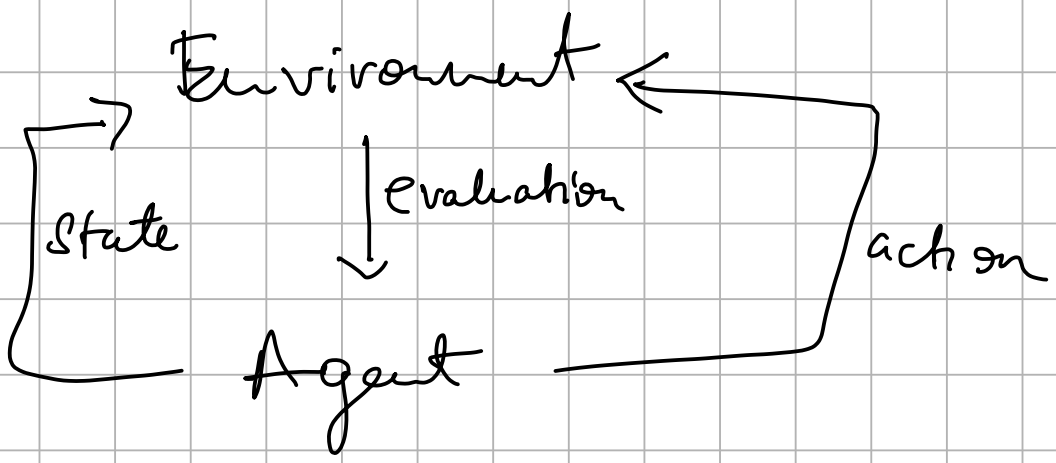
Reinforcement Learning

RL framework



- Learn from close interaction
- stochastic environment
- Noisy delayed scalar evaluation
- maximize a measure of long term peyomance

Temporal Difference

- Simple rule to explain complex behaviors
- Intuition:
  Prediction of outcome at time $t+1$ is better than the prediction at time t. Hence use the later prediction to adjust the earlier predictions.
- Has had profound impact in behavioral psychology and neuroscience

Explore Exploit Dilemma

- One key question - the dilemma between exploration and exploitation.
- Explore to find profitable actions
- Exploit to act according to the best observations already made
- Bandit problems encapsulates 'Explore vs Exploit'

Immediate Reinforcement Learning problems

- Problems where:
  at every time t, we pick an action a_t and get the reward r_t
- We need exploration for any reinforcement learning problems
- We must stop our exploration and "exploit" a solution that was found (we cannot keep exploring forever)

Set of actions $A = \{1, 2, \ldots n\}$ and each action gets a pay-off or cost or reward or evaluation

note

one action can have multiple rewards. Example:

Tossing a coin ← action

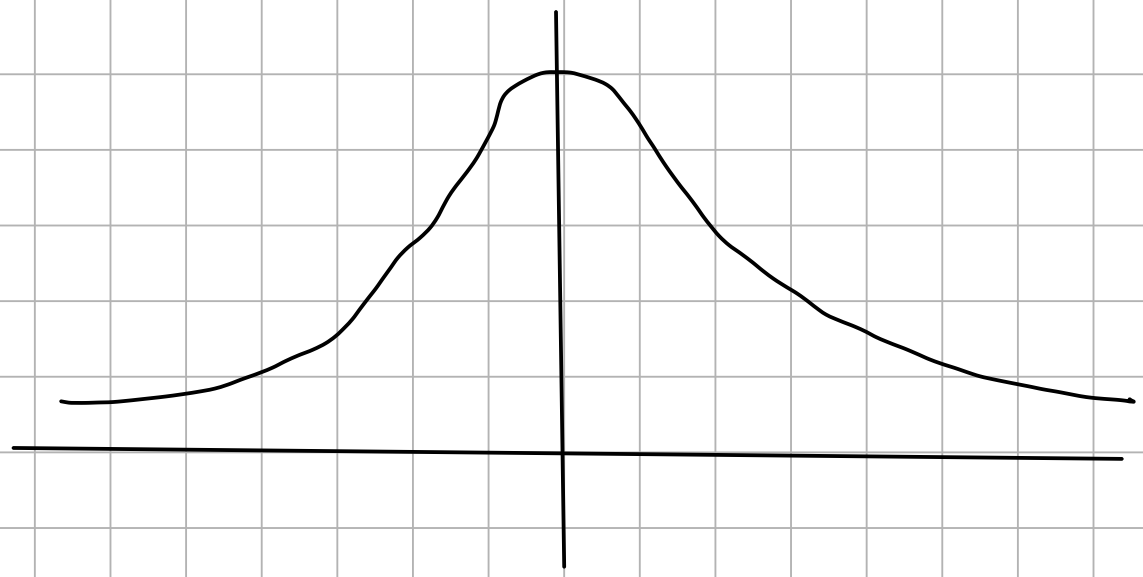
Reward ← { 1 if head
0 if tails

So different reward for the same action

So the reward depends on the probability

So the coin toss reward is Sampled from a Bernaulli Distribution

Lets say any action form $A = \{1, 2, \ldots n\}$ samples from Gaussian distribution



$q_{v*}(a)$

↑ action

↑ gives the best reward of action a (true expected reward)

Bandit Problems [multi arm bandit]

note

actions are called <u>arms</u> sometimes

Bandit Problem means that there are n number of actions and we get a reward after each action taken

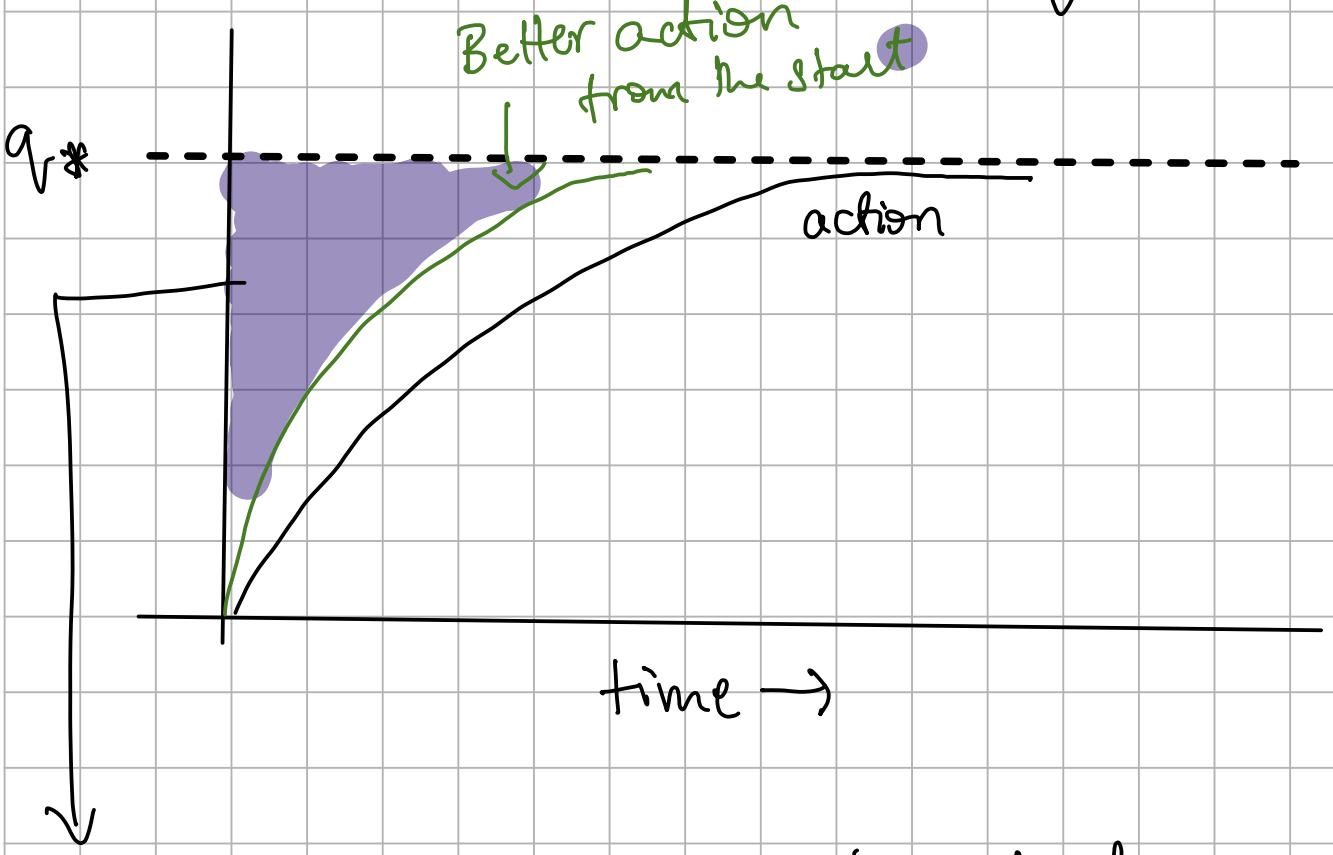
Solving multi arm bandit problems.

one way is

- Asymptotic Correctness
  - this is defined as we keep taking actions, but eventually we must select the 'arm' that has the highest reward (how quickly can we get that arm is another issue seperately)

- Regret Optimality

  ↳ trying to maximize the total reward from the beginning



This area is or loss, this is called regret

So in regret optamility, we are trying to decrease the regret as much as possible

- PAC optimility

Probably correct ← can be correct & wrong

Approximetly correct ← it is definitly wrong but close to the correct answer

PAC ← Probably approximately correct (highly)

PAC is represented as (ϵ, δ) pac

where ϵ ← approximately variable,

δ ← probablily path variable

So for formulation

$$\text{Prob}\left[q_*(a) \geq q_*(a^*) - \epsilon\right] \geq (1-\delta)$$

a^* ← action being taken