Bandit Learning with Positive Externalities

Virag Shah, Jose Blanchet, Ramesh Johari

Management Science and Engg Department, Stanford University

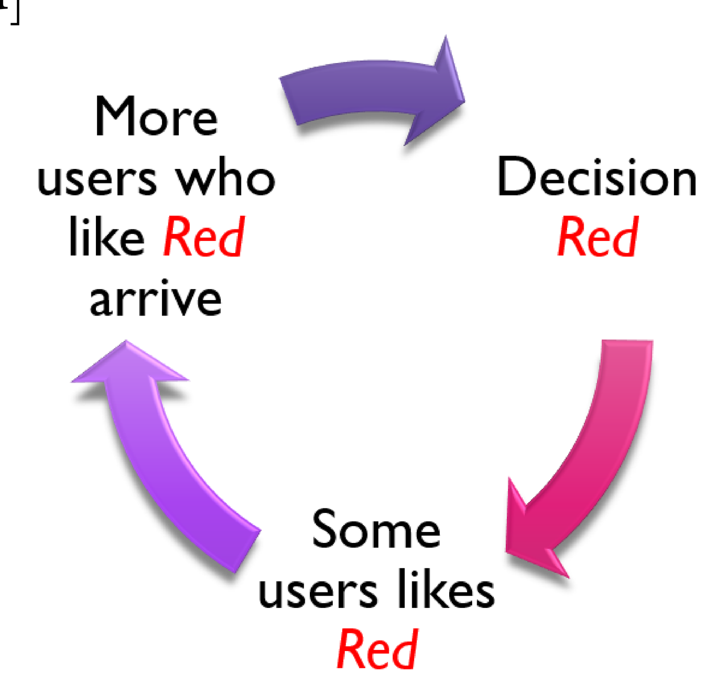
Positive Externalities in Online Platforms

ebay Etsy upwork airbnb Thumbtack urbansitter STITCH FIX instacart

- Learning algorithms are often used to recommend alternatives to users.
- *Common assumption*: arrivals not influenced by decisions.
- *Reality*: **Positive Externalities / Self-Reinforcement / Network Effects** – a positive experience attracts more users of the same type

What could go wrong?

- Suppose Blue users like Blue items (but not Red items)
- Suppose Red users like Red items (but not Blue items)
- Suppose $\mathbb{E}$[Blue-Blue match reward] > $\mathbb{E}$[Red-Red match reward]



- Successful Red-Red matches made early on may trigger more Red user arrivals.
- So the platform might learn to prefer Red-Red matches even if it is suboptimal!

Main Insights from our Results

- There is a cost to being optimistic in the face of uncertainty as initial mistakes are amplified. UCB algorithm, in particular, fails miserably.
- It is possible to prevent the snowballing arising from positive externalities by structuring the exploration procedure well.
- Once enough evidence is gathered, one may even use the externalities to shift the arrivals to a profit-maximizing population.

Model

Standard bandit setting:

- m: Number of arms (items)
- T : Time horizon; one user arrives per time step
- μ_a: Expected reward when arm a pulled (Bernoulli)
- a^*: best arm
- $T_a(t)$: number of times arm a pulled up to time t
- $S_a(t)$: total reward at arm a up to time t
- Goal: maximize expected total reward (R_T).

Positive externalities:

- Let θ_a be initial "bias" of arm a.
- We assume the user arriving at time t *likes* arm a independently with probability:

$$\lambda_a(t) = \frac{f(\theta_a + S_a(t))}{\Sigma_b f(\theta_b + S_b(t))}.$$

- f is the *externality function*. We consider $f(x) = x^\alpha$, $\alpha > 0$. α determines the strength of the positive externality.
- $\mathbb{P}$(reward at $t|$ arm a pulled) $= \mu_a$ if user t likes a, otherwise zero.

Main Results

Suboptimality of UCB Algorithm:

Definition: UCB(γ) algorithm, at time t, pulls the arm a with largest:

$$\text{empirical mean reward up to } t + \sqrt{\frac{\gamma \ln t}{T_a(t-1)}}.$$

Theorem

UCB(γ) exhibits $O(T)$ regret for each $\gamma > 0$. In fact,

$$\lim_{T\to\infty} \mathbb{P}(S_{a^*}(T\ ) = 0) > 0.$$

Optimal Algorithm:

Definition: The *Balanced-Exploration* (BE) algorithm is as follows:

Suppose $w_k = \ln\ln k$ for each $k \geq 1$. Fix $\tau = w_T \ln T$.

- For $t \leq \tau$, pull the arm with lowest cumulative reward $S_a(t-1)$ (ties broken at random).
- For $t > \tau$, pull the arm with highest mean reward $S_a(\tau)/T_a(\tau)$ at time τ.

Theorem

The regret of the BE algorithm is as follows:

1. If $0 < \alpha < 1$ then $\mathbb{E}[R_T] = \tilde{O}(T^{1-\alpha} \ln^\alpha T)$.
2. If $\alpha = 1$ then $\mathbb{E}[R_T] = \tilde{O}(\ln^2 T)$.
3. If $\alpha > 1$ then $\mathbb{E}[R_T] = \tilde{O}(\ln^\alpha T)$.

Further, there exists no algorithm which can achieve better asymptotic performance than above.

- We also provide an adaptive variant of BE, called BE-AE, which successively drops suboptimal arms.