REINFORCEMENT LEARNING.

+ LEARNING ALGO: we can't easily give an input an unambiguous "RIGHT ANSWER" in RL like we can in SL
  e.g. ROBOT: teaching it to walk → "no single "correct" classification of state!

+ RL: includes REWARD FXN. Good vs. Bad results.
  Want to MAXIMIZE REWARD.
  Algorithm needs to choose the actions that MAXIMIZE EXPECTED REWARD OVER TIME.

I. MARKOV DECISION PROCESSES

+ MDP = $(S, A, \{P_{sa}\}, \gamma, R)$.
  + S = set of STATES
  + A = set of ACTIONS
  + $\{P_{sa}\}$ = set of STATE TRANSITION PROBABILITIES.
    For each STATE $s \in S$ ∧ ACTION $a \in A$: P_{sa} gives us the PROBABILITY DISTRIBUTION OVER THE STATE SPACE.
    Specifically: P_{sa} gives distribution over WHAT STATES WE WILL TRANSITION TO if we TAKE ACTION a WHILE AT STATE s.
  + γ = is the DISCOUNT FACTOR. It's always $\in [0, 1)$.
  + $R: S \times A \mapsto \mathbb{R}$ = the REWARD FUNCTION.

+ How MDP Dynamics work
  + Start in s_0.
  + Choose some action $a_0 \in A$ to take in MDP.
  + Choose some action... RANDOMLY TRANSITION to a SUCCESSOR STATE s_1,
  + The STATE will then RANDOMLY TRANSITION ... where $s_1 \sim P_{s_0 a_0}$ (s_1 is drawn from the p. dist of taking action a_0 in state s_0).
  + Now in s_1. Repeat the choosing of an action $a_1 \in A$, and randomly transition to state $s_2 \sim P_{s_1 a_1}$.
  + Pictorial rep: $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} \cdots$
  + After visiting sequence of states $s_0, s_1, s_2, \ldots$, we have
    TOTAL PAYOFF: $R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots$
    (or, if $R: S \to \mathbb{R}$: $R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots$

GOAL: Choose $a_i \in A$ that MAXIMIZE $\mathbb{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots\right]$
γ^t = DISCOUNTED TIMESTEP. Motivation: accrue POSITIVE REWARDS ASAP, (else, reward value "decays"). Hence why $\gamma \in [0, 1)$.

III LEARNING A MODEL FOR AN MDP

+ POLICY + VALUE FXN

+ A POLICY is : $\pi : S \to A$ (states to actions). Given a state, the policy gives us some $a_i \in A$.
+ We EXECUTE a POLICY IF, when $\pi(s) = a$, we TAKE ACTION a.
+ VALUE FXN: Defined FOR A POLICY.
  Value function is the EXPECTED REWARD given that we START in S and FOLLOW ACTIONS SPECIFIED BY π.

$$V^{\pi}(s) = \mathbb{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \mid s_0 = s, \pi\right]$$

+ BELLMAN EQUATIONS + OPT VAL FXN

+ Given fixed policy π, value equation V^{π} satisfies BELLMAN EQNS.

$$V^{\pi}(s) = \underbrace{R(s)}_{\text{Reward of starting in state } s} + \gamma \sum_{s' \in S} \underbrace{P_{s\,\pi(s)}(s')}_{\text{probability that taking action } \pi(s) \text{ at } s \text{ leads to state } s'} \underbrace{V^{\pi}(s')}_{\text{expected reward given we start in } s'}$$

+ $V^{\pi}(s)$ = immediate reward + expected sum of future discount awards.
+ Can rewrite the above summation term as $\gamma \mathbb{E}_{s' \sim P_{s\,\pi(s)}}[V^{\pi}(s')]$.
+ Can use Bellman's Eqns to SOLVE FOR V^{π}. IF $|S| < \infty$, then write $V^{\pi}(s)$ for each s, giving $|S|$ linear equations in $|S|$ variables ($V^{\pi}(s)$'s)
+ OPT VAL FXN: $V^{*}(s) = \max_{\pi} V^{\pi}(s)$.
  The value fxn w/ policy π that MAXIMIZES the expected reward given starting in S over all such value fxns.

$$V^{*}(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V^{*}(s').$$

$$V^{*}(s) = V^{\pi^{*}}(s) \geq V^{\pi}(s)$$

(*) $$\pi^{*}(s) = \arg\max_{a \in A} \sum_{s' \in S} P_{sa}(s') V^{*}(s')$$

$\rightarrow \pi^{*}(s)$: opt. pol for ALL $s \in S$

II. VALUE ITERATION + POLICY ITERATION.

+ VALUE ITER: 1) $\forall s \in S$: $V(s) := 0$ $\quad$ $:=$ ← assignment
  2) while NOT CONVERGED {
  $$\forall s \in S: \; V(s) := R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P_{sa}(s') V(s')$$
  }

+ SYNCHRO. UPDATE: Compute new $V(s)$ $\forall s \in S$, overwrite old values with new values. Can find π from convergence and (*)

+ ASYNCHRO. UPDATE: Order states, loop over, update them.

+ POLICY ITER: 1) Init π random 2) WHILE NOT CONVG: $V := V^{\pi}$ $\pi(s) :=$ (*) but V not V^{*}

III. LEARNING A MODEL FOR AN MDP

+ In MOST PROBLEMS we are NOT GIVEN EXPLICIT STATE TRANSITION PROBS and REWARDS. Need to ESTIMATE FROM DATA.
+ IF we have multiple TRIALS:

$$s_0^1 \xrightarrow{a_0^1} s_1^1 \xrightarrow{a_1^1} s_2^1 \xrightarrow{a_2^1} s_3^1 \xrightarrow{a_3^1} \cdots \; \} \text{ trial 1}$$

$$s_0^2 \xrightarrow{a_0^2} s_1^2 \xrightarrow{a_1^2} s_2^2 \xrightarrow{a_2^2} s_3^2 \xrightarrow{a_3^2} \cdots \; \} \text{ trial 2.}$$

THEN:

$$P_{sa}(s') = \frac{\#\text{ times took action } a \text{ in state } s, \text{ got to } s'}{\#\text{ times took act. } a \text{ in } s.}$$

IF $P_{sa}(s') = 0/0 \rightarrow 1/|S|$.

+ One possible algo to learn in MDP w/ unknown P's:
  1) INIT π randomly
  2) REPEAT {
     1) Execute π in MDP for k trials
     2) Update estimates for P_{sa} (and R) based on the k trials
     3) Apply VALUE ITERATION w/ P_{sa}'s, R's, to get new V.
     4) Update π to be greedy policy WRT V.

  }

IV. Q-VALUE

+ $Q^{\pi}(s,a) = \mathbb{E}\left[R(s_0,a_0) + \gamma R(s_1,a_1) + \gamma^2 R(s_2,a_2) + \cdots \mid s_0 = s, a_0 = a\right]$.
+ With the V^{π}-like decomposition, we write for Q^{π}:

Recall $V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s')$.

So,

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s' \in S, a' \in A} P_{s\pi}(s',a') Q^{\pi}(s',a')$$