RL-Week-1

L1 Intro

- ML → learn func^{n} from input to output from data
- familiar model → learning from data
- Cycling, Talk → trial & error, evaluat^{n} (not instruc^{n}, RL (feedback mechanism) → walking also (people cheer if you do good, see if you fall you cry)
- RL → trial & error learning paradigm (rewards & punishments)
- learn abt. a sys. thru interac^{n}, by behavioural psychology.
- Pavlov's dog → bell, food, salivate, dog
- from 80s, chatgpt (human feedback), complex dyanmics (helicopter, humanoid), complex workspace, stochastic sensing & actua^{n} (visual atten^{n}), human in the loop learning, cognitively motivated learning, news customization, ad selec^{n}, recommenda^{n} engines → finally learn thru self-play
- TD Gammon (learn completely by self play), Arcade games (using complex neural network, learn to play from video input), AlphaGo, AlphaZero, 7 years

- AlphaFold (solved protein structure), Protein structure database

L2 Tic-Tac-Toe & TD Learning

for sup. learning
current posit^n
⇓
the next move can be labelled

- by RL → learn from real (+1 / -1 / 0), you have to learn from repeated play
- MENACE (matchbox engine) - if you make a valid / winning move, beads increase that means probab. of that move has increased.
- assume opponent is an imperfect opponent (it makes mistakes)
- there are certain positions in the game, from where you will certainly win. ∴, increase the probab. of such moves.
- TD (Temporal difference learning) - Δ the probabilities by the timeline.
- pred^n of outcome at time t+1 is better than the pred^n at time t. So, now use the later pred^n to adjust the earlier pred^n.

- RL → Deep RL. Goal → omniscience learning (consume any info. to learn)

L3 Immediate RL & Bandits

- limits of explore & exploit
- IRL ⇒ payoff accrues immediately after an ac^n is chosen. Bandit problem
- explore to find profitable ac^n. exploits to act according to the best observa^{ns} so far. Always doing str. is not optimal.
- MAB (n-arm) selects a particular ac^n from a set of n act^{ns} (1, 2, 3 ... n). Each selec^n gives Rewards from respective probab. distribu^n.
  Arm i has a distri. with mean μ_i

$$\mu^* = \max\{\mu_i\}$$

- Objectives
  1. Identify the correct arm eventually.
  ⋮
- $r_{i,k}$ reward when i^{th} arm for k^{th} time

$$Q(a_i) = \frac{\sum_{k=1}^{n_i} r_{i,k}}{\sum 1 = n_i \quad \{k : r_{i,k}\}}$$

$Q(a_i)$ ↓ expected payoff for ac^n a_i

true reward = μ_i (this i.d.k)

$$Q(a^*) = \max_i \{Q(a_i)\}$$

max. expected reward

$$Q_{k+1}(a_i) = Q_k(a_i) + \alpha\left(r_{i,k} - Q_k(a_i)\right)$$

$\alpha = \frac{1}{k_i+1}$ → yields the avg.

$$\bar{x}_k = \frac{x_1 + \dots + x_k}{k}$$

$$\bar{x}_{k+1} = \frac{x_1 \dots + x_{k+1}}{k+1} = \frac{k\bar{x}_k + x_{k+1}}{k+1}$$

$$\bar{x}_{k+1} = \bar{x}_k + \frac{1}{k+1}\left[x_{k+1} - \bar{x}_k\right]$$

Explora^n methods →

- Epsilon Greedy → asm $a^* = \text{argmax}_i\{Q_k(a_i)\}$ with probab. $1-\varepsilon$ and other asm with prob. ε
- Softmax → select max with probab. $\propto$ to current value estimates.

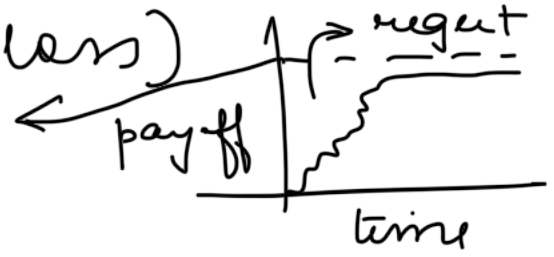
$$\Pi_k(a_i) = \frac{\exp\left(\frac{Q_k(a_i)}{\tau}\right)}{\sum_j \exp\left(\frac{Q_k(a_j)}{\tau}\right)}$$ → temp.

both give asymptomatic convergence (always).

L4 Regrets & PAC Frameworks

- Bandit use case → Yahoo News

- Obj. 2 → max. the total rewards obtained or min. regret (loss)



how to learn while keeping regret as less as possible.

- PAC Framework (Probably Approx. Correct)
  1. identificaⁿ of an ε-optimal arm with probab. $(1-\delta)$
  2. ε optimal : mean of the selected arm satisfies $\Rightarrow \mu > \mu^* - \varepsilon$
  3. min sample complexity - order of samples req. for such an arm identificaⁿ

→ Round based algo.

- PAC Guarantees → Median Eliminaⁿ, Upper Confidence Bounds (UCB), Thompson sampling (Bayesian Approach)

↳ Regret Optimality →

15 UCB → Upper confidence bound

- The arm with the best estimate a^* so far serves as a benchmark & other arms are played only if the upper bound of a suitable confidence interval is atleast a^*.

↳ so be greedy w.r.t. upper confidence bounds!

- Play machine j

$$\max\left\{Q(a_j) + \sqrt{\frac{2\ln n}{n_j}}\right\}$$

n → overall no. of plays

n_j → no. of times machine j has played so far

↓

sub optimal arm j is played less than $(8/\Delta_j)\ln n$ times.

16 Contextual Bandits

- challenge for customizaⁿ & ad selectⁿ for each user, not a generic 1-tier.
- diff. ads for diff. user. 1 bandit → 1 user.
  ↳ hard to train for 1 person
- assume that the params of the reward distribuⁿˢ themselves are determined by set of params. (groups are made for users) → linear parameterizaⁿ of the expectaⁿ.
- each user is represented by a set of features.
  (joint features of user & arm)
- the "static" → of choosing now depends on these features.
  ↳ check for the presence or absence of diff. signals.

- Lin UCB - most popular contextual bandit
  ↳ predicted expected reward assumed to be a linear func^n of features.
  1. use ridge reg. to fit
  2. get the upper confidence
  3. UCB like ac^n selec^n.
  4. better performance with lesser training data.

Practice Assignment