CS 747, Autumn 2022: Lecture 15

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

Autumn 2022

Reinforcement Learning

1. Least-squares and maximum likelihood estimators
2. TD(0) algorithm
3. Convergence of batch TD(0)

Reinforcement Learning

1. Least-squares and maximum likelihood estimators
2. TD(0) algorithm
3. Convergence of batch TD(0)

Estimate p

- You have two coins. You are told that the probability of a head (1-reward) for Coin 1 is $p \in [0, 0.5]$, and that for Coin 2 is $2p$.

Coin 1

$\mathbb{P}\{\text{heads}\} = p$

Coin 2

$\mathbb{P}\{\text{heads}\} = 2p$

Estimate p

- You have two coins. You are told that the probability of a head (1-reward) for Coin 1 is $p \in [0, 0.5]$, and that for Coin 2 is $2p$.
- Hence the corresponding probabilities of a tail (0-reward) are $1 - p$ and $1 - 2p$, respectively.

Coin 1

$\mathbb{P}\{\text{heads}\} = p$

Coin 2

$\mathbb{P}\{\text{heads}\} = 2p$

Estimate p

- You have two coins. You are told that the probability of a head (1-reward) for Coin 1 is $p \in [0, 0.5]$, and that for Coin 2 is $2p$.
- Hence the corresponding probabilities of a tail (0-reward) are $1 - p$ and $1 - 2p$, respectively.
- You toss each coin once and see these outcomes.

Coin 1	Coin 2
	
$\mathbb{P}\{\text{heads}\} = p$	$\mathbb{P}\{\text{heads}\} = 2p$
Outcome = 1	Outcome = 0

Estimate p

- You have two coins. You are told that the probability of a head (1-reward) for Coin 1 is $p \in [0, 0.5]$, and that for Coin 2 is $2p$.
- Hence the corresponding probabilities of a tail (0-reward) are $1 - p$ and $1 - 2p$, respectively.
- You toss each coin once and see these outcomes.

Coin 1	Coin 2
	
$\mathbb{P}\{\text{heads}\} = p$	$\mathbb{P}\{\text{heads}\} = 2p$
Outcome = 1	Outcome = 0

What is your estimate of p (call it $\hat{p}$)?

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,

$$SE(q) = (q - 1)^2 + (2q - 0)^2.$$

$$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0, 0.5]} SE(q) = 0.2.$$

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,

$$SE(q) = (q - 1)^2 + (2q - 0)^2.$$

$$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0,0.5]} SE(q) = 0.2.$$

- **Maximum likelihood estimate.**
  For $q \in [0, 0.5]$,

$$L(q) = q(1 - 2q).$$

$$\hat{p}_{ML} \stackrel{\text{def}}{=} \operatorname*{argmax}_{q \in [0,0.5]} L(q) = 0.25.$$

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,

$$SE(q) = (q - 1)^2 + (2q - 0)^2.$$

$$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0,0.5]} SE(q) = 0.2.$$

- **Maximum likelihood estimate.**
  For $q \in [0, 0.5]$,

$$L(q) = q(1 - 2q).$$

$$\hat{p}_{ML} \stackrel{\text{def}}{=} \operatorname*{argmax}_{q \in [0,0.5]} L(q) = 0.25.$$

- Which estimate is "correct"?

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,

$$SE(q) = (q - 1)^2 + (2q - 0)^2.$$

$$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0,0.5]} SE(q) = 0.2.$$

- **Maximum likelihood estimate.**
  For $q \in [0, 0.5]$,

$$L(q) = q(1 - 2q).$$

$$\hat{p}_{ML} \stackrel{\text{def}}{=} \operatorname*{argmax}_{q \in [0,0.5]} L(q) = 0.25.$$

- Which estimate is “correct”? Neither!

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,

$$SE(q) = (q - 1)^2 + (2q - 0)^2.$$

$$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0, 0.5]} SE(q) = 0.2.$$

- **Maximum likelihood estimate.**
  For $q \in [0, 0.5]$,

$$L(q) = q(1 - 2q).$$

$$\hat{p}_{ML} \stackrel{\text{def}}{=} \operatorname*{argmax}_{q \in [0, 0.5]} L(q) = 0.25.$$

- Which estimate is "correct"? Neither!
- Which estimate is more useful?

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,
  $$SE(q) = (q - 1)^2 + (2q - 0)^2.$$
  $$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0,0.5]} SE(q) = 0.2.$$
- **Maximum likelihood estimate.**
  For $q \in [0, 0.5]$,
  $$L(q) = q(1 - 2q).$$
  $$\hat{p}_{ML} \stackrel{\text{def}}{=} \operatorname*{argmax}_{q \in [0,0.5]} L(q) = 0.25.$$
- Which estimate is “correct”? Neither!
- Which estimate is more useful? Depends on the use!

Two Common Estimates

- **Least-squares estimate.**
  For $q \in [0, 0.5]$,
  $$SE(q) = (q-1)^2 + (2q-0)^2.$$
  $$\hat{p}_{LS} \stackrel{\text{def}}{=} \operatorname*{argmin}_{q \in [0,0.5]} SE(q) = 0.2.$$
- **Maximum likelihood estimate.**
  For $q \in [0, 0.5]$,
  $$L(q) = q(1-2q).$$
  $$\hat{p}_{ML} \stackrel{\text{def}}{=} \operatorname*{argmax}_{q \in [0,0.5]} L(q) = 0.25.$$
- Which estimate is "correct"? Neither!
- Which estimate is more useful? Depends on the use!
- Note that there are other estimates, too.

Reinforcement Learning

1. Least-squares and maximum likelihood estimators
2. TD(0) algorithm
3. Convergence of batch TD(0)

Bootstrapping

- Suppose $\hat{V}^t$ is our current estimate of state-values.

Bootstrapping

- Suppose $\hat{V}^t$ is our current estimate of state-values.
- Say we generate this episode.

$$\boxed{s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top.}$$

Bootstrapping

- Suppose $\hat{V}^t$ is our current estimate of state-values.
- Say we generate this episode.

$$\boxed{s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top.}$$

- At what point of time can we update our estimate $\hat{V}^t(s_2)$?

Bootstrapping

- Suppose $\hat{V}^t$ is our current estimate of state-values.
- Say we generate this episode.

$$\boxed{s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top.}$$

- At what point of time can we update our estimate $\hat{V}^t(s_2)$?

- With MC methods, we would wait for $s_\top$, and then update $\hat{V}^{t+1}(s_2) \leftarrow \hat{V}^t(s_2)(1 - \alpha_{t+1}) + \alpha_{t+1}M$, where $M = 2 + \gamma \cdot 1 + \gamma^2 \cdot 1 + \gamma^3 \cdot 2 + \gamma^4 \cdot 1$.

Bootstrapping

- Suppose $\hat{V}^t$ is our current estimate of state-values.
- Say we generate this episode.

$$\boxed{s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top.}$$

- At what point of time can we update our estimate $\hat{V}^t(s_2)$?
- With MC methods, we would wait for $s_\top$, and then update $\hat{V}^{t+1}(s_2) \leftarrow \hat{V}^t(s_2)(1 - \alpha_{t+1}) + \alpha_{t+1} M$, where $M = 2 + \gamma \cdot 1 + \gamma^2 \cdot 1 + \gamma^3 \cdot 2 + \gamma^4 \cdot 1$.
- Instead, how about this update as soon as we see s_3? $\hat{V}^{t+1}(s_2) \leftarrow \hat{V}^t(s_2)(1 - \alpha_{t+1}) + \alpha_{t+1} B$, where $B = 2 + \gamma \hat{V}^t(s_3)$.

Bootstrapping

- Suppose $\hat{V}^t$ is our current estimate of state-values.
- Say we generate this episode.

$$\boxed{s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top.}$$

- At what point of time can we update our estimate $\hat{V}^t(s_2)$?
- With MC methods, we would wait for $s_\top$, and then update
$\hat{V}^{t+1}(s_2) \leftarrow \hat{V}^t(s_2)(1 - \alpha_{t+1}) + \alpha_{t+1} M$, where
$M = 2 + \gamma \cdot 1 + \gamma^2 \cdot 1 + \gamma^3 \cdot 2 + \gamma^4 \cdot 1$. Monte Carlo estimate.
- Instead, how about this update as soon as we see s_3?
$\hat{V}^{t+1}(s_2) \leftarrow \hat{V}^t(s_2)(1 - \alpha_{t+1}) + \alpha_{t+1} B$, where
$B = 2 + \gamma \hat{V}^t(s_3)$. Bootstrapped estimate.

Temporal Difference Learning: TD(0)

Assume policy to be evaluated is π.
Initialise $\hat{V}^0$ arbitrarily.
Assume that the agent is born in state s^0.

For $t = 0, 1, 2, \ldots$:
  Take action $a^t \sim \pi(s^t)$.
  Obtain reward r^t, next state s^{t+1}.
  $\hat{V}^{t+1}(s^t) \leftarrow \hat{V}^t(s^t) + \alpha_{t+1}\{r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)\}$.
  For $s \in S \setminus \{s^t\}$: $\hat{V}^{t+1}(s) \leftarrow \hat{V}^t(s)$. //Often left implicit.

Temporal Difference Learning: TD(0)

Assume policy to be evaluated is π.
Initialise $\hat{V}^0$ arbitrarily.
Assume that the agent is born in state s^0.

For $t = 0, 1, 2, \ldots$:
 Take action $a^t \sim \pi(s^t)$.
 Obtain reward r^t, next state s^{t+1}.
 $\hat{V}^{t+1}(s^t) \leftarrow \hat{V}^t(s^t) + \alpha_{t+1}\{r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)\}$.
 For $s \in S \setminus \{s^t\}$: $\hat{V}^{t+1}(s) \leftarrow \hat{V}^t(s)$. //Often left implicit.

- $\hat{V}^t(s^t)$: current estimate; $r^t + \gamma \hat{V}^t(s^{t+1})$: new estimate.

Temporal Difference Learning: TD(0)

Assume policy to be evaluated is π.
Initialise $\hat{V}^0$ arbitrarily.
Assume that the agent is born in state s^0.

For $t = 0, 1, 2, \ldots$:
    Take action $a^t \sim \pi(s^t)$.
    Obtain reward r^t, next state s^{t+1}.
    $\hat{V}^{t+1}(s^t) \leftarrow \hat{V}^t(s^t) + \alpha_{t+1}\{r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)\}$.
    For $s \in S \setminus \{s^t\}$: $\hat{V}^{t+1}(s) \leftarrow \hat{V}^t(s)$. //Often left implicit.

- $\hat{V}^t(s^t)$: current estimate; $r^t + \gamma \hat{V}^t(s^{t+1})$: new estimate.
- $r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)$: temporal difference prediction error.

Temporal Difference Learning: TD(0)

Assume policy to be evaluated is π.
Initialise $\hat{V}^0$ arbitrarily.
Assume that the agent is born in state s^0.

For $t = 0, 1, 2, \ldots$:
    Take action $a^t \sim \pi(s^t)$.
    Obtain reward r^t, next state s^{t+1}.
    $\hat{V}^{t+1}(s^t) \leftarrow \hat{V}^t(s^t) + \alpha_{t+1}\{r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)\}$.
    For $s \in S \setminus \{s^t\}$: $\hat{V}^{t+1}(s) \leftarrow \hat{V}^t(s)$. //Often left implicit.

- $\hat{V}^t(s^t)$: current estimate; $r^t + \gamma \hat{V}^t(s^{t+1})$: new estimate.
- $r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)$: temporal difference prediction error.
- α_{t+1}: learning rate.

Temporal Difference Learning: TD(0)

Assume policy to be evaluated is π.
Initialise $\hat{V}^0$ arbitrarily.
Assume that the agent is born in state s^0.

For $t = 0, 1, 2, \ldots$:
    Take action $a^t \sim \pi(s^t)$.
    Obtain reward r^t, next state s^{t+1}.
    $\hat{V}^{t+1}(s^t) \leftarrow \hat{V}^t(s^t) + \alpha_{t+1}\{r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)\}$.
    For $s \in S \setminus \{s^t\}$: $\hat{V}^{t+1}(s) \leftarrow \hat{V}^t(s)$. //Often left implicit.

- $\hat{V}^t(s^t)$: current estimate; $r^t + \gamma \hat{V}^t(s^{t+1})$: new estimate.
- $r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)$: temporal difference prediction error.
- α_{t+1}: learning rate.
- Under standard conditions, $\lim_{t\to\infty} \hat{V}^t = V^\pi$.

Temporal Difference Learning: TD(0)

Assume policy to be evaluated is π.
Initialise $\hat{V}^0$ arbitrarily.
Assume that the agent is born in state s^0.

For $t = 0, 1, 2, \ldots$:
  Take action $a^t \sim \pi(s^t)$.
  Obtain reward r^t, next state s^{t+1}.
  $\hat{V}^{t+1}(s^t) \leftarrow \hat{V}^t(s^t) + \alpha_{t+1}\{r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)\}$.
  For $s \in S \setminus \{s^t\}$: $\hat{V}^{t+1}(s) \leftarrow \hat{V}^t(s)$. //Often left implicit.

- $\hat{V}^t(s^t)$: current estimate; $r^t + \gamma \hat{V}^t(s^{t+1})$: new estimate.
- $r^t + \gamma \hat{V}^t(s^{t+1}) - \hat{V}^t(s^t)$: temporal difference prediction error.
- α_{t+1}: learning rate.
- Under standard conditions, $\lim_{t \to \infty} \hat{V}^t = V^\pi$. How to run on episodic tasks?

Reinforcement Learning

1. Least-squares and maximum likelihood estimators
2. TD(0) algorithm
3. Convergence of batch TD(0)

First-visit MC Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Recall that for $s \in S$,

$$\hat{V}^N_{\text{First-visit}}(s) = \frac{\sum_{i=1}^{N} G(s, i, 1)}{\sum_{i=1}^{N} \mathbf{1}(s, i, 1)}.$$

First-visit MC Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Recall that for $s \in S$,

$$\hat{V}^N_{\text{First-visit}}(s) = \frac{\sum_{i=1}^N G(s, i, 1)}{\sum_{i=1}^N \mathbf{1}(s, i, 1)}.$$

- For $s \in S$, $V : S \to \mathbb{R}$, define

$$Error_{\text{First}}(V, s) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \mathbf{1}(s, i, 1)\,(V(s) - G(s, i, 1))^2\,.$$

First-visit MC Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Recall that for $s \in S$,

$$\hat{V}^N_{\text{First-visit}}(s) = \frac{\sum_{i=1}^{N} G(s, i, 1)}{\sum_{i=1}^{N} \mathbf{1}(s, i, 1)}.$$

- For $s \in S$, $V : S \to \mathbb{R}$, define

$$Error_{\text{First}}(V, s) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \mathbf{1}(s, i, 1)\,(V(s) - G(s, i, 1))^2.$$

- Observe that for $s \in S$, $\hat{V}^N_{\text{First-visit}}(s) = \text{argmin}_V\, Error_{\text{First}}(V, s)$.

Every-visit MC Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Recall that for $s \in S$,

$$\hat{V}^N_{\text{Every-visit}}(s) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{\infty} G(s, i, j)}{\sum_{i=1}^{N} \sum_{j=1}^{\infty} \mathbf{1}(s, i, j)}.$$

Every-visit MC Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Recall that for $s \in S$,

$$\hat{V}^N_{\text{Every-visit}}(s) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{\infty} G(s, i, j)}{\sum_{i=1}^{N} \sum_{j=1}^{\infty} \mathbf{1}(s, i, j)}.$$

- For $s \in S$, $V : S \to \mathbb{R}$, define

$$Error_{\text{Every}}(V, s) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \sum_{j=1}^{\infty} \mathbf{1}(s, i, j) \left(V(s) - G(s, i, j)\right)^2.$$

Every-visit MC Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Recall that for $s \in S$,

$$\hat{V}^N_{\text{Every-visit}}(s) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{\infty} G(s, i, j)}{\sum_{i=1}^{N} \sum_{j=1}^{\infty} \mathbf{1}(s, i, j)}.$$

- For $s \in S$, $V : S \to \mathbb{R}$, define

$$Error_{\text{Every}}(V, s) \stackrel{\text{def}}{=} \sum_{i=1}^{N} \sum_{j=1}^{\infty} \mathbf{1}(s, i, j) \left(V(s) - G(s, i, j)\right)^2.$$

- Observe for $s \in S$, $\hat{V}^N_{\text{Every-visit}}(s) = \text{argmin}_V \, Error_{\text{Every}}(V, s)$.

Batch TD(0) Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- After any finite N episodes, the estimate of $TD(0)$ will depend on the initial estimate V^0.
- To "forget" V^0, run the N collected episodes over and over again, and make TD(0) updates.

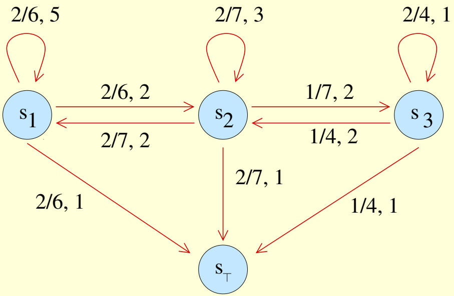
Batch TD(0) Estimate

Episode 1 Episode 2 Episode 3 Episode 4 Episode 5 Episode 6 (= Episode 1) Episode 7 (= Episode 2) Episode 8 (= Episode 3) Episode 9 (= Episode 4) Episode 10 (= Episode 5) Episode 11 (= Episode 1) Episode 12 (= Episode 2) $\vdots$

- Anneal the learning rate as usual ($\alpha_t = \frac{1}{t}$).
- $\lim_{t \to \infty} V^t$ will not depend on $\hat{V}^0$.
- It only depends on N episodes of real data.
- Refer to $\lim_{t \to \infty} \hat{V}^t$ as $\hat{V}^N_{\text{Batch-TD(0)}}$.
- Can we conclude something relevant about $\hat{V}^N_{\text{Batch-TD(0)}}$?

Batch TD(0) Estimate

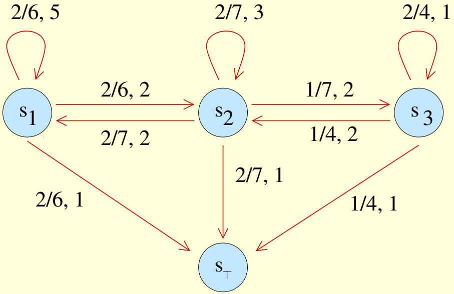
Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.



- Let M_{MLE} be the MDP $(S, A, \hat{T}, \hat{R}, \gamma)$ with the highest likelihood of generating this data (true T, R unknown).

Batch TD(0) Estimate

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.



- Let M_{MLE} be the MDP $(S, A, \hat{T}, \hat{R}, \gamma)$ with the highest likelihood of generating this data (true T, R unknown).
- $\hat{V}^N_{\text{Batch-TD(0)}}$ is the same as V^π on M_{MLE}!

Comparison

- Data.

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Estimates.

	s_1	s_2	s_3
$\hat{V}^T_{\text{First-visit}}$	7.33	6.25	3
$\hat{V}^T_{\text{Every-visit}}$	5.83	4.29	3.25
$\hat{V}^T_{\text{Batch-TD(0)}}$	7.5	7	6

Comparison

- Data.

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Estimates.

	s_1	s_2	s_3
$\hat{V}^T_{\text{First-visit}}$	7.33	6.25	3
$\hat{V}^T_{\text{Every-visit}}$	5.83	4.29	3.25
$\hat{V}^T_{\text{Batch-TD(0)}}$	7.5	7	6

- Which estimate is "correct"? Which is more useful?
- Is it recommended to bootstrap or not?

Comparison

- Data.

Episode 1: $s_1, 5, s_1, 2, s_2, 3, s_2, 1, s_\top$.
Episode 2: $s_2, 2, s_3, 1, s_3, 1, s_3, 2, s_2, 1, s_\top$.
Episode 3: $s_1, 2, s_2, 2, s_1, 5, s_1, 1, s_\top$.
Episode 4: $s_3, 1, s_\top$.
Episode 5: $s_2, 3, s_2, 2, s_1, 1, s_\top$.

- Estimates.

	s_1	s_2	s_3
$\hat{V}^T_{\text{First-visit}}$	7.33	6.25	3
$\hat{V}^T_{\text{Every-visit}}$	5.83	4.29	3.25
$\hat{V}^T_{\text{Batch-TD(0)}}$	7.5	7	6

- Which estimate is "correct"? Which is more useful?
- Is it recommended to bootstrap or not?
- Usually a "middle path" works best. Coming up next week!