Reinforcement Learning…

but first
Exploration v. Exploitation

Tie Matching

- You have a co-worker that always wears either a magenta tie, or a blue tie
- Turns out, it's random: he wears the magenta tie with probability P_m and the blue tie with probability $P_b=(1- P_m)$
- Let's say you get a reward r when you guess right

Expectation of reward

- If you guess magenta with probability Q_m and blue with probability $Q_r=(1- Q_m)$

$$E[r]=Q_mP_m+(1-Q_m)(1-P_m)=2Q_mP_m+1-Q_m-P_m$$

- Since this is linear in Q_m which is bound by the rules of probability, the extrema is either $Q_m=0$ if $P_m<0.5$, and $Q_m=1$ if $P_m>0.5$
- You always pick the tie with greater probability for maximum reward!

Interestingly..

- This is not what people do when given this task!
- Psych experiments have shown that people tend to set $Q_m= P_m$
- This is called *probability matching*
- However, it has been shown that people get the right answer (rather than probability matching) if:
  - They were provided with
    - large financial incentives,
    - meaningful and regular feedback, or
    - extensive training

This simple problem

- Indicates an interesting quandry of artificial intelligence:
  - Is AI supposed to be super-humanly rational (correct)?
  - Or is it supposed to be convincingly human (ala Turing Test)?
  - BTW, the Turing Test isn't what most people think it is, but that's another subject

What if you don't know P_m?

- Clearly, we must experiment to determine P_m
- Let's generalize by looking at the 2-armed bandit problem
- BTW, a one-armed bandit is a slot machine, aka a fruit machine, aka a machine that takes money from chavey kids in London, and old women in Vegas, while providing meaningless but colorful lights and noises as entertainment.
- Like SkyOne, but random

The two-armed bandit problem (Holland, 1975)

- Imagine two slot machines, side by side
- Sort of a slot machine with two arms:
- One gives reward with mean μ and std. dev. σ, the other has mean μ’ and std. dev. σ’
- μ>μ’
- The distributions have some overlap
- We should allocate all trials to the higher-payoff arm
- But we don’t know which arm is which

Exploration versus Exploitation

- This dilemma is a keystone of AI
- You must allocate some trials to both arms, so you can get an impression of which is which (exploration)
- Then you must allocate all remaining trials to the higher-payoff (exploitation)

Two sources of loss

- Let’s assume we have *N* trials to allocate
- We allocate *n* to the apparently less-good arm
- The loss associated with this strategy is

$$L(N-n,n)=|\mu-\mu'|\left[q(N-n,n)(N-n)+\left(1-q(N-n,n)n\right)\right]$$

- Where *q(N-n,n)* is the probability that the apparently less-good arm is the best arm

The optimal allocation of trials

- You can minimize *L* by choosing *n=n** such that:

$$n^* \simeq b^2 \ln\left[\frac{N^2}{8\pi b^4 \ln N^2}\right]$$

$$b=\sigma/\left(\mu-\mu'\right)$$

A Quasi-Realizable Strategy

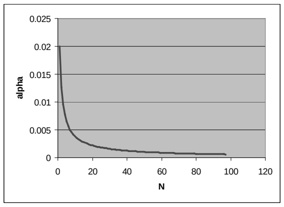
- Let’s say we take *N* trials, allocate *n** to each arm, and the remaining *N-2n** to the observed best arm
- We could then iterate, allocating another *N* trials, using a similar strategy to correct towards the appropriate value of *n** for *2N*
- *Etc.*

How does *n** vary as a percentage of *N?*

- As a percentage:

$$\alpha=\frac{n^*}{N}\simeq\frac{b^2}{N}\ln\left[\frac{N^2}{8\pi b^4 \ln N^2}\right]$$

- You should allocate an exponentially decreasing percentage of trials to the observed worst



For a k-armed bandit

- A similar result holds:

$$\alpha \simeq \frac{(r-1)b^2}{N}\ln\left[\frac{N^2}{8\pi(r-1)^2 b^4 \ln N^2}\right]$$

$$b = \sigma/(\mu-\mu_r)$$

So, we know

- Allocating increasing numbers of trials to the observed best is a near-optimal strategy
- However, that doesn’t make things entirely clear, since it’s mainly an argument about form, not details

There is a body of literature

- On *learning automata* that consider algorithms for updating action selection policies based on experience
- Consider *linear reward-penalty:*

$$\pi_{t+1}(d_t) = \pi_{t+1}(d_t)[1-\alpha]+\alpha = \pi_{t+1}(d_t)+\alpha\left[1-\pi_{t+1}(d_t)\right]$$

- Which is applied for the “correct” action, and other action probabilities are adjusted to compensate
- There is also *linear reward-inaction,* which only updates when the correct action is known to have been taken

Another approach

- Consider retaining the average reward value that you get every time a give action is taken:

$$Q_{t+1}(a) = \frac{r_1(a)+r_2(a)+r_3(a)\ldots+r_{k_a}(a)}{k_a}$$

- How should we select actions based on these values?

Greedy action selection

- Starting with all *Q* values set to equal, random values
- Always select the action with the highest *Q* value
- This is, in general, a bad idea, which leads to *premature convergence*

Softmax Action Selection

- Use the following action selection probabilities

$$\pi_t(a) = \frac{\exp\left(Q_t(a)/\tau_t\right)}{\sum_b \exp\left(Q_t(b)/\tau_t\right)}$$

- Where t is a temperature parameter that can be tuned downward, towards increasingly greedy strategies

In general, we update values incrementally

- Recall

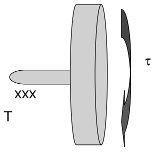
$$Q_{t+1}(a) = \frac{r_1(a) + r_2(a) + r_3(a) ... + r_{k_a}(a)}{k_a}$$

$$= Q_t(a) + \frac{1}{k_a + 1}\left[r_{k_a+1} - Q_t(a)\right]$$

$$Q_{t+1}(a) = Q_t(a) + \alpha(k_a)\left[r_{k_a+1} - Q_t(a)\right]$$

Flywheels

- Are a mechanical smoothing filter



$$T = C\omega$$

$$J\dot{\omega} = \tau - C\omega$$

$$\frac{J}{C}\dot{T} = \tau - T$$

$$T^{t+1} - T^t = \alpha\left(\tau^t - T^t\right)$$

Online Averaging With a "Flywheel Equation"

- Consider a flywheel with a noisy input torque t, and a output torque T
- A discrete time model of the flywheel is

$$T^{t+1} = T^t\left(1 - \alpha\right) + \alpha\tau^t$$

- Where α is inversely related to mass

At Steady State

- The mean value of *T* is the same as that of *t* but with lower noise
- Low mass (high α) flywheels have fast transients, and less "smoothing"
- High mass (low α) flywheels have slow transients, but smooth, steady output
- We can use a flywheel to estimate the average of the *Q* values

Stochastic Approximation Theory Sez:

- Flywheel-like updates of *Q* converge with probability one if

$$\sum_{k_a=1}^{\infty}\alpha(k_a) = \infty, \ \sum_{k_a=1}^{\infty}\left[\alpha(k_a)\right]^2 < \infty$$

- Which is true for online averaging, but not for flywheel updates with fixed α
- However, we often use fixed α, to cope with nonstationarities, particularly in the dynamic problems we'll cope with later