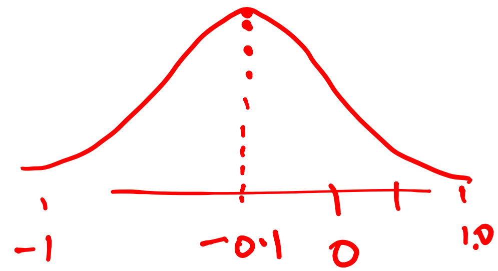
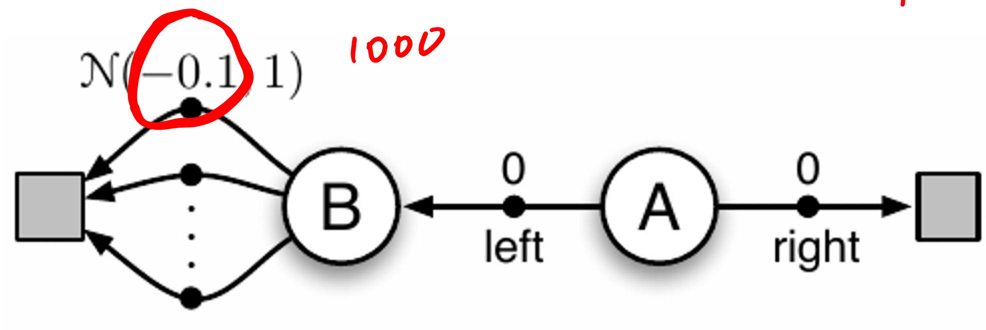
# Maximization Bias

B. Ravindran

# Maximization Bias



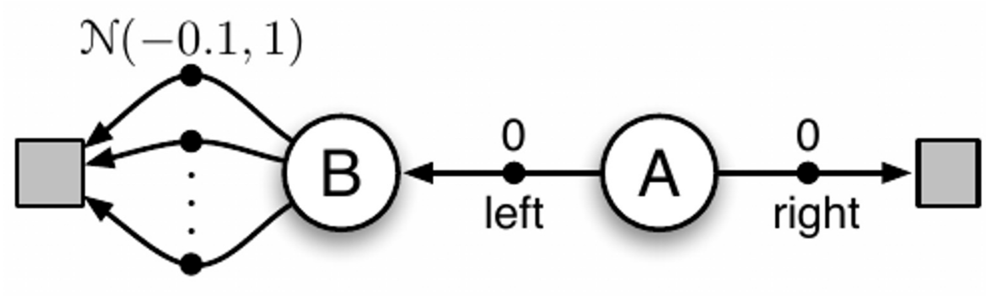

Consider the simple example below:




1. A is the starting state.
2. T(A, left, B) = 1
3. R(A, left) = 0, R(A, right) = 0
4. From B, there are |N| actions available, each of which results in a terminal state. And these |N| actions are normally distributed with mean = -0.1 and std = 1

# Maximization Bias

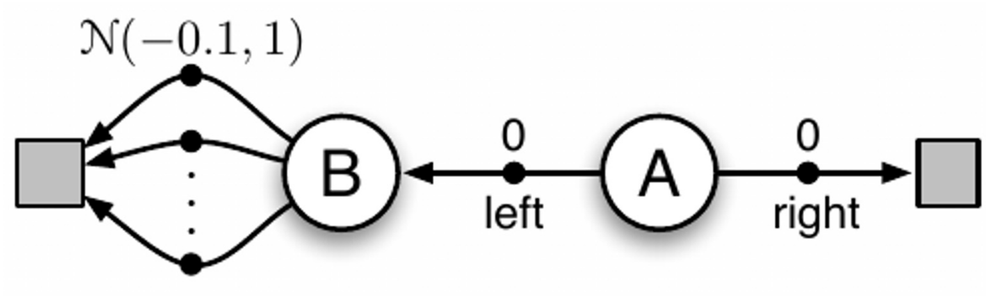
Consider the simple example below:




1. A is the starting state.
2. T(A, left, B) = 1
3. R(A, left) = 0, R(A, right) = 0
4. From B, there are |N| actions available, each of which results in a terminal state. And these |N| actions are normally distributed with mean = -0.1 and std = 1

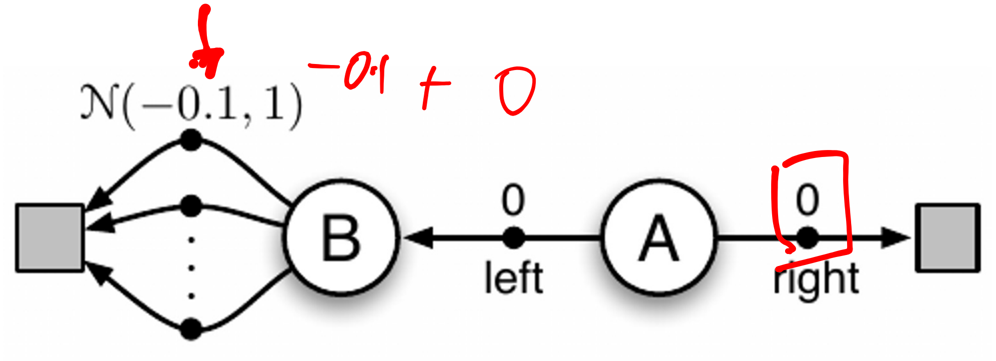
# Maximization Bias

Which direction to move from A?



# Maximization Bias

Which direction to move from A?

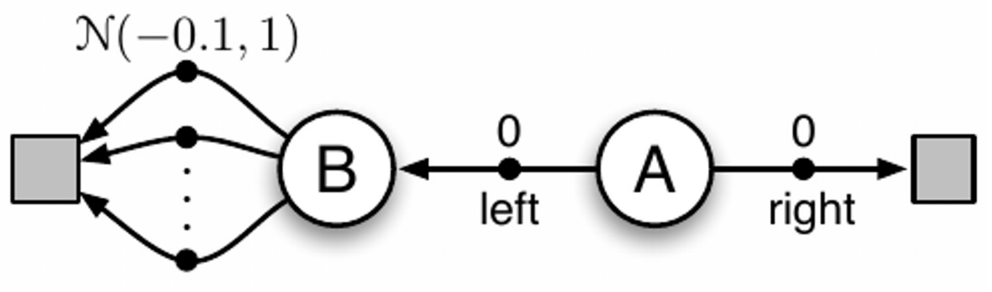



E[ $G_t$ | $s_0$ = A, $a_0$ = left ] = - 0.1

E[ $G_t$ | $s_0$ = A, $a_0$ = right ] = 0

# Maximization Bias

Which direction to move from A?



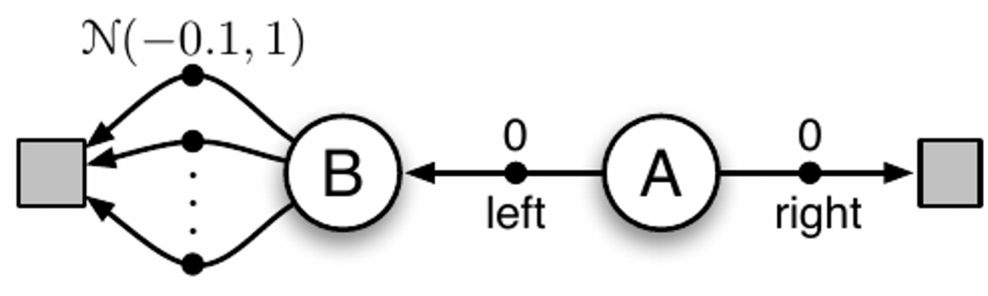

$E[\, G_t \mid s_0 = A, a_0 = \text{left} \,] = -0.1$

$E[\, G_t \mid s_0 = A, a_0 = \text{right} \,] = 0$

# Maximization Bias

$Q(B,a) = 0$
$\forall a$

What happens when we learn a policy using Q-learning?




$Q(A, \text{left})$

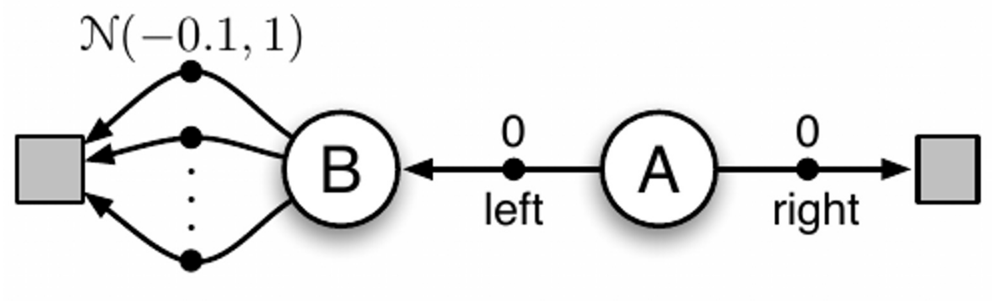
$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)\right]$$

$Q(A,\text{left}) + \alpha\left[0 + \gamma \max_a Q(B,a) - Q(A,\text{left})\right]$

$i = 1, \ldots, 1000 \quad Q(B,a_i) \leftarrow Q(B,a_i) + \alpha\left[r + \cdots - Q(B,a_i)\right]$

# Maximization Bias

What happens when we learn a policy using Q-learning?

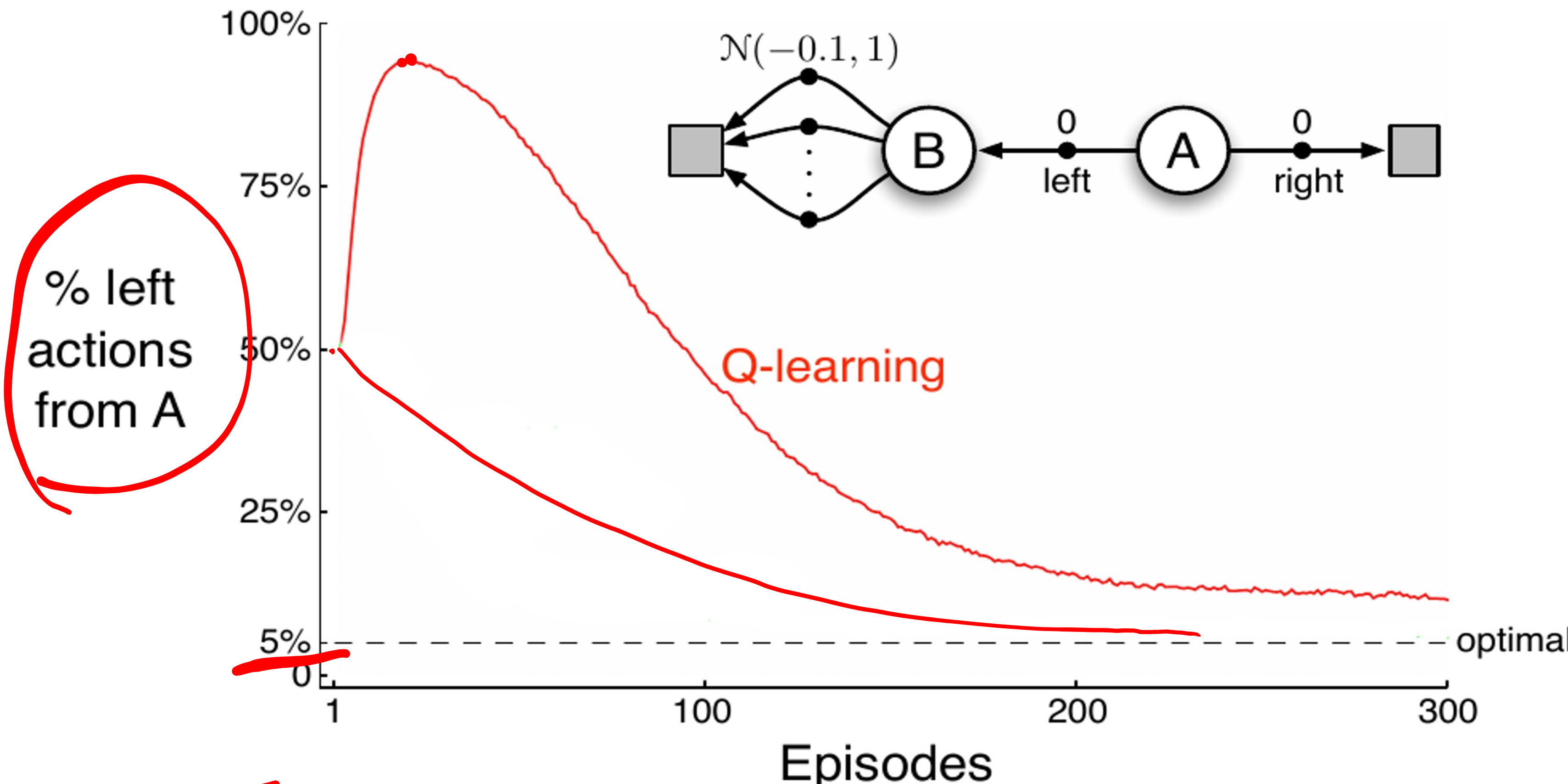



$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right]$$

Using maximum over estimate
as an estimate of the maximum!

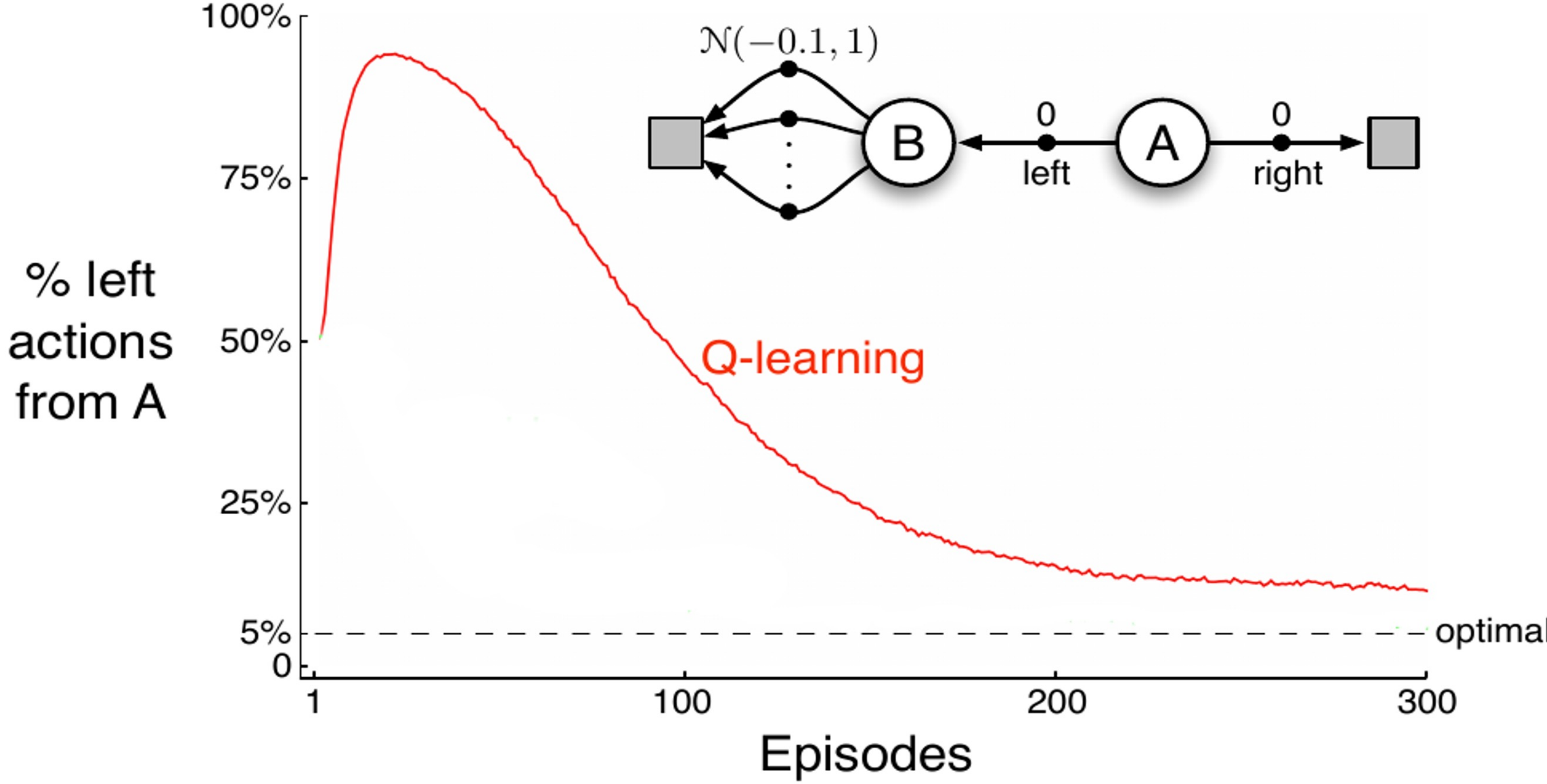
Leads to a positive bias, called **maximization bias**.

# Maximization Bias




$\epsilon$ = 0.1 for above example
Therefore, 10% of the actions are random.
Optimal => 5% can be right (random) and 95% should be left.

# Maximization Bias



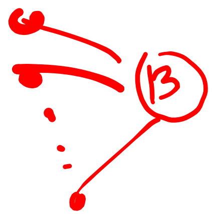

ε = 0.1 for above example
Therefore, 10% of the actions are random.
Optimal => 5% can be right (random) and 95% should be left.

# Double Q-learning

The problem can also be viewed as:

using the same samples both to determine the maximizing action and to estimate its value

# Double Q-learning




$\hat{a} = \arg\max_a Q_1(s,a)$

$Q_2(s, \hat{a})$

The problem can also be viewed as:

using the same samples both to determine the maximizing action and to estimate its value

Solution: Use different estimates for maximizing the action and estimating its value

$Q_1$

$r_1 \quad r_2 \quad r_3 \quad r_4 \quad \ldots \quad r_{100}$

$Q_2$

$Q_1$ ave$(r_1 + r_3 + \ldots)$

$Q_2$ ave$(r_2 + r_4 + \ldots)$

# Double Q-learning

**Double Q-learning, for estimating $Q_1 \approx Q_2 \approx q_*$**

Algorithm parameters: step size $\alpha \in (0, 1]$, small $\varepsilon > 0$
Initialize $Q_1(s, a)$ and $Q_2(s, a)$, for all $s \in \mathcal{S}^+, a \in \mathcal{A}(s)$, such that $Q(\textit{terminal}, \cdot) = 0$

Loop for each episode:
  Initialize $S$
  Loop for each step of episode:
    Choose $A$ from $S$ using the policy $\varepsilon$-greedy in $Q_1 + Q_2$
    Take action $A$, observe $R$, $S'$
    With 0.5 probabilility:

$$Q_1(S, A) \leftarrow Q_1(S, A) + \alpha\Big(\overbrace{R + \gamma Q_2\big(S', \arg\max_a Q_1(S', a)\big)}^{\text{TD target}} - Q_1(S, A)\Big)$$

    else:

$$Q_2(S, A) \leftarrow Q_2(S, A) + \alpha\Big(R + \gamma Q_1\big(S', \arg\max_a Q_2(S', a)\big) - Q_2(S, A)\Big)$$

    $S \leftarrow S'$
  until $S$ is terminal

# Double Q-learning

Double Q-learning vs Q-learning (ε = 0.1)