Policy Gradient for Contextual Bandits

David S. Rosenberg

NYU: CDS

March 26, 2021

Contents

Recap of the contextual bandit setting

[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

- Note: None of the discussion below depends on a specific policy class.
- However, it's helpful to have a policy class in mind.
- Let

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

  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 $g_\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$,
  - e.g. a neural network.
- 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.

SGD for CPMs vs policy gradient

Conditional Probability Modeling (CPM)

- Input space $\mathcal{X}$
- Label space $\mathcal{Y}$
- Hypothesis space of functions $x \mapsto p_\theta(y \mid x)$
- Parameterized by $\theta \in \Theta$
- For any θ and x, $p_\theta(y \mid x)$ is a distribution on $\mathcal{Y}$.
- Mathematically, no different from a policy.

Conditional Probability Modeling (CPM)

- Given training set $\mathcal{D} = ((X_1, Y_1), \ldots, (X_n, Y_n)$ iid from $P_{\mathcal{X} \times \mathcal{Y}}$.
- Maximum likelihood estimation for dataset:

$$\begin{aligned} \theta &\in \arg\max_{\theta \in \Theta} \prod_{i=1}^{n} p_\theta(Y_i \mid X_i) \\ \iff \theta &\in \arg\max_{\theta \in \Theta} \sum_{i=1}^{n} \log\left[p_\theta(Y_i \mid X_i)\right] \end{aligned}$$

SGD for MLE of CPM

- Consider SGD to compute the MLE of a CPM.
- For observation (X_i, Y_i), we'll update θ by

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log p_\theta(Y_i \mid X_i)$$

  for some learning rate $\alpha > 0$.
- This updates θ so there's more probability mass on **correct output** Y_i for input X_i.

The policy gradient update

- Below we'll derive the following policy gradient update to θ:

$$\theta \leftarrow \theta + \alpha R_i(A_i) \nabla_\theta \log \pi_\theta(A_i \mid X_i)$$

- Compare this to the SGD update for CPM:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log p_\theta(Y_i \mid X_i)$$

- Note that if $R_i(A_i) \equiv 1$, the two are equivalent.

Policy gradient vs conditional probability modeling

- In maximum likelihood with CPM, we're making the correct label Y_i more likely.
- With policy gradient, we're always increasing the probability of selected actions (with positive rewards), but the increase is larger with big positive rewards than with small positive rewards.

Policy gradient for contextual bandits

How to update the policy?

- Let A be an action chosen according to $\pi(a;\theta)$.
- Let $(X,R) \in \mathcal{X} \times \mathbb{R}^k \sim P$ be a generic context/reward vector pair.
- We want to find θ to maximize

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

- And now we differentiate w.r.t. θ.... but first...

Clever Trick

- But first a clever trick:

$$\nabla_\theta \log \pi_\theta(a \mid x) \quad = \quad \frac{\nabla_\theta \pi_\theta(a \mid x)}{\pi_\theta(a \mid x)}$$

- Rearranging, we get

$$\nabla_\theta \pi_\theta(a \mid x) \quad = \quad \pi_\theta(a \mid x) \nabla_\theta \log \pi_\theta(a \mid x).$$

- This assumed that $\pi_\theta(a \mid x) > 0$.

Gradient of Objective Function

- For a given θ, we want to find direction to increase $J(\theta)$:

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

- In the setting of reinforcement learning, this result is often referred to as the Policy Gradient Theorem.

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

$$\nabla_\theta J(\theta_t) = \mathbb{E}_{\theta_t}\left[R(A)\nabla_\theta \log \pi_{\theta_t}(A \mid X)\right],$$

  where $A \sim \pi_{\theta_t}(a \mid X)$.
- $(X_t, A_t, R_t(A_t))$ has exactly the right distribution.
- So

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

  is an unbiased estimate of $\nabla_\theta J(\theta_t)$.

- Suppose we ran multiple rounds with the same policy θ. We can also get a gradient estimate (a better one) by averaging all those results together. For convenience, we'll just index them by $1, \ldots, N$. So the gradient estimate would be

$$\theta \leftarrow \theta + \eta \left[\frac{1}{N} \sum_{i=1}^{N} R_i(A_i) \nabla_\theta \log \pi_\theta(A_i \mid X_i) \right].$$

- If each of those rounds had a different policy θ_i, then we could use importance sampling to get an unbiased estimate:

$$\theta \leftarrow \theta + \eta \left[\frac{1}{N} \sum_{i=1}^{N} \frac{\pi_{\theta_i}(A_i \mid X_i)}{\pi_\theta(A_i \mid X_i)} R_i(A_i) \nabla_\theta \log \pi_\theta(A_i \mid X_i) \right].$$

Basic policy gradient for contextual bandits

Policy gradient algorithm (step size $\eta > 0$):

1. Initialize $\theta_1 = 0 \in \mathbb{R}^k$.
2. For each round $t = 1, \ldots, T$:
   1. Observe context X_t.
   2. Choose action A_t from distribution $\mathbb{P}(A_t = a \mid X_t) = \pi_{\theta_t}(a \mid X_t)$.
   3. Receive reward $R_t(A_t)$.
   4. $\theta_{t+1} \leftarrow \theta_t + \eta R_t(A_t) \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$.

Using a baseline

Subtracting a Baseline from Reward

- Our objective function is
$$J(\theta) = \mathbb{E}_\theta [R(A)].$$
- Suppose we introduce a new reward vector $R_0 = R - b$, for constant b.
- Then
$$J_b(\theta) = \mathbb{E}_\theta (R_0(A)) = \mathbb{E}_\theta (R(A)) - b.$$
- 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:

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

- The updates are different, so we'll get different optimization paths.
- Which is the best b?
- One approach is to find a b that gives the best approximation of $\nabla_\theta J(\theta_t)$.
- First we'll show that the estimator is unbiased for any b.
- Then we'll think about good 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 distribution on a, parameterized by θ.
- Then $\mathbb{E}_{A \sim p_\theta(a)}[\nabla_\theta \log p_\theta(A)] = 0$.
- **Proof:** (for case that a is discrete, everything differentiable as needed)

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

Estimate with baseline is unbiased

- Since the score has expectation 0,

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

- 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?

- The control variate is $b\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$. We know it's expectation – it's 0. We hope it's correlated with the original estimator $R_t(A_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)$.
- We could also take the approach Let's start by pretending that θ is one-dimensional. Then according to our control variate work, the b that minimizes the variance is

$$b = \text{Corr}\left(R_t(A_t)\nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t), \nabla_\theta \log \pi_{\theta_t}(A_t \mid X_t)\right)$$

  The optimal

What to use for the baseline?

- We're summing random vectors of the form

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

- Each is an unbiased estimate of $\nabla_\theta J(\theta)$.
- We're trying to "reduce the variance."
- But what is the "variance"?
- First, note that this expression is generally a **vector**.
- So there is no scalar "variance" we can just try to optimize.
- So raise your eyebrows if you see a derivation of the b that gives "minimal variance."

Basic approach to the baseline

- The easiest thing to use for a baseline is

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

- I haven't seen a great justification for this choice. (I have seen very bad ones!)
- A challenge for the class: find a solid mathematical justification for this choice (or any better choice).
  - Google, whatever.

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!

Learning the baseline

- Learn function $\phi(x)$ to predict the reward for a given input x.
- Use $\phi(X_i)$ as the baseline for round i.
- We can learn ϕ at the same time as we learn our policy.
  - e.g. minimize $\big(R(A_i) - b_\phi(X_i)\big)^2$.
- This is an approach suggested in Sutton's book.[SB18, Sec 13.4].

Self-critical baseline

- Here's another clever way to set a baseline from [RMM+17]:
- Find (or approximate) the action that is optimal under our policy:

$$a^* \approx \arg\max_a \pi_{\theta_t}(a|X_t),$$

  and then use the reward $r(a^*)$ as a baseline for determing θ_{t+1}.
- Intuition is that, if the current action performs better than the action our policy says is best, then we should make the current action more likely.
- But if it performs worse than what our policy says is best, let's make it less likely.
- A reasonable idea and seems to performs well in practice (at least for sequence prediction).

"Optimal" baseline

- Notice that we're estimating a gradient, which is a vector.
- Let's allow a different baseline 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

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

- 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:

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

- 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 $\frac{dV_j}{db} = 0$:

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

- So our estimate the j'th entry, we should use the baseline b_j.
- We can try to estimate the expectations from the logs:

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

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

where

$$G_i^j = \left[\nabla_\theta \log \pi_{\theta_t}(A_i \mid X_i)\right]_j .$$

- Warning: I can't find this derivation in the literature. It's inspired by Berkeley's CS 285: Lecture 5, Slide 19, but their slide is quite vague on specifics. They don't even 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.

References

Resources

- Policy gradient for contextual bandits is a simplified version of the REINFORCE algorithm for the reinforcement learning setting.

References I

[RMM+17] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel, *Self-critical sequence training for image captioning*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 7 2017, p. nil.

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