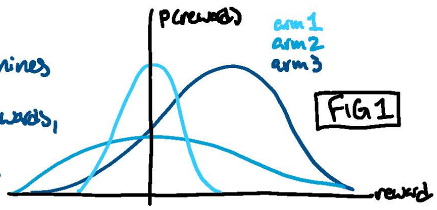
Determining a policy.

Thompson sampling (TS) achieves lowest regret bound for multi-armed bandit problems. [Agrawal & Goyal 2012, plus lots of others]

We can use Thompson sampling by framing our maximizing-information objective as a multi-armed bandit problem.

TS: does probability matching.

Say you have 3 slot machines that output different rewards, with different distributions.



Given a distribution of reward probabilities obtained by sampling, we choose which arm to pull by sampling from each distribution and choosing the maximum reward. This achieves the theoretical lower bound on regret (Lai & Robbins). I.e., this is the ideal balance between exploration & exploitation.

Regret definition:

$q(a) = E[R \mid A = a]$

$V_* = q(a^*) = \max_{a \in A} q(a)$

regret $l_t = E[V^* - q(A_t)]$

total regret $L_t = E[\sum_{t=1}^{T} V^* - q(A_t)]$

regret is the difference between your action and the optimal action.

There are 2 major differences between our problem and the (contextual) multi-armed bandit

1) We want to maximize information and basically don't care about collecting reward
2) The animal decides what states to visit, not us.

For 1, we can simply replace all the reward terms in the bandit problem with information (entropy) instead.

For 2, we notice that our aim is to perturb the animal as little as possible to avoid unphysiological behavior. Because of this, the states the worm visits will follow a mostly consistent distribution. When the worm is in a state, we can sample (perturb) there. Thus for a given time length T, the maximum number of times we can sample state s (= pull arm s) is $T * p(s)$. The number of pulls for an arm s has an upper bound, but otherwise we can set sampling probabilities at each state to the probabilities dictated by TS.

eg) from Fig 1, say we have $p(s_1 \max) = .5$, $p(s_2 \max) = .25$, $p(s_3 \max) = .25$. Then say probs of being in each state are $p(s_1) = .4$, $p(s_2) = .4$, $p(s_3) = .2$

Say we have a maximum proportion allowed for sampling $l = 0.2$.

Then, our sampling probs are $\xi(s_1) = .2\, p(s_1 \max) = .1$

$\xi(s_2) = .05$

$\xi(s_3) = .05$.

None of these exceed the allowed maximum for each state.

In the case that a $\xi(s_i) > p(s_i)$, we set $\xi(s_i) = p(s_i)$ and recast

$l' = l - p(s_i)$

$p'(s_j \max) = \frac{p(s_j \max)}{\sum_{k \neq i} p(s_k \max)} \quad \forall j \neq i$.

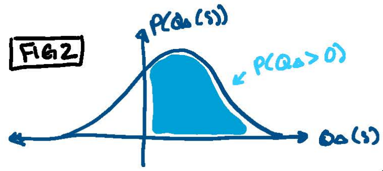
Then we reevaluate $\xi(s)$'s. For instance, if the probabilities $p(s_i)$ had been $p(s_1) = .05$, $p(s_2) = .5$, $p(s_3) = .45$, then we would have gotten $\xi(s_1) = .05$, $\xi(s_2) = .075$, $\xi(s_3) = .075$.

Then the light-on proportions would be half of this, since we want equal light-on and light-off data to form our Qs.

So given entropy distributions on states $h(s)$ [more accurately $h(p(a|s))$ from notes March 6], we have what should be the optimal policy with our assumptions.

However, now we need to get those entropy distributions.

FIG 2



From last time, we have $h(p(a|s))$ by integrating over the distribution $p(Q_\Delta(s)>0)$.

Let $p(Q_\Delta(s)>0) = \bar{\theta}_s$. $h(p(a|s)) = -\bar{\theta}_s \ln \bar{\theta}_s - (1-\bar{\theta}_s)\ln(1-\bar{\theta}_s)$. What is $p(\theta_s)$? If we have that, we can get $p(h(p(a|s)))$.

Idea:

This is a coin flip problem. $\theta_s \in (0,1)$. We have a mean from FIG 2 and could model θ_s if we think of a binomial dist where $Bin(n,p)$ uses $p = \bar{\theta}_s = p(Q_\Delta(s)>0)$ and n is the number of light on samples at state s. The coin flip here is whether $Q_\Delta > 0$, a binary outcome. Then,

$p(n_s\theta_s) \sim Bin(n_s\theta_s, n_s, \bar{\theta}_s)$ where $n_s\theta_s \in \mathbb{N}$.

We can rescale and normalize to obtain $p(\theta_s)$. The final step is to get $p(h(p(a|s)))$.

Note: sampling here will still be discrete because we used binomials.

Variable transform: from $p(\theta)$ to $p(-\theta\ln\theta - (1-\theta)\ln(1-\theta))$, $\theta \in [0,1]$.

Let $g(\theta) = -\theta\ln\theta - (1-\theta)\ln(1-\theta)$. $g(\theta) = g(1-\theta)$. Also, since $p(\theta)$ is a scaled binomial, for every value of $p(\theta)$ there's a matching value for $p(1-\theta)$; that is, θ exists at points uniformly distributed between 0 & 1.

Considering the goal is simply to get $p(g(\theta))$, we can store probabilities by:

PSEUDOCODE

vector of p(g(θ)) → prob_g; vector of g values → gs; valid θ from Bin distrib

```
prob_g ← zeros(round(len(θ)/2))
gs ← zeros(len(prob_g))
for i in {0, 1, ..., len(gs)}:
    if θ_i != 0.5:
        prob_g[i] ← p(θ_i) + p(θ_{len(θ)-i})
    else:
        prob_g[i] ← p(θ_i)
    gs[i] ← -θ_i lnθ_i - (1-θ_i)ln(1-θ_i)
```

Then we can sample from the vector gs with prob_g probabilities.

These samples can then be used in the Thompson sampling scheme described above.