**Reinforcement Learning - N-armed Bandit**

**1. Basics of Reinforcing Learning**

* Reinforcement Learning vs Supervised Learning
  + Evaluative feedback vs Instructive feedback

(Uses training information to evaluate actions vs Instructs by giving correct actions)

* + Exploitation vs Exploration
* Tabular forms
  + Simple form: state space and action space are small enough to be represented as array/table
  + Exact solutions to optimal value function/policy can be found
* Finite MDP (Markov Decision Process)
  + (S,A,R,Gamma,P) – Bellman equations
  + 1) Dynamic programming – Model is known, Bootstrapping
  + 2) Monte Carlo – Model is unknown, No bootstrapping (e.g. importance sampler)
  + 3) Temporal differencing – Model is unknown, Bootstrapping

**2. N-armed bandit**

* Special case of RL: only a SINGLE STATE (Non-associative setting)
* Set-up:
  + Repeatedly choose among N different actions
  + Each choice has a reward from a stationary (or non-stationary) probability distribution depends on the action
  + Goal is to maximize the expected total reward (e.g. over 1000 actions/time steps)

(Each action has an expected reward/value of action, but this is not known with certainty)

* Greedy action (exploiting): always select the highest estimate value – immediate reward
* Non-greedy action (exploring): may produce greater total reward in the long run – uses uncertainty!
* Key idea: methods of exploration!

**3. Action-value methods**

* q(a) be the true value of action (true average)
* Q\_t(a) be the estimated value of action (sample average)
* By LLN, asymptotically converges

1. Greedy method

* A\_t = argmax[a] Q\_t(a) – maximizes immediate reward/exploitation
* Can got stuck in suboptimal actions

1. Eps-greedy method
   * With small probability eps, independently select an action at random
   * By LLN, asymptotically converges
   * Note: can gradually reduce eps with experience
2. Greedy vs Non-greedy
   * If reward variance is large, then non-greedy tends to perform better
   * If stationary + deterministic (no variance), then greedy tends to perform better
   * Even in the deterministic case when the bandits are non-stationary, exploration usually helps
   * Numerically:
     + Greedy method usually improves very fast at the beginning, but can got stuck in suboptimal actions
     + Eps-greedy:
       - If eps is big (e.g. 0.1), explores more, finds the optimal action earlier, but never selects it > 90% of time
       - If eps is small (e.g. 0.01), improves slowly, but eventually performs better
3. Incremental implementation
   * New <- old + step \* (target – old)
   * Les memory, faster computation
   * Non-constant step: guaranteed to converge to the true action values by LLN
   * In general, converge with probability 1 if:
     + Sum of steps is infinite (steps are large enough to eventually overcome any initial conditions or random fluctuations)
     + Sum of steps^2 is finite (eventually steps become small enough to assure convergence)
     + However, usually converges very slowly!
4. Non-stationary bandits
   * Bandit is no longer i.i.d., but changes (slowly) over time
   * Use constant step size – discount the rewards
   * Not guaranteed to converge (failed (2) above!)

**4. Optimistic Initial Values (Not generalizable to non-stationary problems)**

* Simple way to encourage exploration
* Initially, the optimistic method performs worse because it explores more, but eventually performs better because its exploration decreases over time
* Not effective for non-stationary problems because its drive for exploration is inherently temporary

**5. Upper-Confidence-Bound Action Selection (Not generalizable to non-stationary problems)**

* Eps-greedy action selection forces the non-greedy actions to be tried, but indiscriminately, with no preference for those that are nearly greedy or particularly uncertain
* UCB: select among the non-greedy actions according to their potential for actually being optimal, taking account into both how close their estimates are to being maximal and the uncertainties in those estimates
* Key idea:
  + Square root term is a measure of uncertainty or variance in the estimate
  + The quantity being max over is thus a sort of upper bound on the possible true value of action
  + C parameter determines the confidence level
  + Natural log means that the increase gets smaller over time but is unbounded, all actions will eventually be selected but as time goes by it will be a longer time, and thus lower selection frequency, for actions with a lower value estimate or that have already been selected more times
* Problems:
  + Not generalizable in non-stationary problems
  + Difficult to deal with LARGE STATE problems

**6. Gradient Bandits**

* All methods so far estimates action values => select actions
* GB: use preference H\_t(a) to select action (preference has no meanings) => obtain a selection probability distribution (softmax distribution), idea of stochastic gradient ascent (robust convergence properties)
* Involves baseline (usually average of rewards)

**7. Associative Search (Contextual Bandits)**

* Associative - associative different actions with different situations: learn a policy p(a|s) – a mapping from situations to the actions that are best in those situations
  + Example: several bandits, each play pick a bandit according to a distribution
  + Similar to N-armed bandit non-stationary problem
  + But unless q(a) changes slowly, all methods above don’t work well
* Non-associative – either find the single best action (stationary) or track the best action as it changes over time (non-stationary)
* 3 stages:
  + N-armed bandit: single state, usually stationary
  + Associative – multiple state (non-stationary)
  + Full RL – multiple state, actions affect next situation