Date: Thursday, April 18, 2019

Recap

Note: Optimal mean or proportion

Method	Exploration / Exploitation	Optimal found?
A/B testing	100% Exploration	Yes — Through testing
Greedy	100% Exploitation	No — Local optimal
ϵ-greedy	Explore at rate ϵ	Yes — eventually
Softmax	Explore at softmax Prob. rate	Yes — better @ ranking than ϵ-greedy

Note: So far, greedy, ϵ-greedy, and softmax look at the mean/proportion at each round only!
↳ There is no concern for variability or uncertainty of the reward.

- We can make optimization more efficient by accounting for the spread or variability of the reward distributions at each round.

  Two methods:
  1) Upper confidence bound (UCB)
  2) Probability matching / Bayesian Sampling

[Slide is wrong, correction:]

$$\text{UCB: } \arg\max_k \left(r_{kt} + \sqrt{\frac{2\log(t)}{n_{kt}}} \right)$$

So for UCB, instead of just updating r_{kt} as in the previous methods, we now update $r_{kt} + \sqrt{\frac{2\log(t)}{n_{kt}}}$ at each round.

UCB in practice

for $t = 1, \dots, T$

Choose arm with $\underset{k}{\arg\max} \left(r_{kt} + \sqrt{\frac{2\log(t)}{n_{kt}}} \right)$

Return arm with highest value

↳ This is greedy in nature, and can be made more efficient by relying on the entire distribution of rewards.

Randomized probability matching:

reward of arm k up to time t (estimated)

Value of interest: $P(r_{kt}$ is optimal$)$

$$P\left(r_{kt} = \max_{j \in 1,\dots,k} \{r_{jt}\}\right) = P_{kt}$$

we allocate units to arm k with probability P_{kt}.

The probability we wrote up (P_{kt}) can be thought of in the following way:

$R = (r_1, r_2, \dots, r_k)$ = True rewards for each arm

$\pi(R)$ = Prior distribution on R

a) If r_j is a proportion, use a beta distribution (or uniform $(0,1)$)

Assume rewards are independent and each distributed as $U(0,1)$, then

$$\pi(R) = \prod_{j=1}^{k} 1 = 1$$

b) If r_j is continous, use a $N(\mu_j, 1)$ distribution where μ_j = the initialized reward for arm j.

$$\pi(R) = \prod_{j=1}^{k} \frac{1}{\sqrt{2\pi}} \exp\left\{ \frac{(r_j - \mu_j)^2}{2} \right\}$$

(Product of normals)

Data: The reward up to time t. That is $r_{1t}, \dots, r_{kt}$

$r_t = (r_{1t}, \dots, r_{kt})$ ← estimated values up to round t same as what is done in ϵ-greedy.

Generating process: $f(r_t | R)$

rewards up to time t

true rewards

Our value of interest can be thought of as the posterior distribution of R given the rewards up to time t:

$$(*) \quad \boxed{P_{kt} = P(r_k = \max\{r_1, r_2, \dots, r_k\} \mid (r_{1t}, \dots, r_{kt}))}$$

To get to $(*)$ is by calculating

$$P(R \mid r_{1t}, \dots, r_{kt}) \propto \pi(R) f(r_{1t}, \dots, r_{kt} \mid R)$$

Once we get ↗, we can then simulate say 1000 times and calculate the proportion of times r_k was maximum as our value for P_{kt}.

1) rewards are proportions → use dirichlet-multinomial model
(Dirichlet posterior)

2) rewards are continous → use normal-normal model
(Normal Posterior)

For proportions:

Algorithm: Initialize and calculate rewards
↳ Normalize to probabilities for $t = 1, 2, \dots, T$

Calculate Dirichlet posterior parameters for $P_{1t}, \dots, P_{kt}$.

Simulate 1000 samples from Dirichlet and calculate

$P_{kt} = P(r_k \text{ is max at time } t)$

Pull arms with probabilities $P_{1t}, \dots, P_{kt}$ and update prob.

For continous values:

Algorithm: Initialize and calculate rewards for $t = 1, 2, \ldots, T$

Calculate Normal posterior parameters for $r_{1t}, \ldots, r_{kt}$.

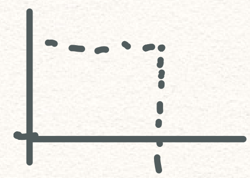
Simulate 1000 samples from Normal and calculate

$p_{kt} = P(r_k \text{ is max at time } t)$

Pull arms with probabilities $p_{1t}, \ldots, p_{kt}$ and update prob.

$$\pi(R) = \prod_{j=1}^{k} f(r_j) = \prod_{j=1}^{k} 1$$

$x \sim U(0,1)$, $f(x) = 1$



RPM (Randomized Prob. Matching)

- has been found to be the most efficient (in terms of iterations needed) to identify the optimal condition.
- This is viewed by looking at the optimality probability plots.

···· is optimal and converges to 1

— converges to 0