting b_t^j in that slide is that it's "just expected reward, but weighted by gradient magnitudes!".

"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

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

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

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

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

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

Online update of value estimator

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

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

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

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

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

This is like 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

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

- 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

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

- This has variance coming from X_t and from $\hat{r}$, but the variance of $\hat{r}$ decreases as we fit it on more data.
- The new estimate is **biased**, but 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 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.

References I

[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.