Exploration/exploitation strategies

Monday, March 8, 2021 10:03 AM

Target: figure out a set of strategies to use for the entropy case.

Differences between the usual RL paradigm and us:

- we are looking for entropy, not reward.

Usually, high-reward states stay high-reward. For us, high-entropy states will hopefully turn into low-entropy ones as we sample more.

Normally, as $t\to\infty$, $\hat{\mu}_{s_i}\to\mu_{s_i}$. ($\hat{\mu}_{s_i}$: mean of samples, μ_i: population mean)

For us, as $t\to\infty$, $h(\pi(a|s_i))\to 0$.

CLT gives us:

Let $\{X_n : 1\le n\}$ be a sequence of iid RVs.

$S_n=\sum_{i=1}^{n}X_i \; \forall n\ge 1$ with $E[S_n]=\sum_{i=1}^{n}E[X_i]$ & $Var[S_n]=\sum_{i=1}^{n}Var[X_i]$.

$S_n^*=\frac{S_n-n\mu}{\sigma\sqrt{n}}\left(=n\cdot\frac{\hat{\mu}-\mu}{\sigma\sqrt{n}}\right)$ for $\mu=E[X_i]$ & $\sigma^2=Var[X_i]$

If $S_n^*\sim F_n(t)$ then $F_n(t)\to\Phi(t)$ as $n\to\infty$.

In other words,

$\hat{\mu}\sim G_n(t)\to N(\mu,\frac{\sigma^2}{n})$

The more samples we get, the distribution of $\hat{\mu}$ becomes normal & narrower $\sim\frac{1}{\sqrt{n}}$.

So our estimate of $Q_\Delta(s_i)$ gets closer to the true value as $\frac{1}{\sqrt{n}}$, which means the probability approaches what?

$P(Q_\Delta(s_i)>0)=1-\Phi\left(\frac{-\hat{\mu}}{\hat{\sigma}/\sqrt{n}}\right)$ where Φ is the cdf of the standard normal. $\hat{\mu},\hat{\sigma}$ estimated mean, std of $Q_\Delta(s_i)$ from sample.

$$=1-\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\frac{-\hat{\mu}}{\hat{\sigma}/\sqrt{n}}}e^{-t^2/2}dt=1-\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{-\frac{\hat{\mu}\sqrt{n}}{\hat{\sigma}}}e^{-t^2/2}dt$$

As $n\to\infty$, the term $-\frac{\hat{\mu}\sqrt{n}}{\hat{\sigma}}$ becomes $\pm\infty$ as $n\to\infty$ and $\hat{\mu}\to\mu$ depending on the sign of μ. When $\mu>0$, $\lim_{n\to\infty}P(Q_\Delta(s_i)>0)\to 1$ and when $\mu<0$, $\lim_{n\to\infty}P(Q_\Delta(s_i)>0)\to 0$.

This is all kind of obvious. What am I trying to do? For true $Q_\Delta(s_i)\neq 0$,

$\lim_{n\to\infty}h(\pi(a|s_i))=\lim_{n\to\infty}[-p(Q_\Delta(s_i)>0)\log p(Q_\Delta(s_i)>0)-(1-p(Q_\Delta(s_i)>0))\log(1-p(Q_\Delta(s_i)>0))]$

$=0$ (using $\lim_{x\to 0}x\log x=0$.)

Usually RL wants estimates of value $\hat{q}(s_i)$ for state s_i to $\to q(s_i)$, true value asap

We want $h(\pi(a|s_i))\to 0$ asap.

$\hat{q}(s_i)\to q(s_i)$ as $\frac{1}{\sqrt{n}}$.

Can I get a similar expression for h?

$h(\pi(a|s_i))\to 0$ as ?

David Silver exploration/exploitation lecture. (9 in RL course)

1 intro
2 multiarmed bandits
3 contextual bandits
4 MDPs

Principles:

- Naïve exploration
  - Add noise to greedy policy (ϵ-greedy)
- Optimistic init
  - Optimism in the face of uncertainty
  - Need estimates of uncertainty on value
- Information state search
  - Consider agent's info as part of state
  - Look ahead to see how information helps reward
    - Correct but computationally difficult
- Probability matching
  - Select action according to probability they are best

Multiarmed bandit case.

$\langle A, R\rangle$ no state.

Regret definition:

$q(a)=E[R|A=a]$

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

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

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

Maximizing cumulative reward ≡ minimizing total regret.

What about for our case? The optimal action is the one with the most information. They compare to best-case scenario, where optimal action is taken all the time. What's our best-case? Best states to sample would still be ones with most information. The most uncertainty. It's not quite the same as choosing actions, is it? It's choosing states. Or rather, the action is to sample a state. Some states you're sure about and some you're not.

What is the MAB case? The goal there is most reward. The goal here is most information. I guess the equivalent would be knowing exactly what the distributions look like. The estimate is $\hat{h}(s_i)$ compared to some final "true" $h(s_i)$. Then I'd only be sampling around the border. So maybe it's like – I don't actually get the information from a state I choose to sample at. I'm getting an improvement on the estimate of the actual information.

One regret step is $l_t=E[V^*-q(a_t)]=E[q(a^*_t)-q(a_t)]$.

Our regret step is $l_t=E[h^*-h(p(a))]$

$l_t=\sum_{s_i}p(s_i)\xi(s_i)\hat{h}(p(a|s_i))-\sum_{s_i}p(s_i)\xi(s_i)h^*(p(a|s_i))$ (Problem is, we don't have this value here. We have a current guess)

In the MAB case, say I have two states that are visited equally and I can chose when to sample if my sampling is limited. One is certain, another is not.

I can choose between two actions. Which gives me the most information? And what's the best case? The dists are [sketch] & [sketch] so at the end, I want to be only picking state 1. I approach the generating distribution (μ,σ^2).

$l_t=E[h(a^*)-h(a_t)]$ okay maybe I just need to watch rest of lecture.

The count $N_t(a)$ is expected # of selections for a

Gap Δ_a is $V_*-q(a)$.

Real time ≠ math time

my case is just like choosing actions, except with a cap on how often I can chose them. and I choose states instead of actions.

Regret includes both:

$L_t=E[\sum_{\tau=1}^{t}V_*-q(A_\tau)]$

$=\sum_{a\in A}E[N_t(a)](V_*-q(a))$

$=\sum_{a\in A}E[N_t(a)]\Delta_a$.

$L_t=E[\sum_{\tau=1}^{t}h_*-h(s_i)]$ (h_*: info from best state; the states we pick are the actions)

$=\sum_{s\in S}E[N_t(s)\xi(s)](h_*-h(s))$

$=\sum_{s\in S}E[N_t(s)\xi(s)]\Delta_s$ ← information gap btwn our state & best state.

Problem is we don't know the gap. Normally, we only don't know h^*. Here, we don't really know $h(s)$ either. But I suppose that's true for the first case, where $q(s)$ is just a sample.

Need to recall the relationship between a sample (state-picking) and the $h(s)$ returned. The sample becomes part of a pool $\{q_\Delta^{(t)}(s)\}_{t=1,\ldots,T}$ (kind of) and we go $f(D):(\hat{\mu}_s,\hat{\sigma}_s^2)$ (for each s).

Then $h(p(a|s))=-p(Q_\Delta(s)>0)\log p(Q_\Delta(s)>0)-p(Q_\Delta(s)\le 0)\log p(Q_\Delta(s)\le 0)$

$=-\left(1-\Phi\left(\frac{-\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\right)\log\left(1-\Phi\left(\frac{-\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\right)-\Phi\left(\frac{-\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\log\left(\Phi\left(\frac{-\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\right)$

$=-\Phi\left(\frac{\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\log\left(\Phi\left(\frac{\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\right)-\Phi\left(\frac{-\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\log\left(\Phi\left(\frac{-\hat{\mu}_s}{\hat{\sigma}_s/\sqrt{n}}\right)\right)$

So every point $q_\Delta(s)\to$ some $\Delta h(s)$.

Normally:

$Q_t(a)=\frac{1}{N_t(a)}\sum_{t=1}^{T}\mathbb{1}_{A_t=a}R_t$. We estimate with the mean.

Instead, we have

$H_t(s)=g(f(D))$.

Greedy has linear total regret.

One soln: optimistic initialization. Initialize values to maximum possible, then act greedily. Also has linear total regret. Can get locked out of actions forever.

Compare to ϵ-greedy, the opposite of greedy sort of: Continue to explore forever.

Ensures minimum regret $l_t\ge\frac{\epsilon}{A}\sum_{a\in A}\Delta_a$ (every step, you at least have the average regret ϵ of the time.)

Softmax exploration also has linear total regret.

Can get sublinear regret, however, with a simple fix.

Pick a decay schedule:

$d=\min_{a|\Delta_a>0}\Delta_a$, $c>0$

$\epsilon_t=\min\{1,\frac{c|A|}{d^2t}\}$

decay so that ϵ scales as inverse of time, multiplied by the smallest regret². What? Gap between best & second best. Many other schedules like this probably work.

This requires knowledge of the gaps, however. When gaps are small, explore more. When gaps are large, explore less.

Gets logarithmic regret.

Now we want to do this without knowing R.

First: note that there's a lower bound on regret (it is logarithmic). What does it depend on? Similarity between best arm & other arms.

Hard problems have similar options (with diff means) – lots of noise but one is still better.

Formally, we describe this as Δ_a and the similarity in distributions $D_{KL}(R^a||R^{a^*})$.

What is our hardest case? Lots of noise in the $H(s)$. (vs. $Q(a)$.) In the Q's, the Q distribution itself is very wide. In the $H(s)$ case? [sketches: hard q, estimates; $H(s)$; in order for the H's to be wide, the probs?; variability from both $\hat{\mu}$ and $\hat{\sigma}$; $P(Q_\Delta)$, Q_Δ; $H(s_i)$, Prob] What distributions must be the same then? The $H(s)$ estimate, which means the distributions of both $\hat{\mu}$ and $\hat{\sigma}$. Which in turn means the distribution of $Q_\Delta(s)$ versus $Q^*_\Delta(s)$.

The lower bound depends on similarity between the most informative (boundary) state and the other states.

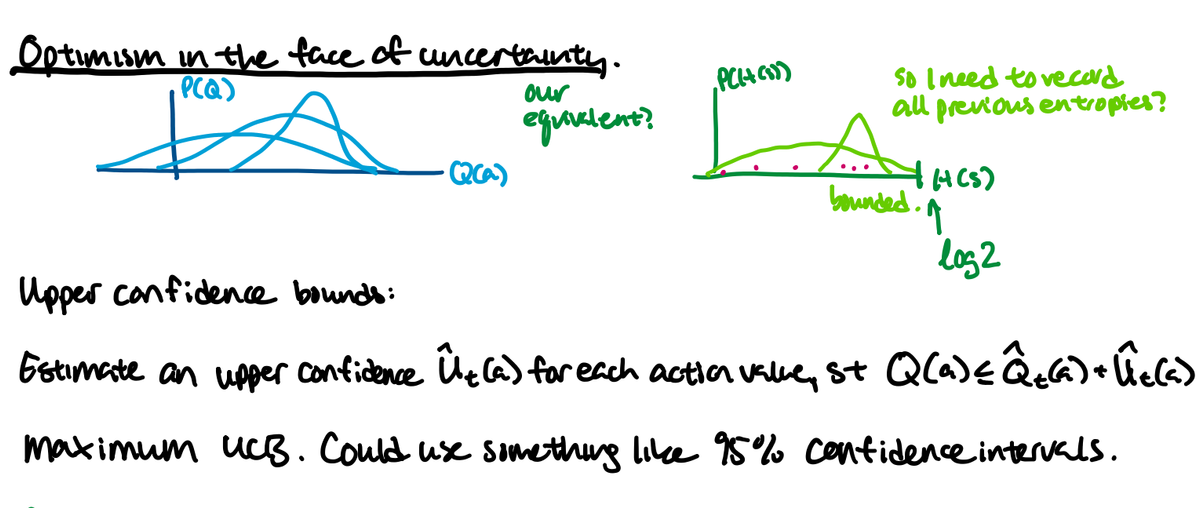
$D_{KL}(Q_\Delta(s)||Q_\Delta^*(s))$

Then theorem (Lai & Robbins): asymptotic total regret is at least logarithmic in the number of steps.

$\lim_{t\to\infty}L_t\ge\log t\sum_{a|\Delta_a>0}\frac{\Delta_a}{D_{KL}(R^a||R^{a^*})}$ → $\propto$ gap, $\propto\frac{1}{D_{KL}}$.

("fundamental to bandits" could look this up later) (bigger D_{KL}, smaller scaling of $\log t$. Add up all gaps. And bigger gaps mean more regret.)

Optimism in the face of uncertainty.



Upper confidence bounds:

Estimate an upper confidence $\hat{U}_t(a)$ for each action value, st $Q(a)\le\hat{Q}_t(a)+\hat{U}_t(a)$ and choose maximum UCB. Could use something like 95% confidence intervals.

Potential issue would be that all the H's depend on previous H's. Although they don't need to, I suppose. Would choose between many low-quality estimates or a few high-quality ones.

That's interesting – what should I do here? I get $h(s)$ estimates after every episode. But the estimates are $g(f(D))$ and depend on $f(D)=(\hat{\mu},\hat{\sigma})$. The distribution guess gets better after more data are collected. Just like in ensembles, the tradeoff is between using all the data and ensuring a fit or using parts of the data (each of which should have some underlying distribution) and lowering the variance in prediction. Sounds like I should be making models with parts of data and not all of it.

The problem is, I may not collect data from places I'm sure of anymore.

Hoeffding's inequality: general theorem.

Let $X_1,\ldots,X_t$ iid $\in[0,1]$. Let $\bar{X}$ be the sample mean. Then

$P[E[X]>\bar{X}_t+u]\le e^{-2tu^2}$ (prob that we're at least u too low.)

We can apply this

$P[Q(a)>\hat{Q}_t(a)+U_t(a)]\le e^{-2N_t(a)U_t(a)^2}$.

pick prob that true value exceeds UCB & solve for $U_t(a)$.

$e^{-2N_t(a)U_t(a)^2}=p$

$U_t(a)=\sqrt{\frac{-\log p}{2N_t(a)}}$

Reduce p as time goes on, eg $p=t^{-4}$. Ensures optimal action as $t\to\infty$.

This leads to UCB1:

$A_t=\underset{a\in A}{\operatorname{argmax}}\left(Q_t(a)+\sqrt{\frac{2\log t}{N_t(a)}}\right)$ $\log t^2=-\log t^{-2}$

This achieves asymptotic logarithmic total regret

$\lim_{t\to\infty}L_t\le 8\log t\sum_{a|\Delta_a>0}\Delta_a$.

Bayesian bandits.

Consider dist $p(Q|\vec{w})$ over action value fn w/ parameter $\vec{w}$. They use posterior to guide exploration. Can be used w/ Bayesian UCB or probability matching (Thompson sampling).

Prob matching:

$\pi(a|h_t)=P[Q(a)=\max_{a'}Q(a')|R_1,\ldots,R_{t-1}]$

Thompson sampling is asymptotically optimal (for Bernoulli bandits). Just sample from posterior and pick best.

Information value

If we can quantify the value of information, then we can make exactly the reward-info tradeoff. The value depends on the budget of actions.

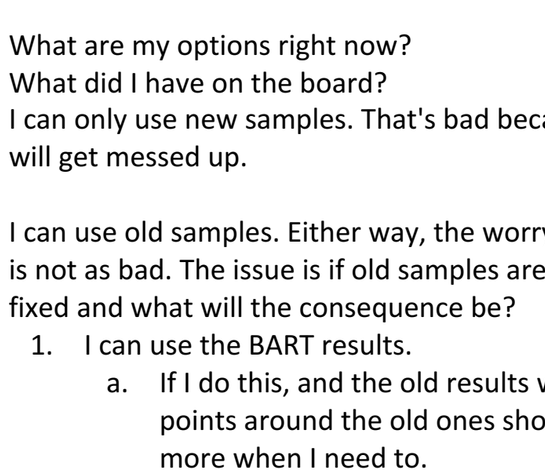
How does Thompson sampling work here? I have $h(s)$ for all states. I explore based on the estimated reward, but for me it's estimated entropies. I need to make distributions for entropies as well? Upper bound at ln2, lower at 0. [sketch: P(H), H, ln2] Sample from β. How would that work?

I'd want to pick states to sample: top states. Entropy samples: [sketch] Some will randomly be higher or lower. It would be like randomly drawing and seeing which states land in the top. And each time I take a step I'll use the next sample.

That seems all right. What else do I need to figure out? Independent $h(s)$ values but no more sampling certain states. No data from those states for the next model. I think I need to add points in. Have a minimum number of points per state?

Couple things to work out for implementation.

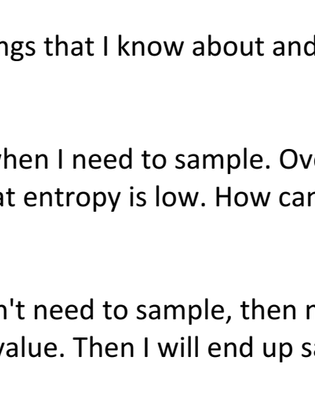
- If I'm sampling h's, $h(p(a|s))$, can I turn it into a distribution based on $p(a|s)$?



Let $\theta=p(Q_\Delta>0)$

Let $h(\theta)=-\theta\ln\theta-(1-\theta)\ln(1-\theta)$

I want $P(h(\theta))$.



1. $p(\theta|\{Q_\Delta\})$? Options: beta, binom.

What are my options right now?
What did I have on the board?
I can only use new samples. That's bad because I'll never sample things that I know about and the model will get messed up.

I can use old samples. Either way, the worry is that I won't sample when I need to sample. Oversampling is not as bad. The issue is if old samples are incorrectly confident that entropy is low. How can that be fixed and what will the consequence be?

1. I can use the BART results.
   a. If I do this, and the old results were really sure that I don't need to sample, then new data points around the old ones should change this entropy value. Then I will end up sampling more when I need to.
2. I can not use the BART results.
   a. If I do this, then I'll keep old and incorrectly confident data. Then I'll never sample again at those states. This is definitely the worse of the two.

That means I should be using the first option. I'll use old data for new BART estimates and use the new model outputs at the old states.

How do I want to do this? I'd prefer to randomly sample the old data. How do I decide how much old data to use? Also, why was this conclusion so hard to arrive at? Wonder what was tripping me up. Maybe it was just an afternoon slump.

I can pick a number of data points to use. How do I choose? Could pick the median number of points or something. That's completely random.

What are some options? I can go through it again like I did above. Median, mean, max. Not the max. Or I can set a fixed number. A number proportional to the distribution, perhaps. Maybe something like a random sample of the previous data, to some reasonable minimum. Maybe like 10% of the expected number of samples over five minutes. Five minutes gives, what, 9000 samples. 9000/288=31.25 expected samples if evenly distributed. I know they're very much not evenly distributed. Maybe I'll just set the number at 30.

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.

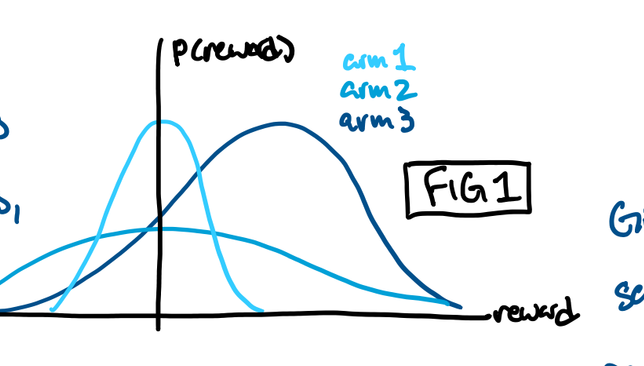


FIG 1

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:

$q(a)=E[R|A=a]$

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

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

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

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

$l'=l-p(s_i)$

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

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 Q_Δ.

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.

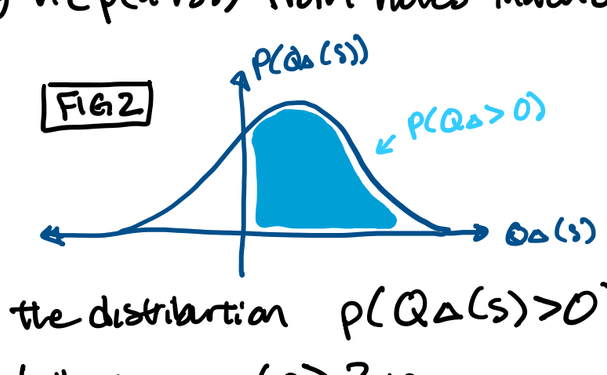


FIG 2

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

```
prob_g ← zeros(round(len(θ)/2))      # vector of p(g(θ))
gs ← zeros(len(prob_g))               # vector of g values; valid θ from Bin distrib
for i in {0, 1, ..., len(gs)}:
    if θi != 0.5:
        prob_g[i] ← p(θi) + p(θ_len(θ)-i)
    else:
        prob_g[i] ← p(θi)
    gs[i] ← -θi lnθi - (1-θi)ln(1-θi)
```

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.