Maximisation bias

Van Hasselt (2010, 2015)

$$Q(s,a) = Q(s,a) + \alpha \left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a) \right)$$

→ tends to overshoot (by a lot)

Abstract setting:

$X_1, \ldots, X_n$ random variables

$\delta_1, \ldots, \delta_n$

$\max_i \mathbb{E}[X_i]$

single estimator:

for each i,

Sample X_i a number of times

→ choose $\arg\max_i \hat{\mathbb{E}}[X_i] = *$

this is the candidate best machine

→ evaluate using $\hat{\mathbb{E}}[X_*]$

double estimator:

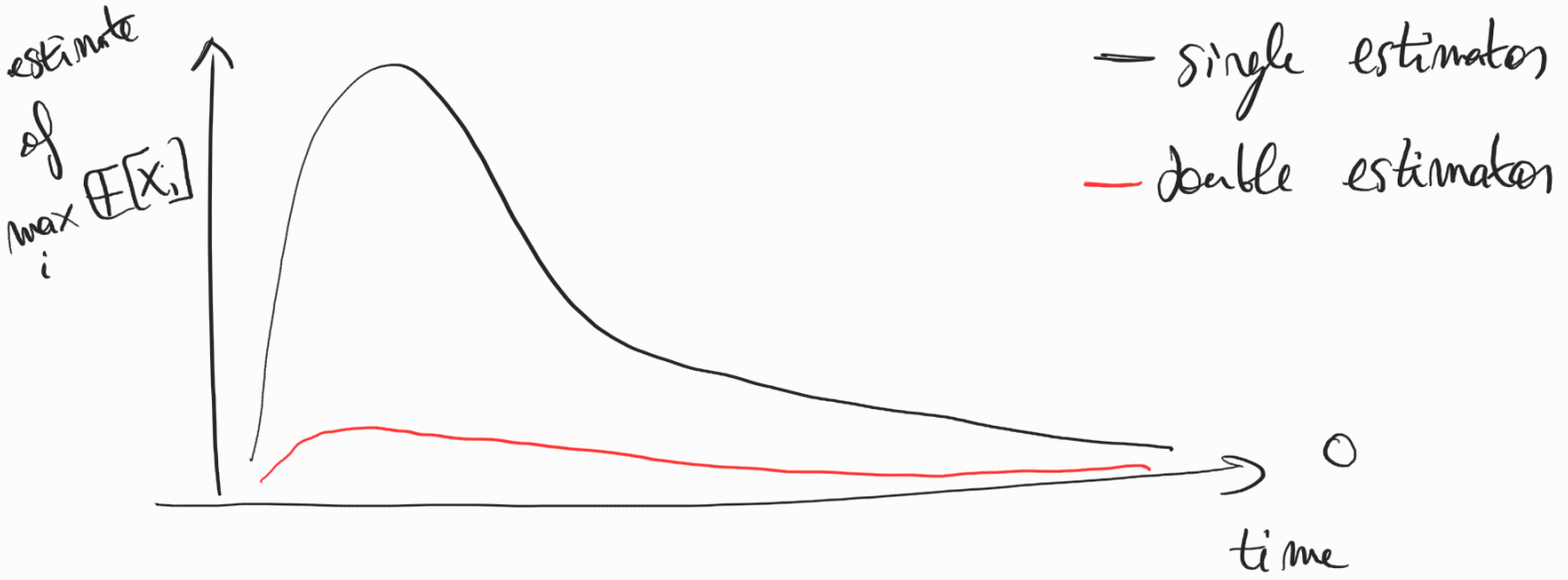
for each i,

Sample X_i a number of times, twice

→ choose $a = \arg\max_i \hat{E}[X_i] = *$ using the first set of samples

this is the candidate best machine

→ evaluate using $\hat{E}[X_*]$ using the second set of samples



Double Q-learning

$$a' = \arg\max_{a'} Q_2(s', a')$$

$$Q_1(s,a) \leftarrow Q_1(s,a) + \alpha\left(r + \gamma Q_1(s', a') - Q_1(s,a)\right)$$

$Q_1(s,a) = Q_1(s,a) + \alpha($ $Q_1(s',a')$ $Q_1(s,a)$

symmetrically for updating Q_2.

Exercise: take yesterday's code and implement double Q-learning

REPEAT:

$S \leftarrow S_0$

while (trajectory not over):

choose action a using ε-greedy strategy from q

sample (s,a) : r, s'

with probability 1/2:

$$a' = \arg\max_{a'} Q_2(s',a')$$

$$Q_1(s,a) = Q_1(s,a) + \alpha(r + \gamma \; Q_1(s',a') - Q_1(s,a)$$

else:

$$a' = \arg\max_{a'} Q_1(s', a')$$

$$Q_2(s,a) = Q_2(s,a) + \alpha\left(r + \gamma\, Q_2(s', a') - Q_2(s,a)\right)$$