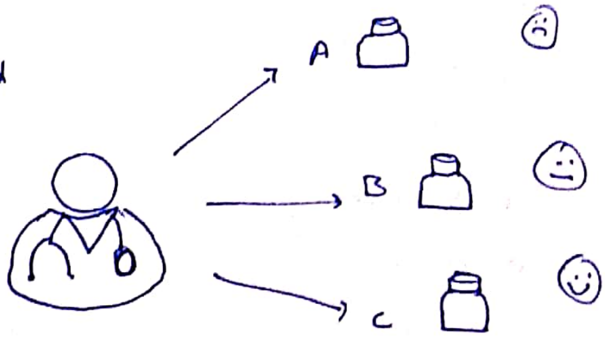
# Fundamentals of reinforcement learning

$$q_*(a) \doteq E[R_t \mid A_t = a] \quad \forall a \in \{1, \dots, K\}$$
$$= \sum_r p(r \mid a)\, r$$

Goal: maximize expected reward

$$\underset{a}{\operatorname{argmax}}\ q_*(a)$$




- Each action may have different distribution for $q_*()$
- $q_*()$ is mean for each distribution

**Summary**

- Decision making under uncertainty can be formalized by the K-armed bandit problem.
- Fundamental ideas: actions, reward, value function.

**Decaying past rewards**

$$Q_{n+1} = Q_n + \alpha_n (R_n - Q_n)$$
$$= \alpha_n R_n + Q_n - \alpha Q_n$$
$$= \alpha_n R_n + (1-\alpha) Q_n$$
$$= \alpha_n R_n + \alpha_n R_{n-1}(1-\alpha) + \alpha_n R_{n-2}(1-\alpha)^2 + \dots + (1-\alpha)^n Q_1$$

Target: ① Define exploration-exploitation tradeoff
② Define epsilon-greedy

**Epsilon greedy action selection**

$$A_t \leftarrow \begin{cases} \underset{a}{\operatorname{argmax}}\ Q_t(a) & \text{with prob } 1-\epsilon \\ a \sim \text{Unif}(\{a_1, \dots a_k\}) & \text{with prob } \epsilon \end{cases}$$

**Optimistic initial value**

- → Can only drive early exploration
- → Not well-suited for non-stationary problems
- → May not know how to choose optimistic initial value.

Scanned with CamScanner

## Upper confidence Bound - Action selection

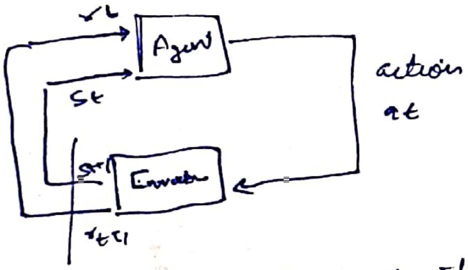
- Optimism in the face of uncertainty.

$$A_t = \arg\max \left[ \underbrace{Q_t(a)}_{\text{Exploit}} + \underbrace{c\sqrt{\frac{\ln t}{N_t(a)}}}_{\text{Explore}} \right]$$

~~beat~~ Tempo

Input $\longrightarrow$ Main chord

## Markov Decision Process (General framework for sequential decision making)




$P(s', r \mid s, a)$
$\hookrightarrow$ Dynamics of MDP defined by prob distributions

Markov property: Present state contains all necessary info to predict the future.

## Policies & value functions

- policy $\rightarrow$ mapping from state to probabilities of selecting each possible actions.
- Value function $\rightarrow$ $V_\pi(s)$ $\Rightarrow$ Expected return when starting in state $s$ and following $\pi$ thereafter.

$$V_\pi(s) = E_\pi[G_t \mid S_t = s] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right] \quad \forall s \in S$$

Similarly $q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a] = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$

Bellman equation, $V_\pi(s)$ related to $V_\pi(\,)$ of successor states.

$$V_\pi(s) = \sum_a \pi(a \mid s) \sum_{s', r} P(s', r \mid s, a)\left[r + \gamma V_\pi(s')\right] \quad \forall s \in S$$

# Fundamentals of reinforcement learning

## Optimal value functions

$V_*$ $\quad v_{\pi_*}(s) \doteq E_{\pi_*}[G_t \mid S_t = s] = \max_\pi v_\pi(s) \quad \forall s \in \mathcal{S}$

$q_*$ $\quad q_{\pi_*}(s,a) = \max_\pi q_\pi(s,a) \quad \forall s \in \mathcal{S} \ \& \ a \in \mathcal{A}$

$$v_*(s) = \max_a \sum_{s'} \sum_r p(s', r \mid s, a)[r + \gamma v_*(s')]$$

↳ Bellman optimality eqn for $v_*$

Week 4

① Policy evaluation

↳ diffn btwn policy eval & control
↳ dynamic programming setting
↳ iterative policy evaluation algo

② Policy iteration

↳ policy improvement theorem
↳ value function for a policy to produce better policy
↳ finding optimal policy.
↳ Dance of policy & value
↳ optimal policy & optimal value function.

③ Generalized policy iteration

↳ value iteration
↳ synchronous & asynchronous dp methods
↳ brute force for optimal
↳ Monte Carlo for value function
↳ advantage of dp

Scanned with CamScanner