Optimal λ to use:

— tune up λ as a f^n of time?
— Objective f^n to optimize (f^n of λ)

10/Nov/2018

Recap:

$$\underbrace{G_t - V(S_t)}_{\text{monte carlo error.}} = \sum_{k=1}^{T-1} \gamma^{k-t} \delta_k \quad —(1)$$

where $\underbrace{\delta_k = R_{k+1} + \gamma V(S_{k+1}) - V(S_k)}_{\text{TD error.}}$

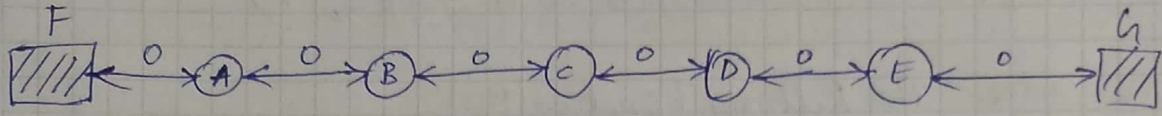
(1) holds provided V does not change with time.

DP-methods — value iteration / policy iteration.

TD	D P	Monte Carlo
– No system model	– Requires System model	– No system model
– Allows incremental updates	– Allows incremental updates	– Updates after completion of episode.

Although there is no theoritical proof, studies have shown TD is better than MC.

Eg: Random Walk



F & G are terminating states. Once we reach terminating state, reward = 0

Markow Reward process: A MRP is an MDP w/o actions ~~rewards~~

We assume that all episodes start in the center state C & proceed left or right by one step with equal prob

Episodes may terminate in either state F or state G.

When an episode terminates in G, a reward of 1 is obtained

——— " ——— F ——— " — 0 is obtained

Instance of a episode

C, 0, D, 0, C, 0, B, 0, C, 0, D, 0, E 0, 1, G

We are in a undiscounted setting. True value of a state is the prob of terminating in state G when starting from that state.

Let π be our policy $\pi(\text{left} \mid \text{non-term state}) = \pi(\text{right} \mid \text{non-term state}) = 1/2$

Note $V_\pi(A) = E[R_1 + R_2 \dots R_t \mid S_0 = A]$

$= E[R_T \mid S_0 = A]$ $\quad R_1 + R_2 \dots R_{T-1} = 0$ non terminating state

If MRP terminates in F, $r_T = 0$

If MRP terminates in G, $r_T = 1$

$$E[R_T \mid S_0 = A] = E[\, I\{\text{termination in } G\} \mid S_0 = A\}$$

$$V_\pi(A) = P(\text{termination in } G \text{ given } S_0 = A)$$

We solve the Bellman Equation now

$$V_\pi(A) = E[R_1 + V_\pi(\text{next state}) \mid S_0 = A]$$

$$= 0 + \frac{1}{2} V_\pi(A) + \frac{1}{2} V_\pi(B)$$

$$= 0 + 0 + \frac{1}{2} V_\pi(B)$$

$$V_\pi(B) = 2 V_\pi(A) \quad \text{— ①}$$

$$V_\pi(B) = E[r_1 + V_\pi(\text{next state}) \mid S_0 = B]$$

$$= 0 + \frac{1}{2} V_\pi(A) + \frac{1}{2} V_\pi(C)$$

$$\cancel{V_\pi(B) =} \quad \left(2 - \frac{1}{2}\right) V_\pi(A) = \frac{1}{2} V_\pi(C)$$

$$3 V_\pi(A) = V_\pi(C) \quad \text{— ②}$$

$$V_\pi(C) = E[r_1 + V_\pi(\text{next state}) \mid S_0 = C]$$

$$= 0 + \frac{1}{2} V_\pi(B) + \frac{1}{2} V_\pi(D)$$

$$3 V_\pi(A) = V_\pi(A) + \frac{1}{2} V_\pi(D)$$

$$4 V_\pi(A) = V_\pi(D) \quad \text{— ③}$$

$$V_\pi(D) = E[r_1 + V_\pi(\text{next state}) \mid S_0 = D]$$

$$= 0 + \frac{1}{2} V_\pi(C) + \frac{1}{2} V_\pi(E)$$

$$4 V_\pi(A) = \frac{3}{2} V_\pi(A) + \frac{1}{2} V_\pi(E)$$

$$V(E) = 5 V_\pi(A)$$

$V_\pi(E) = E[r_1 + V_\pi(\text{next state}) \mid S_0 = E]$

$= \frac{1}{2} + \frac{1}{2} V_\pi(G) + \frac{1}{2} V_\pi(D)$

$5 V_\pi(A) = \frac{1}{2} + 0 + \frac{1}{2} V_\pi(0)$

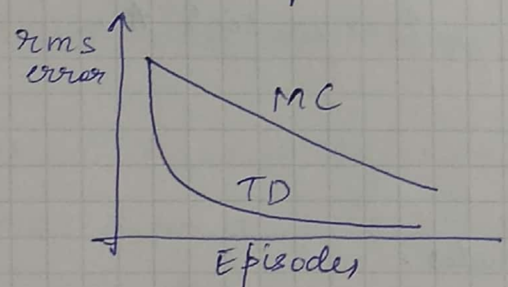
$3 V_\pi(A) = \frac{1}{2} \Rightarrow V_\pi(A) = 1/6$

$\therefore V_\pi(B) = 2/6, \; V_\pi(C) = 3/6, \; V_\pi(D) = 4/6, \; V_\pi(E) = 5/6$

Indicate the prob of ending in a terminating state

Apply TD & MC methods:

TD(0) = 100 episodes

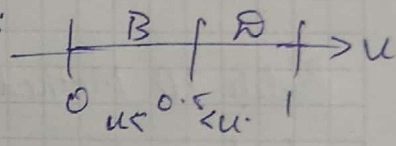


To DO: program.

$U \sim U[0,1]$

step 1 start: C

Step If:



TD has more oscillations,

MC has less oscillation since there aggregation

Example: Designing a prediction from data

Consider we have 8 different episodes

Episode 1 : A, 0, B, 0

2 : B, 1,

3 : B, 1

4 : B, 1

5 : B, 1

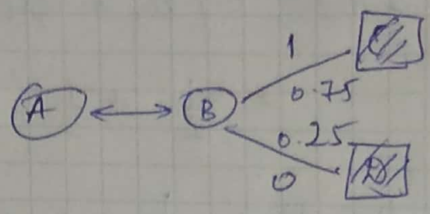
6 : B, 1

7 : B, 1

8 : B, 0

Predict value of A?

Predict value of B?



$V_\pi(B) = \frac{6}{8} = 3/4$

$V_\pi(A) = 3/4$

$$V_\pi(A) = E\{r_1 + V_\pi(\text{next, State})\} = r_1 + V_\pi(B)$$
$$= 0 + \frac{3}{4}$$
$$=$$

TD would predict:

$$V_\pi(A) = V_\pi(B) = 3/4$$

MC would predict:

$V_\pi(A) = 0$ since it sees at eph: $1 = 0$ & never sees A.

$V_\pi(B) = 0$

Batch TD Methods

Certainity Equivalance estimates:

Suppose we estimate transition prob $i \longrightarrow j$ as

$$\left(\frac{\text{no of transitions } i \longrightarrow j}{\text{Total no of transitions out of } i} \right)$$

& Expected reward = sample average of rewards on these transitions, then value function can be estimated assuming the above estimated model is the true model.

N_{ij} = no of transitions from $i \longrightarrow j$

N_i = — — — — out of i

$$P_{ij}^{(\pi)} = \frac{N_{ij}}{N_i}$$

R_i^n = n^{th} sample reward in state i

$$R_i = \frac{1}{N_i} \sum_{n=1}^{N_i} R_i^n$$

$$V(i) = R_i + \sum_j P_{ij} (V_i)$$

$$\boxed{V(i) = \frac{1}{N_i} \sum_{n=1}^{N_i} R_i + \sum_j \frac{N_{ij}}{N_i} V(j)}$$

Maximum Likelihood Parameter Estimation

Consider a random experiment whose we know the outcomes from a parameterized $p_\theta(\cdot)$ of distributions where θ is the parameter.

Goal: Estimate θ given that a certain outcome (k) has occured.

Thus, parameter θ will maximize the prob of occurance of 'k'

Let $\hat{\theta}_{ml}(k) \equiv$ parameter θ that maximises the likelihood $p_\theta(k)$ wrt θ

2 Maximizations possible in general

- max likelihood wrt θ given observation k
- max likelihood wrt k given θ.

Scanned by CamScanner

Example (max posterior)

Consider a biased coin showing heads w.p p

suppose the coin is tossed n-times

Let X = no of heads in n-tosses.

For $0 \leq k \leq n$, $\longrightarrow P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$

Let $\hat{p}_{ML}$ = value of p that maximises $P(X=k)$

$$\frac{d P(X=k)}{dp} = \frac{d}{dp}\left(\binom{n}{k} p^k (1-n)^{n-k}\right)$$

$$= \binom{n}{k}\left(k p^{k-1} (1-p)^{n-k} - (n-k) p^k (1-p)^{n-k-1}\right)$$

$$= \binom{n}{k}\left(\frac{k}{p} - \frac{(n-k)}{(1-p)}\right) p^k (1-p)^{n-k}$$

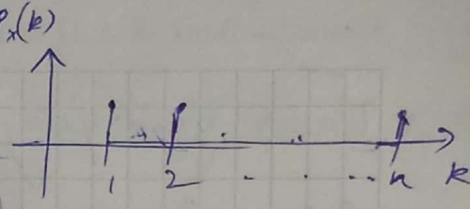
$$= \binom{n}{k} (k-np) p^{(k-1)} (1-p)^{n-k-1}$$

$$= 0 \quad \text{for } p = \frac{k}{n}$$

$$\left.\frac{d^2 P(X=k)}{dp^2}\right|_{p=k/n} = \underbrace{-\binom{n}{k} n\, p^{k-1} (1-p)^{n-k-1}}_{<0} + \underbrace{\binom{n}{k} (k-np) \frac{d}{dp}\left(p^{k-1} (1-p)^{n-k-1}\right)}_{0}$$

Example: (max likelihood)

Suppose X is drawn at random from numbers 1 to n with each possibility equally likely. Assume 'n' is unknown but that $X=k$ is observed. Find the ML estimation of n given $X=k$ is observed.



Note: $p_X(k) = P(X=k)$

$$= \frac{1}{n}\, I_{\{1 \le k \le n\}}$$

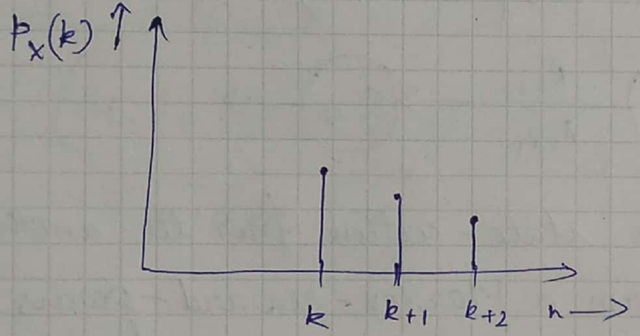
One can view

$p_X(k)$ as a function of n & not k

$p_X(k) = 0$ if $n \le k-1$

$= \frac{1}{k}$ if $n = k$

$= \frac{1}{k+1}$ if $n = k+1$



The max value of n, $\hat{n}_{ML} = k$ $(\theta = k)$

TD – algorithm for prediction

SARSA – on-policy TD control.

Idea:

Use generalize policy iteration together with TD learning.

Under given policy π, the sequence of states $\{S_n\}$ is markov.

- Under given policy π, the sequence of state-action tuples $\{(S_n, A_n)\}$ is also Markov.

$$P(S_{n+1}=j, A_{n+1}=a \mid S_n=i, A_n=b)$$

$$= P(A_{n+1}=a \mid S_{n+1}=j, S_n=i, A_n=b)\, P(S_{n+1}=j \mid S_n=i, A_n=b)$$

$$= P(A_{n+1}=a \mid S_{n+1}=j)\, P(S_{n+1}=j \mid S_n=i, A_n=b)$$

$$= \pi(a \mid j)\, p_\pi(i, b, j)$$

$$(S_n, A_n) \xrightarrow{\pi(A_{n+1}/S_{n+1})\, p_\pi(S_n, A_n, S_{n+1})} (S_{n+1}, A_{n+1})$$

$$\cdots \to (S_i) \xrightarrow[A_i]{R_{i+1}} (S_{i+1}) \xrightarrow[A_{i+1}]{R_{i+2}} (S_{i+2}) \to \cdots$$

Transitions from one state-action pair to another state-action pair form a Markov-reward-process

- state to state transitions (under a given policy) estimate value function $V^\pi(.)$
- State to state-action functions (- - -) estimate action-value function $Q^\pi(\cdot,\cdot)$

TD(0) on the joint markov chain gives,

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right]$$

Assumption: All state action tuples are visited infinitely often.

$$\dot{Q}(s,A) = R(s,A) + \gamma \sum_{(s',A')} p(s,A,s')\,\pi(A'/s')\,Q(s',A') - Q(s,A)$$

one stable fixed pt of DDE is RHS = 0

$$Q(S,A) = R(S,A) + \gamma \sum p(s,A,s')\,\pi(A'|s')\,Q(s',A')$$

Bellman eqn for of the state-action markov chain.

The TD(0) update for the joint (state-action) Markov Chain is done after every transition from a non-terminal state S_t. If S_{t+1} is a terminal state, then $Q(S_{t+1}, A_{t+1}) \equiv 0$

$$\underbrace{(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})}_{\text{SARSA algo}} \ldots$$

- Continually estimate q_π for the behavior policy π & then update π greedily wrt q_π [using GPI or some variant]

- Adopt and ϵ-greedy policy.

SARSA

Algo param: step size $\alpha \in (0,1]$, small $\epsilon > 0$
Initialize $Q(S,a)$, for all $s \in S^T$, $a \in A(s)$ arbitrarily expect that $Q(\text{terminal}, \cdot) = 0$

&

Loop for each episode

1. Intialize S
2. Choose A from S using policy derived from Q (eg: ε greedy)

Loop for each step of episode:

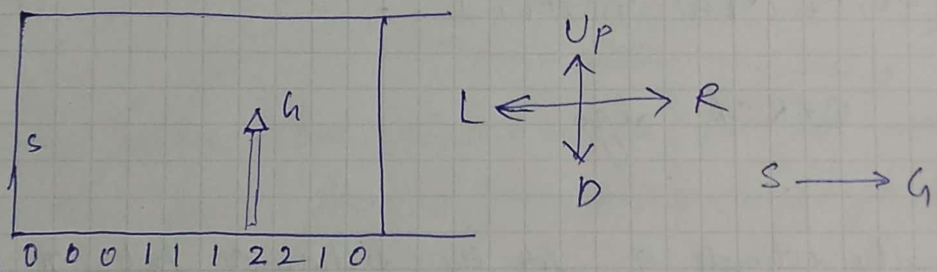
- Take action A, observe R, S'
- Choose A' from S' using policy derived from Q (eg greedy)

$$Q(S,A) \leftarrow Q(S,A) + \alpha\,[R + \gamma\, Q(S',A') - Q(S,A)]$$

- $S \leftarrow S'$, $A \leftarrow A'$

Until S is terminal state

Eg: Windy Grid World



Wind intensity

the number indicates the offset

$\epsilon = 0.1$ S → G $\alpha = 0.5$, $Q(s,a) = 0 \ \forall (s,a)$ initialization

Each episode ~ 17 after 8000 steps