Policy Gradient

David S. Rosenberg

NYU: CDS

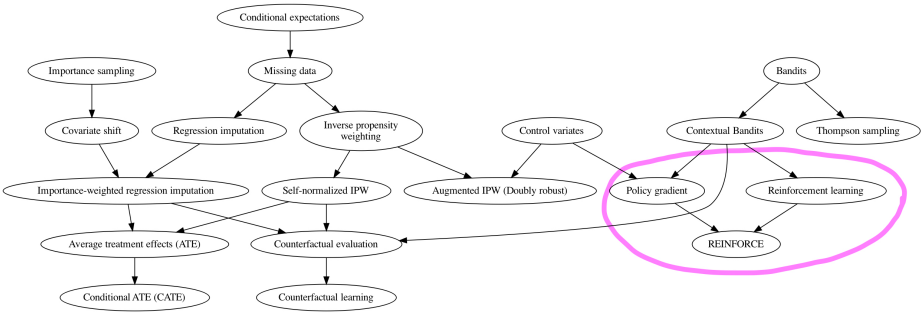
March 26, 2021

Contents

1. Recap of the bandit setting
2. Policy gradient methods
3. Policy gradient with baseline (i.e. control variate)
4. Experiments

Recap of the bandit setting

Where are we in the course?



- Counterfactual learning was about **offline** contextual bandits.
- But we still have some more to say about **online** contextual bandits.
- And about their generalization: reinforcement learning.

New type of policy:

- For multiarmed bandits, we discussed
  - ε-greedy
  - Thompson sampling
- In this module we discuss a new approach: policy gradient
- The method generalizes to contextual bandits and reinforcement learning, where it's called REINFORCE.

[Online] stochastic k-armed bandit

Stochastic k-armed bandit

1. Environment samples **reward vectors** for all rounds:

$$R_1, \ldots, R_T \text{ i.i.d. } P,$$

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**/arm $A_t \in \{1, \ldots, k\}$ based on history

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

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

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

Bandit policies

- Policies give some structure to action selection.
- A policy at round t
  - gives a conditional distribution over the action A_t to be taken
  - conditioned on the history $\mathcal{D}_t$.
- We'll denote the policy at round t as $\pi_t(\cdot \mid \mathcal{D}_t)$.
- Choosing an action according to policy π_t means we choose A_t **randomly** s.t.

$$\mathbb{P}(A_t = a) = \pi_t(a \mid \mathcal{D}_t).$$

Policy gradient methods

Policy gradient methods

- To learn a policy, we need a hypothesis space of policies.
- We create a "policy space" parameterized by θ:

$$\pi(a;\theta) = \frac{\exp(\theta(a))}{\sum_{a'=1}^{k} \exp(\theta(a'))} \quad a = 1, \ldots, k,$$

  where $\theta = (\theta(1), \ldots, \theta(k)) \in \mathbb{R}^k$.
- For any $\theta \in \mathbb{R}^k$, $\pi(a;\theta)$ gives a probability distribution over $a \in \{1, \ldots, k\}$.
- Larger $\theta(a)$ means action a is more likely.
- Note that for any $c \in \mathbb{R}$, $\pi(a;\theta + c) = \pi(a;\theta)$.

- You should recognize the $\pi(a;\theta)$ as the softmax function.
- Note that it only takes $k-1$ numbers to specify a probability distribution on k possibilities, since if we know $k-1$ of the probabilities, the last one is determined.
- So we shouldn't be surprised that $c \in \mathbb{R}, \pi(a;\theta+c) = \pi(a;\theta)$ – in other words, that one of the degrees of freedom we have in choosing θ has no effect on the distribution.

Policy gradient (rough idea)

Policy gradient method (rough idea):

1. Initialize $\theta_1 = 0$.
2. For each round $t = 1, \ldots, T$:
   1. Choose action A_t from distribution $\mathbb{P}(A_t = a) = \pi(a; \theta_t)$.
   2. Receive reward $R_t(A_t)$.
   3. $\theta_{t+1} \leftarrow \text{Update}(\theta_t, (A_t, R_t(A_t))$.

So our policy at round t is

$$\pi_t(a \mid \mathcal{D}_t) = \pi(a; \theta_t),$$

where the dependency on $\mathcal{D}_t$ is hidden in θ_t.

How to update the policy?

- We're trying to maximize expected rewards.
- Let A be an action chosen according to $\pi(a;\theta)$.
- Let R be a generic reward vector: $R \sim P$.
- We want to find θ to maximize

$$\begin{aligned}
\mathbb{E}[R(A)] &= \mathbb{E}[\mathbb{E}[R(A) \mid A]] \\
&= \sum_{a=1}^{k} \pi(a;\theta)\mathbb{E}[R(A) \mid A = a] \\
&= \sum_{a=1}^{k} \pi(a;\theta)\mathbb{E}[R(a) \mid A = a] \\
&= \sum_{a=1}^{k} \pi(a;\theta)\mathbb{E}[R(a)]\,.
\end{aligned}$$

Direct method / Greedy algorithm

- We can we estimate $\mathbb{E}[R(a)]$ for $a = 1, \ldots, k$.
- Then choose θ to put probability 1 on $\arg\max_a \mathbb{E}[R(a)]$.
- This brings us back to the greedy algorithm.
- In policy gradient, **slowly,** iteratively change θ to increase $\mathbb{E}[R(A)]$, $A \sim \pi(a; \theta)$.
- We can do this without explicitly estimating $\mathbb{E}[R(a)]$!
- Instead, we just estimate the gradient of $\mathbb{E}[R(A)]$ w.r.t. θ.

Differentiate the value

- Recall $\theta = (\theta(1), \ldots, \theta(k)) \in \mathbb{R}^k$.
- Let's find the partial w.r.t. each entry of θ. For $\alpha \in 1, \ldots, k$:

$$\begin{aligned}\frac{\partial}{\partial \theta(\alpha)} \mathbb{E}_\theta [R(A)] &= \frac{\partial}{\partial \theta(\alpha)} \left(\sum_{a=1}^{k} \mathbb{E}[R(a)] \pi(a; \theta) \right) \\ &= \sum_{a=1}^{k} \mathbb{E}[R(a)] \frac{\partial \pi(a; \theta)}{\partial \theta(\alpha)}\end{aligned}$$

- Do you remember the derivative of softmax? (Exercise)
- By the quotient rule and some algebra,

$$\begin{aligned}\frac{\partial \pi(a; \theta)}{\partial \theta(\alpha)} &= \frac{\partial}{\partial \theta(\alpha)} \left[\frac{\exp(\theta(a))}{\sum_{a'=1}^{k} \exp(\theta(a'))} \right] \\ &= \pi(a; \theta) (\mathbb{1}[a = \alpha] - \pi(\alpha; \theta))\end{aligned}$$

Differentiate the value

- Let's find the partial w.r.t. each entry of θ:

$$\begin{aligned}
\frac{\partial}{\partial \theta(\alpha)} \mathbb{E}_\theta [R(A)] &= \sum_{a=1}^{k} \mathbb{E}[R(a)] \pi(a;\theta) (\mathbb{1}[a=\alpha] - \pi(\alpha;\theta)) \\
&= \sum_{a=1}^{k} \pi(a;\theta) \mathbb{E}[R(a) (\mathbb{1}[a=\alpha] - \pi(\alpha;\theta))] \\
&= \mathbb{E}_{A\sim\pi(\cdot;\theta)} [R(A) (\mathbb{1}[A=\alpha] - \pi(\alpha;\theta))]
\end{aligned}$$

- For SGD, we just need an unbiased estimate of the gradient.
- What can we use?
- Hint: The answer is on this slide.

- Not that inside the sum, $\mathbb{1}[a = \alpha] - \pi(\alpha; \theta)$ is just a deterministic scalar. So we can move it inside the expectation.
- We could consider estimating

$$\mathbb{E}_{A \sim \pi(\cdot;\theta)} \left[R(A) \left(\mathbb{1}[A = \alpha] - \pi(\alpha; \theta) \right) \right]$$

  using the full logged data.
- We'd need importance sampling, since the action distribution $\pi(\cdot; \theta_t)$ changes in every round.
- However, as we do this in consecutive rounds, our gradient estimates will be correlated – in particular, their errors will be correlated — this might be an issue?which could be an issue.
- We're going to take a simpler, faster approach.

What do we actually want?

- Suppose we're at round t. So our current $\theta = \theta_t$.
- We choose our action $A_t \sim \pi(a; \theta_t)$ and we observe $R_t(A_t)$.
- So $R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ is an unbiased estimate of our partial derivative:

$$\begin{aligned}
&\mathbb{E}\left[R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)\right] \\
&= \mathbb{E}_{A \sim \pi(\cdot; \theta_t)}\left[R(A)\left(\mathbb{1}\left[A = \alpha\right] - \pi(\alpha; \theta_t)\right)\right] \\
&= \frac{\partial}{\partial \theta(\alpha)} \mathbb{E}_{\theta_t}\left[R(A)\right].
\end{aligned}$$

- At each round, use $R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ as a step direction for updating θ_t.
- The only tuning parameter is the step size at each time step.

- We could consider estimating

$$\mathbb{E}_{A\sim\pi(\cdot;\theta)}\left[R(A)\left(\mathbb{1}\left[A=\alpha\right]-\pi(\alpha;\theta)\right)\right]$$

  using the all the bandit logged feedback.
- We'd need importance sampling, since the action distribution $\pi(\cdot;\theta_t)$ changes in every round.
- However, as we do this in consecutive rounds, our gradient estimates will be correlated – in particular, their errors will be correlated — this might be an issue?

Policy gradient algorithm

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. Choose action A_t from distribution $\mathbb{P}(A_t = a) = \pi(a; \theta_t)$.
   2. Receive reward $R_t(A_t)$.
   3. $\theta_{t+1}(\alpha) \leftarrow \theta_t(\alpha) + \eta R_t(A_t) \left(\mathbb{1}[A_t = \alpha] - \pi(\alpha; \theta_t)\right)$ for $\alpha = 1, \ldots, k$.

- Note the interaction between the step size and the scale of the rewards.
- We can give some additional interpretation to what's going on here...

First interpretation of policy gradient

- Update: $\theta_{t+1}(\alpha) \leftarrow \eta R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ for $\alpha = 1, \ldots, k$.
- Suppose $R_t(A_t) > 0$.
- For $\alpha = A_t$, $\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t) > 0$, so $\theta_{t+1}(A_t)$ gets larger.
- For $\alpha \neq A_t$, $\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t) < 0$, so $\theta_{t+1}(\alpha)$ gets smaller.
- No matter what the reward is, the action played always gets more likely.
- Does this make sense? Making an action more likely even if the reward is poor?
- Note that the size of the change is controlled by $R_t(A_t)$.
- So the larger $R_t(A_t)$ is, the larger the increase in the probability of the action.

- Thought experiment: what if the reward in round 1 is very large. This will make the probability of playing A_1 in round 2 very large. Which means we'll very likely play A_1 in round 2 as well. Which will make A_1 even more likely to be played in round 3 (as long as the reward for A_1 in round 2 is > 0. You can see how we can get stuck only ever playing A_1. This is a cautionary tale to be careful about our step sizes, or in general of taking steps that are too large.

Policy gradient with baseline (i.e. control variate)

Step direction is unbiased, what about variance?

- We have an unbiased step direction $R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$.
- It's unbiased for the true partial derivative $\frac{\partial}{\partial \theta(\alpha)} \mathbb{E}_\theta\left[R(A)\right]$ for each $\alpha \in 1, \ldots, k$.
- What about the variance?
- Randomness is coming in two places:
  - from the reward vector $R_t \sim P$ and
  - from the component of R_t that we select, which is determined by A_t.
- Perhaps we can reduce the variance using a control variate?

Control variate for policy gradient

- We're looking for something correlated with

$$R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right),$$

  but with known expectation.
- Can we reduce some of the randomness due to the choice of A_t?
- Consider using $\beta_t\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ as our control variate, for some $\beta_t \in \mathbb{R}$.
- Note that

$$\mathbb{E}_{\theta_t}\left[\beta_t\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)\right] = \beta_t\left[\mathbb{P}\left(A_t = \alpha\right) - \pi(\alpha, \theta_t)\right] = 0.$$

- Since it shares the term $\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$, we think / hope that it will be correlated.
- And it has known expectation.
- So $\beta_t\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ is a viable candidate for a control variate.

- Trivial but important question: if we want something correlated with $R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$, and the more correlated the better, why aren't we just using the thing itself: $R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$? It would be great if we could use that as a control variate. Because that would mean we know what its expectation is. And if we knew the expectation $\mathbb{E}\left[R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)\right]$, we'd be in great shape, because that's exactly the gradient we are trying to estimate. Of course, we can't do this because we don't know that expectation. We have to be satisfied with something that isn't as correlated, but for which we know the expectation (in this case, it's zero).
- What if we fit an expected rewards estimator $\hat{r}(a) \approx \mathbb{E}\left[R(a)\right]$? Seems like $\hat{r}(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ would be a better control variate. We could compute the expectation directly with

$$\sum_{a=1}^{k} \pi(a; \theta_t)\hat{r}(a)\left(\mathbb{1}\left[a = \alpha\right] - \pi(\alpha; \theta_t)\right).$$

  Seems worth a try (project idea?).

Gradient estimate with baseline

- Using the control variate, our new unbiased estimate for $\frac{\partial}{\partial \theta_t(\alpha)} \mathbb{E}_{\theta_t}[R(A)]$ is

$$(R_t(A_t) - \beta_t)(\mathbb{1}[A_t = \alpha] - \pi(\alpha; \theta_t)).$$

- β_t is called the **baseline**, and it amounts to an additive shift on rewards.
- A typical, easy choice for β_t is $\beta_t = \frac{1}{t}\sum_{i=1}^{t} R_t(A_t)$.
- So update is $\theta_{t+1}(\alpha) \leftarrow \eta(R_t(A_t) - \beta_t)(\mathbb{1}[A_t = \alpha] - \pi(\alpha; \theta_t))$.
- Action A_t gets more likely when $R_t(A_t) > \beta_t$, everything else less likely.
- Action A_t gets less likely when $R_t(A_t) < \beta_t$, everything else more likely.
- The extent of the change is determined by how far $R_t(A_t)$ is from β_t.

Choosing the "optimal" β_t

- From our study of control variates, if we let
  - $Y_\alpha = R_t(A_t)\left(\mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)\right)$ and
  - $f_\alpha(A_t) = \mathbb{1}\left[A_t = \alpha\right] - \pi(\alpha; \theta_t)$,
- then the β_t^α minimizing the variance of our estimate of $\mathbb{E}Y_\alpha$ is

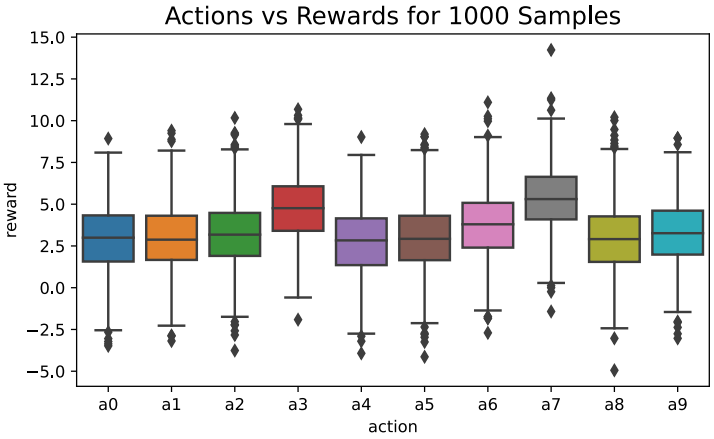
$$\beta_t^\alpha = \rho_\alpha \frac{\mathrm{SD}(Y_\alpha)}{\mathrm{SD}(f_\alpha(A_t)}},$$

  where $\rho_\alpha = \mathrm{Corr}(Y_\alpha, f_\alpha(A_t))$.
- We can estimate these β_t^α from historical rounds of play.
- Note that each α will have a different β_t^α.

- Choosing β_t's in this way is an attempt to minimize the variance of our estimates of each entry of the gradient $\frac{\partial}{\partial \alpha}\mathbb{E}[R(A)]$.
- However, these estimates will be correlated. Since we're really estimating the whole gradient vector, might it make sense to consider the full covariance matrix for our estimated gradient? Is there a better way to choose the β_t^{α}'s jointly, with this perspective? (Possible project idea.)

Experiments

Working example: 10-armed bandit



Plot and simulation code courtesy of Ryan Carroll.

Working example: 10-armed bandit

- The reward distribution is given by

$$R_i(a) \sim \mathcal{N}(q_*(a), \sigma = 2),$$

  for each action, where $q_*(1), \ldots, q_*(k)$ are **unknown parameters**.
- We previously tried Thompson sampling with various prior distributions.
  - NOTE: Thompson sampling uses knowledge about reward distribution, including that $\sigma = 2$.
- Now we'll try policy gradient with various step sizes η and with/without a baseline.
  - Policy gradient doesn't assume anything about the reward distribution.
  - For baseline at round t, we'll use $\beta_t = \frac{1}{t} \sum_{i=1}^{t} R_i(A_i)$.

Policy gradient results

- Learning rate: η Baseline $\beta_t = \frac{1}{t}\sum_{i=1}^{t} R_i(A_i)$

strategy	mean	SD	SE
Policy gradient $\eta = 0.025$, with baseline	4.833	0.153	0.011
Policy gradient $\eta = 0.025$, no baseline	4.745	0.249	0.018
Policy gradient $\eta = 0.05$, with baseline	5.054	0.156	0.011
Policy gradient $\eta = 0.05$, no baseline	4.882	0.375	0.026
Policy gradient $\eta = 0.1$, with baseline	**5.121**	0.231	0.016
Policy gradient $\eta = 0.1$, no baseline	4.917	0.442	0.031
Policy gradient $\eta = 0.4$, with baseline	4.991	0.499	0.035
Policy gradient $\eta = 0.4$, no baseline	4.267	0.934	0.066

- For each strategy, we ran 200 trials of 1000 steps each.
- For each trial, we compute the mean reward across 1000 rounds.
- "Mean" and "SD" in the table are the mean and SD of these mean rewards across the 200 trials. SE is the SE of the "Mean" in the table.
- We see that the baseline help substantially, and that tuning the learning rate can make a large difference as well.

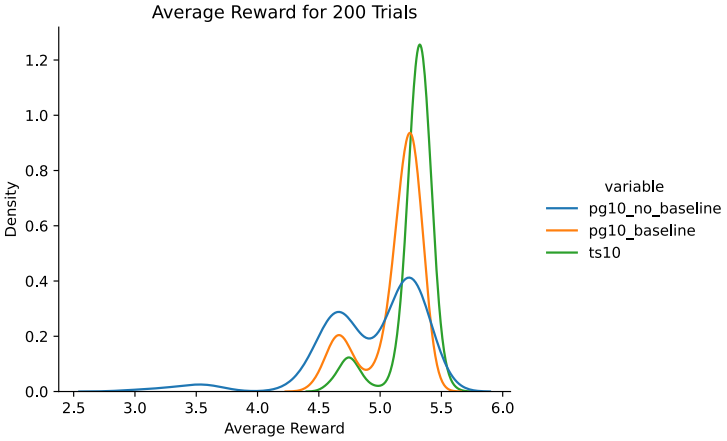
Policy gradient vs Thompson sampling

strategy	mean	SD	SE
Policy gradient $\eta = 0.1$, with baseline	5.121	0.231	0.016
Thompson sampling $\sigma_0 = 2$	5.129	0.306	0.022
Thompson sampling $\sigma_0 = 5$	5.229	0.214	0.015
Thompson sampling $\sigma_0 = 10$	5.279	0.169	0.012

- Thompson sampling beats our best policy gradient setting.
- Thompson sampling was able to leverage more information about the rewards distribution.

- For each strategy, we ran 200 trials of 1000 steps each.
- For each trial, we compute the mean reward across 1000 rounds.
- "Mean" and "SD" in the table are the mean and SD of these mean rewards across the 200 trials. SE is the SE of the "Mean" in the table.
- We see that the baseline help substantially, and that tuning the learning rate can make a large difference as well.

Policy mean reward densities



Plot and simulation code courtesy of Ryan Carroll.

References

Resources

- Sutton and Barto's book on reinforcement learning discusses policy gradient for multiarmed bandits [SB18, Sec 2.8].

References I

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