CMPUT 365: Temporal Difference Methods for Prediction

Rupam Mahmood

Feb 28, 2022

Admin

Due dates C2M2:

- Practice quiz: Tue Mar 1
- Graded notebook: Sat Mar 5

Midterm:

- Marks and feedback are on eclass and assign2

Review of MC

- MC methods come with a huge advantage: model-free estimation of value functions
- Disadvantage is that they make per-episode updates, not every time step
- They are computationally and memory-wise expensive and cumbersome

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

  $G \leftarrow 0$

  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

    $G \leftarrow \gamma G + R_{t+1}$

    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:

      Append G to $Returns(S_t)$

      $V(S_t) \leftarrow \text{average}(Returns(S_t))$

Prediction as estimating value functions

- Predictions are building blocks for many control methods
- The usefulness of predictions goes beyond control
- Forming a predictive question: How many times will you get honked at today?
- (Pseudo-) reward: +1 for each honk
- Termination of episode: end of the day
- Behavior: the way you drive (think of your average speed, frequency of changing lanes, etc.)
- Can be answered by estimating $v_\pi(s) \doteq E_\pi[G_t \mid S_t = s]$

Much of prediction is about estimating expected values

$$E[X]$$

compute exactly if you know the model

$$= \sum_x x P(X = x)$$

Estimate from samples

$$\approx \frac{1}{n} \sum_{k=1}^{n} X_k$$

Much of prediction is about estimating expected values (cont'd)

$$v_\pi(s) \doteq E_\pi[G_t \mid S_t = s]$$

Dynamic programming

E.g., Iterative policy evaluation

Sample-based

Monte Carlo (MC)

From MC to TD(0)

Monte Carlo estimator for on-policy prediction: $V(s) \doteq \boxed{\dfrac{\sum_{t \in \mathcal{T}(s)} G_t}{|\mathcal{T}(s)|}}$ → sample average.

Incremental Monte Carlo estimator: $V(S_t) \leftarrow V(S_t) + \dfrac{1}{N(S_t)}\left[G_t - V(S_t)\right]$

→ The target (the return is unbiased from the true value)

Constant-α MC: $V(S_t) \leftarrow V(S_t) + \alpha \boxed{\left[G_t - V(S_t)\right]}$ The most important MC update !!

↳ the MC error (only update when episode ends).

TD(0): $V(S_t) \leftarrow V(S_t) + \alpha \left[\boxed{R_{t+1} + \gamma \underline{V(S_{t+1})}} - V(S_t)\right]$ => TD update.

Value estimate
(not the true value since we don't know it!!)

this is slightly incorrect, 因为 $V(S_{t+1})$ 不是 true value.
∴ G can't be unbiased !!

Parham Golestaneh Talaei对所有人说 下午1:48

PG V(S') is the value of the next state that was computed during the last trajectory right? Just like the in place policy iteration that we use the previous state values => Yes!!

Unlike Monte Carlo, TD(0) works online

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated
Initialize:
  $V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$
  $Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):
  Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$
  $G \leftarrow 0$
  Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
    $G \leftarrow \gamma G + R_{t+1}$
    Unless S_t appears in $S_0, S_1, \ldots, S_{t-1}$:
      Append G to $Returns(S_t)$
      $V(S_t) \leftarrow \text{average}(Returns(S_t))$

How would you characterize the differences between MC and TD(0) algorithmically?

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
  Initialize S
  Loop for each step of episode:
    $A \leftarrow$ action given by π for S
    Take action A, observe R, S'
    $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$
    $S \leftarrow S'$
  until S is terminal

Unlike Monte Carlo, TD(0) works online (cont'd)

Tabular TD(0) for estimating v_π

Input: the policy π to be evaluated
Algorithm parameter: step size $\alpha \in (0, 1]$
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop for each episode:
    Initialize S
    Loop for each step of episode:
        $A \leftarrow$ action given by π for S
        Take action A, observe R, S'
        $V(S) \leftarrow V(S) + \alpha\big[R + \gamma V(S') - V(S)\big]$ doing learning (more calculation during the episode than MC).
        $S \leftarrow S'$
    until S is terminal

Aaron Skiba对所有人说 下午1:50
AS So 'offline' just means not learning as it's computing? => Yes.

Say an oracle gives us return G from future at each step. Replace $R + \gamma V(S')$ with G. This is an online but acausal Monte Carlo method. Will it be first-visit or every-visit?

Just like MC, TD(0) estimates the value function but differently

TD(0): $V(S_t) \leftarrow V(S_t) + \alpha \boxed{[\underbrace{R_{t+1} + \gamma V(S_{t+1})}_{\text{the TD target.}} - V(S_t)]}$ the estimate

the TD temporal difference error.

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] \qquad (6.3)$$
$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \qquad (\text{from } (3.9))$$
$$= \mathbb{E}_\pi[\underbrace{R_{t+1} + \gamma v_\pi(S_{t+1})}_{\text{the MC Target}} \mid S_t = s]. \qquad (6.4)$$

Roughly speaking, Monte Carlo methods use an estimate of (6.3) as a target, whereas DP methods use an estimate of (6.4) as a target. The Monte Carlo target is an estimate because the expected value in (6.3) is not known; a sample return is used in place of the real expected return. The DP target is an estimate not because of the expected values, which are assumed to be completely provided by a model of the environment, but because $v_\pi(S_{t+1})$ is not known and the current estimate, $V(S_{t+1})$, is used instead. The TD target is an estimate for both reasons: it samples the expected values in (6.4) *and* it uses the current estimate V instead of the true v_π. Thus, TD methods combine the sampling of

Monte Carlo with the bootstrapping of DP. As we shall see, with care and imagination

公式 policy → Value

MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

设为0.

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

the memory only stores one trajectory. at the end of the trajectory. you need to remember the whole trajectory.

Generate an episode following π: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$ ⇒遵从 policy π.

MC: 想要更新，得等待episode出现 从头走到尾

(when you generate this sequence, you are also remembering it)

$G \leftarrow 0$

one episode is done.

after one episode is over and before the other episode is done. (should the agent just wait until this computation is done?)

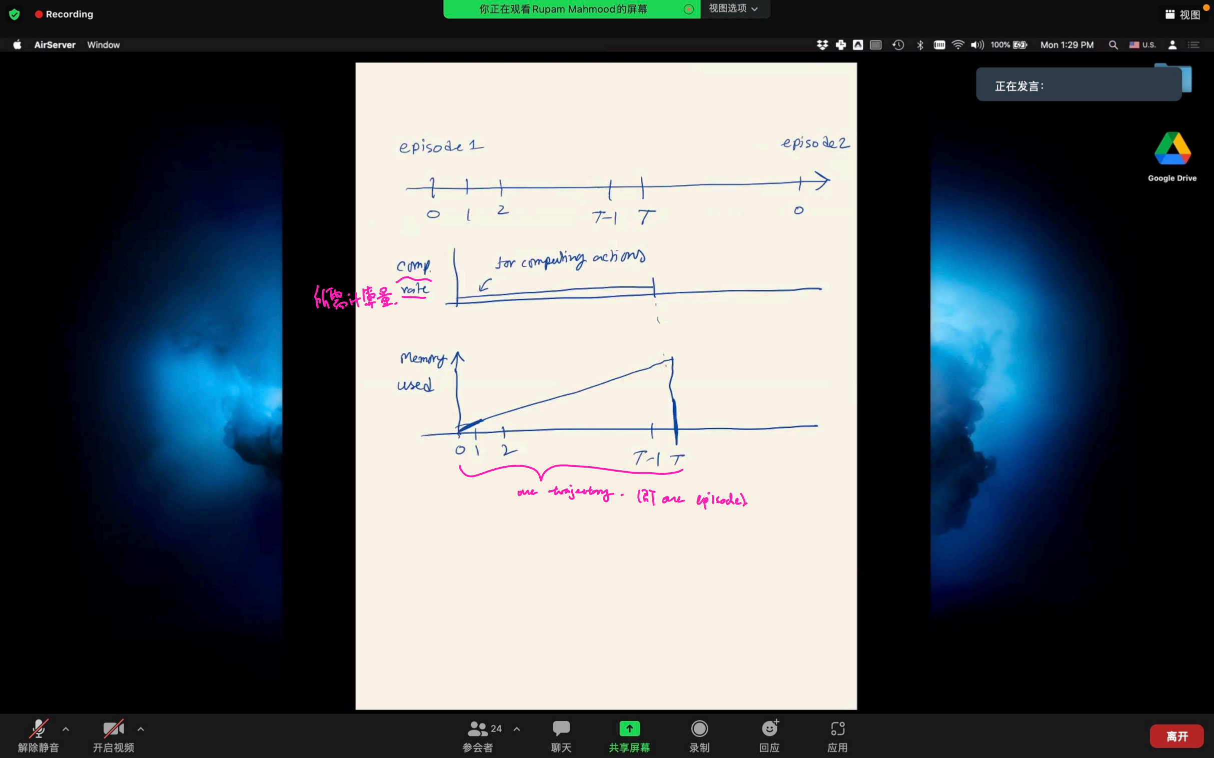
Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:

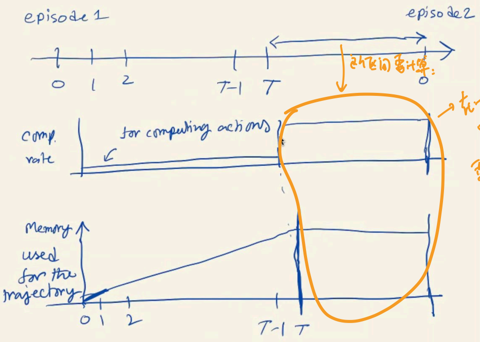
$G \leftarrow \gamma G + R_{t+1}$

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$ ⇒在不知P的情况下，仅用π来计算 V-value.

lots of computation!!
expensive!!


Recording
你正在观看Rupam Mahmood的屏幕
视图选项
视图
AirServer
Window
Mon 1:29 PM
正在发言：
Google Drive
episode 1
episode 2
0
1
2
T-1
T
0
comp. rate
所需计算量.
for computing actions
Memory used
0 1 2
T-1 T
one trajectory. (即 one episode)
解除静音
开启视频
参会者
24
聊天
共享屏幕
录制
回应
应用
离开



这个区间需计算:

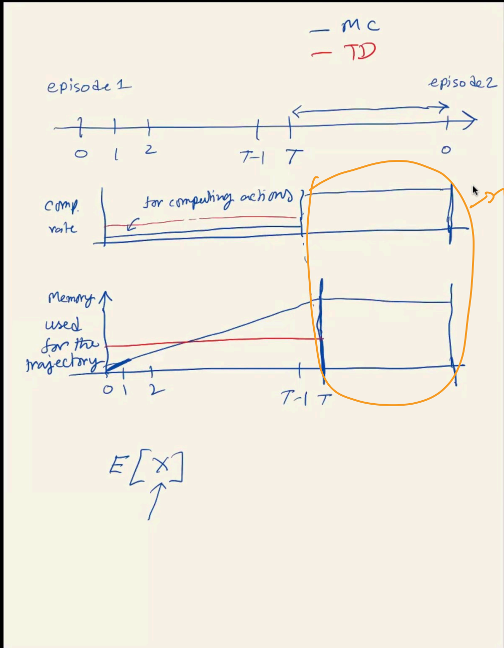
在一个episode结束后,
另一个episode开始前

需计算 $G = \gamma G + R_{t+1}$,

Append G,

$V(S_t) = \text{avg}(\text{Returns}(S_t))$

∴所需计算量极大.



TD预测下一步. 减少了很多计算量.
但它没有MC准确.

MC is better understanding than TD.