Three Key Ideas for RL

- **Model-based vs model-free learning**
  - What function is being learned?

- **Approximating the Value Function**
  - Smaller $\rightarrow$ easier to learn & better generalization

- **Exploration-exploitation tradeoff**
Q Learning

- For all \( s, a \)
  - Initialize \( Q(s, a) = 0 \)
- Repeat Forever
  Where are you? \( s \).
  Choose some action \( a \)
  Execute it in real world: \( (s, a, r, s') \)
  Do update:
  
  \[
  Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha \left[ r + \gamma \max_{a'} Q(s', a') \right]
  \]

Questions

- How to explore?
  - Random Exploration
  - Uniform exploration
  - Epsilon Greedy
    - With (small) probability \( \varepsilon \), act randomly
    - With (large) probability \( 1 - \varepsilon \), act on current policy
  - Exploration Functions (such as UCB)
  - Thompson Sampling
- When to exploit?
- How to even think about this tradeoff?
Regret

- Even if you learn the optimal policy, you still make mistakes along the way!
- Regret is a measure of your total mistake cost: the difference between your (expected) rewards, including youthful sub-optimality, and optimal (expected) rewards
- Minimizing regret goes beyond learning to be optimal – it requires optimally learning to be optimal

Two KINDS of Regret

- **Cumulative Regret:**
  - achieve near optimal cumulative lifetime reward (in expectation)

- **Simple Regret:**
  - quickly identify policy with high reward (in expectation)
RL on Single State MDP

- Suppose MDP has a single state and \( k \) actions
  - Can sample rewards of actions using call to simulator
  - Sampling action \( a \) is like pulling slot machine arm with random payoff function \( R(s,a) \)

![Multi-Armed Bandit Problem](image)

Multi-Armed Bandit Problem

UCB Algorithm for Minimizing Cumulative Regret

- \( Q(a) \) : average reward for trying action \( a \) (in our single state \( s \)) so far
- \( n(a) \) : number of pulls of arm \( a \) so far
- Action choice by UCB after \( n \) pulls:

\[
a_n = \arg \max_a Q(a) + \sqrt{\frac{2 \ln n}{n(a)}}
\]

- Assumes rewards in \([0,1]\) – normalized from \( R_{\text{max}} \).

Slide adapted from Alan Fern (OSU)
Theorem: The expected cumulative regret of UCB $E[Reg_n]$ after $n$ arm pulls is bounded by $O(\log n)$

- Is this good?
  Yes. The average per-step regret is $O\left(\frac{\log(n)}{n}\right)$

Theorem: No algorithm can achieve a better expected regret (up to constant factors)

Two KINDS of Regret

- **Cumulative Regret**: achieve near optimal cumulative lifetime reward (in expectation)

- **Simple Regret**: quickly identify policy with high reward (in expectation)
Simple Regret Objective

- **Protocol:** At time step $n$ the algorithm picks an “exploration” arm $a_n$ to pull and observes reward $r_n$ and also picks an arm index it thinks is best $j_n$ ($a_n$, $j_n$, and $r_n$ are random variables).
  - If interrupted at time $n$ the algorithm returns $j_n$.

- **Expected Simple Regret ($E[SReg_n]$):** difference between $R^*$ and expected reward of arm $j_n$ selected by our strategy at time $n$
  \[ E[SReg_n] = R^* - E[R(a_{j_n})] \]

What about UCB for simple regret?

**Theorem:** The expected simple regret of UCB after $n$ arm pulls is upper bounded by $O(n^{-c})$ for a constant $c$.

Seems good, but we can do much better (at least in theory).
- Intuitively: UCB puts too much emphasis on pulling the best arm
- After an arm is looking good, maybe better to see if $\exists$ a better arm
Incremental Uniform (or Round Robin)

Algorithm:
- At round \( n \) pull arm with index \((k \mod n) + 1\)
- At round \( n \) return arm (if asked) with largest average reward

Theorem: The expected simple regret of Uniform after \( n \) arm pulls is upper bounded by \( O(e^{-cn}) \) for a constant \( c \).

- This bound is exponentially decreasing in \( n! \)
  Compared to polynomially for UCB \( O(n^{-c}) \).

Can we do even better?

Algorithm \( \epsilon \)-Greedy: (parameter \( \epsilon \))
- At round \( n \), with probability \( \epsilon \) pull arm with best average reward so far, otherwise pull one of the other arms at random.
- At round \( n \) return arm (if asked) with largest average reward

Theorem: The expected simple regret of \( \epsilon \)-Greedy for \( \epsilon = 0.5 \) after \( n \) arm pulls is upper bounded by \( O(e^{-cn}) \) for a constant \( c \) that is larger than the constant for Uniform (this holds for “large enough” \( n \)).
Summary of Bandits in Theory

PAC Objective:
- **UniformBandit** is a simple PAC algorithm
- **MedianElimination** improves by a factor of \( \log(k) \) and is optimal up to constant factors

**Cumulative Regret:**
- **Uniform** is very bad!
- **UCB** is optimal (up to constant factors)

**Simple Regret:**
- **UCB** shown to reduce regret at polynomial rate
- **Uniform** reduces at an exponential rate
- **0.5-Greedy** may have even better exponential rate

Theory vs. Practice

- The established theoretical relationships among bandit algorithms have often been useful in predicting empirical relationships.
- But not always ....
Simple regret vs. number of samples

UCB maximizes $Q_a + \sqrt{\frac{2 \ln(n)}{n}}$

UCB[sqrt] maximizes $Q_a + \sqrt{\frac{2 \sqrt{n}}{n}}$