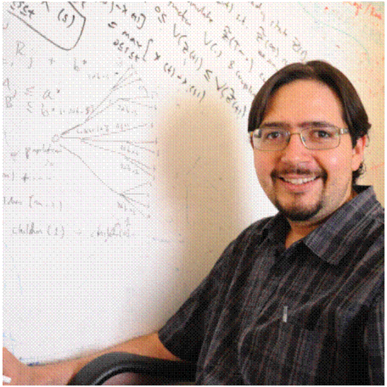
Bandit learning with positive externalities

Virag Shah | Jose Blanchet | *Ramesh Johari*
Stanford University
rjohari@stanford.edu

March 28, 2018

Collaborators

Virag Shah



Jose Blanchet

Motivation: Recommendations in online platforms

- Learning algorithms used to recommend alternatives to users.
- *Common assumption*: arrivals not influenced by decisions.

Motivation: Recommendations in online platforms

- Learning algorithms used to recommend alternatives to users.
- *Common assumption*: arrivals not influenced by decisions.
- This talk is about learning with *positive externalities*: A positive experience attracts more users of the same type

A simple setting

Suppose there are two types of users of a platform:

- blue users like blue items (but not red items)
- red users like red items (but not blue items)

What could go wrong?

Positive rewards can be self-reinforcing:

What could go wrong?

Positive rewards can be self-reinforcing:

Suppose a red-red match made early on. Then:

What could go wrong?

Positive rewards can be self-reinforcing:

Suppose a red-red match made early on. Then:

- more red type users likely to arrive, and

What could go wrong?

Positive rewards can be self-reinforcing:

Suppose a red-red match made early on. Then:

- more red type users likely to arrive, and
- blue items less likely to generate positive reward.

What could go wrong?

Positive rewards can be self-reinforcing:

Suppose a red-red match made early on. Then:

- more red type users likely to arrive, and
- blue items less likely to generate positive reward.

So the platform might learn to prefer red-red matches.

What could go wrong?

Positive rewards can be self-reinforcing:

Suppose a red-red match made early on. Then:

- more red type users likely to arrive, and
- blue items less likely to generate positive reward.

So the platform might learn to prefer red-red matches.

If $\mathbb{E}[$ blue-blue match reward $] > \mathbb{E}[$ red-red match reward $]$,
then this is a suboptimal outcome.

Our results: Summary

- In the presence of positive exernalities, optimal algorithms for the classical multiarmed bandit can fail spectacularly.
- In our model, we instead develop an optimal algorithm via *balanced exploration*.

Model

The model: Standard bandit

- m: Number of arms (“items”)

The model: Standard bandit

- m: Number of arms (“items”)
- T: (Discrete) time horizon; one user arrives per time step

The model: Standard bandit

- m: Number of arms (“items”)
- T: (Discrete) time horizon; one user arrives per time step
- μ_a: probability of unit reward when arm a pulled (Bernoulli)

The model: Standard bandit

- m: Number of arms (“items”)
- T: (Discrete) time horizon; one user arrives per time step
- μ_a: probability of unit reward when arm a pulled (Bernoulli)
- a^*: best arm (for simplicity, assume unique)

The model: Standard bandit

- m: Number of arms (“items”)
- T: (Discrete) time horizon; one user arrives per time step
- μ_a: probability of unit reward when arm a pulled (Bernoulli)
- a^*: best arm (for simplicity, assume unique)
- $T_a(t)$: number of times arm a pulled (“recommended”) up to time t

The model: Standard bandit

- m: Number of arms (“items”)
- T: (Discrete) time horizon; one user arrives per time step
- μ_a: probability of unit reward when arm a pulled (Bernoulli)
- a^*: best arm (for simplicity, assume unique)
- $T_a(t)$: number of times arm a pulled (“recommended”) up to time t
- $S_a(t)$: number of times arm a generates reward up to time t

The model: Standard bandit

- m: Number of arms ("items")
- T: (Discrete) time horizon; one user arrives per time step
- μ_a: probability of unit reward when arm a pulled (Bernoulli)
- a^*: best arm (for simplicity, assume unique)
- $T_a(t)$: number of times arm a pulled ("recommended") up to time t
- $S_a(t)$: number of times arm a generates reward up to time t

Goal: *maximize expected total reward (ETR$_T$)*.

We study performance asymptotic in T.

The model: 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:

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

The model: 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:

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

$\mathbb{P}($reward at $t|$ arm a pulled$)$ = μ_a if user t likes a, otherwise zero.

The model: 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:

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

$\mathbb{P}($reward at $t|$ arm a pulled$)$ = μ_a if user t likes a, otherwise zero.

f is the *externality function*: it determines the strength of the positive externality.

The model: 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:

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

$\mathbb{P}($reward at $t|$ arm a pulled$)$ = μ_a if user t likes a, otherwise zero.

f is the *externality function*: it determines the strength of the positive externality.

For now let’s assume $f(x) = x$. (Generally we consider $f(x) = x^\alpha$, $\alpha \geq 0$.)

The baseline oracle

Since we study performance that is asymptotic in T, natural to consider a baseline oracle that *always chooses arm* a^*.

Proposition

The oracle earns ETR$_T^* = \mu_{a^*} T - \Omega(\ln T)$.

Intuition:
Second term comes from needing to remove any initial bias toward suboptimal arms, since:

$$\mathbb{P}(\text{user } t \text{ likes } a^*) \approx 1 - \frac{\sum_{a \neq a^*} \theta_a}{O(t) + \sum_{a \neq a^*} \theta_a}.$$

Regret

Measure performance of any algorithm against baseline oracle as *expected regret* R_T:

$$R_T = \mathsf{ETR}^*_T - \mathsf{ETR}_T.$$

Performance of benchmark policies

UCB and the standard bandit

The UCB(γ) (“upper confidence bound”) algorithm is a benchmark algorithm for the standard multiarmed bandit (MAB) problem.

At time t, UCB(γ) pulls the arm a with largest:

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

Well-known fact for standard MAB:
UCB(γ) achieves regret (against always playing a^*) of $O(\ln T)$, and this is optimal.

UCB with positive externalities

The red-blue example suggests, though, that UCB-like algorithms may not explore enough.

We show UCB(γ) has linear expected regret $R_T = \Theta(T)$.

UCB with positive externalities

The red-blue example suggests, though, that UCB-like algorithms may not explore enough.

We show UCB(γ) has linear expected regret $R_T = \Theta(T)$.

In fact, the situation is much worse:

Proposition

For UCB(γ):

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

In other words: positive probability of *never* receiving a reward on arm a^*!

(Same result holds for any super-logarithmic externality function f.)

Random explore-then-exploit

Why does UCB fail? *It stops exploring too quickly.*

A simple benchmark that explores more:

- Explore uniformly at random for some fixed time τ.
- Commit to empirical best arm at τ for rest of horizon.

This is *random explore-then-exploit* (REE).

Random explore-then-exploit

The performance of REE (with optimized τ) is somewhat better than UCB:

Proposition

For REE, $R_T = O(T^c)$, where $c < 1$.

Analysis of REE proceeds by viewing each arm as a *generalized urn process*.

Model arms as *independent* continuous-time branching processes, in which branches occur at rate $1/m$.

The *jump chain* of the combined process exactly captures positive externalities.

An optimal algorithm

What’s going wrong?

UCB and REE highlight the failure modes in this model:

- If we do not explore enough, then we risk missing the optimal arm entirely, because arriving users simply don’t like it.
- If we explore at random, then even if we identify the optimal arm, too much regret is incurred in undoing the bias on suboptimal arms.

We develop an algorithm that uses structured exploration to overcome these challenges.

Balanced exploration

The *balanced exploration* algorithm is as follows:
Fix $\tau = \Theta(\ln T)$.

- 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.

Proposition

Balanced exploration has regret $R_T = O(\ln^2 T)$.

Lower bound

To get intuition into why balanced exploration works, easiest to study the matching lower bound.

Proposition

Any policy must have expected regret $R_T = \Omega(\ln^2 T)$.

Lower bound

Intuition for lower bound:

- When each arm is explored for at least τ steps, it is *as if* the initial bias on each arm is proportional to τ.

Lower bound

Intuition for lower bound:

- When each arm is explored for at least τ steps, it is *as if* the initial bias on each arm is proportional to τ.
- Recall that if initial bias is θ, and arm a^* always pulled, then:

$$\mathbb{P}(\text{user } t \text{ likes } a^*) = 1 - \frac{\sum_{a \neq a^*} \theta_a}{\Theta(t) + \sum_{a \neq a^*} \theta_a}.$$

Lower bound

Intuition for lower bound:

- When each arm is explored for at least τ steps, it is *as if* the initial bias on each arm is proportional to τ.
- Recall that if initial bias is θ, and arm a^* always pulled, then:

$$\mathbb{P}(\text{user } t \text{ likes } a^*) = 1 - \frac{\sum_{a \neq a^*} \theta_a}{\Theta(t) + \sum_{a \neq a^*} \theta_a}.$$

- So from time $\Theta(\tau)$ onwards, if a^* correctly identified, incur regret that is $\Omega(\tau \ln T)$.

Lower bound

Intuition for lower bound:

- When each arm is explored for at least τ steps, it is *as if* the initial bias on each arm is proportional to τ.
- Recall that if initial bias is θ, and arm a^* always pulled, then:

$$\mathbb{P}(\text{user } t \text{ likes } a^*) = 1 - \frac{\sum_{a \neq a^*} \theta_a}{\Theta(t) + \sum_{a \neq a^*} \theta_a}.$$

- So from time $\Theta(\tau)$ onwards, if a^* correctly identified, incur regret that is $\Omega(\tau \ln T)$.
- Any algorithm with τ that is smaller than $O(\ln T)$ is guaranteed to incur high regret, via a standard change of measure argument; therefore we must have $\tau = \Omega(\ln T)$, and the result follows.

The strength of positive externalities

The complete picture

Suppose $f(x) = x^{\alpha}$, $\alpha \geq 0$.

	$\alpha = 0$	$0 < \alpha < 1$	$\alpha = 1$	$\alpha > 1$
UCB	$O(\ln T)$	$\Omega(T)$	$\Omega(T)$	$\Omega(T)$
REE	$O(\ln T)$	$\Omega\left(T^{1-\alpha} \ln^{\frac{\alpha}{1-\alpha}} T\right)$	$\Omega\left(T^{\frac{\mu_b}{\mu_b + \theta_{a^*} \mu_{a^*}}}\right)$	$\Omega(T)$
Balanced exp.	$O(\ln T)$	$O(T^{1-\alpha} \ln^{\alpha} T)$	$O(\ln^2 T)$	$O(\ln^{\alpha} T)$
Lower bound	$\Omega(\ln T)$	$\Omega(T^{1-\alpha} \ln^{\alpha} T)$	$\Omega(\ln^2 T)$	$\Omega(\ln^{\alpha} T)$

Conclusion

Looking ahead

In our model:

1. More general reward distributions
2. Dependence of regret on m (number of arms)

More broadly:

1. Personalization and contextual bandits
2. Other objectives