Bandits and Agents: How to incentivize exploration?

Alex Slivkins (Microsoft Research NYC)

Joint work with

Yishay Mansour (Tel Aviv University)
Vasilis Syrgkanis (MSR-NE)
Steven Wu (Penn)

EC’15, EC’16 working papers ongoing work
Motivation: recommender systems

- Watch this movie
- Dine in this restaurant
- Vacation in this resort
- Buy this product
- Drive this route
- See this doctor

- Take this medicine (medical trials)
- Use these settings (systems)
Exploration

Recommender system:
• user arrives, needs to choose a product
• receives recommendation (& extra info)
• chooses a product, leaves feedback

For common good, user population should balance
• exploration: trying out various alternatives to gather info
• exploitation: making best choices given current info

Example: coordinate via system’s recommendations.
Exploration and incentives

Recommender system:
- *agent* arrives, needs to choose a product
- receives recommendation (& extra info)
- chooses a product, leaves feedback

*Agents* make decisions based on available info & initial biases

An alternative that seems worse initially may remain unexplored because agents have no incentives to explore it!

How to incentivize agents to explore?
Exploration and incentives

How to incentivize agents to try seemingly sub-optimal actions?

“External” incentives:
- monetary payments / discounts
- promise of a higher social status
- people’s desire to experiment

prone to selection bias; not always feasible

based on agents' biases and/or system’s current info)
Exploration and incentives

How to incentivize agents to try seemingly sub-optimal actions?

Based on agents' biases and/or system's current info

“External” incentives:

- monetary payments / discounts
- promise of a higher social status
- people’s desire to experiment

Prone to selection bias; not always feasible

Alternative approach: use **information asymmetry** to create **intrinsic incentives** to follow system’s recommendations
Basic model

- K actions; T rounds
- In each round, a new agent arrives:
  - algorithm recommends an action (\& extra info)
  - agent chooses an action, reports her reward \( \in [0,1] \)
- IID rewards: distribution depends only on the chosen action
- Mean rewards are unknown; common Bayesian prior
- Objective: social welfare (= cumulative reward)

If agents follow recommendations \( \Rightarrow \) “multi-armed bandits”

“actions” = “arms”

classical model in machine learning for explore-exploit tradeoff
Basic model: BIC bandit exploration

How to account for agents’ incentives?

*Ensure that following recommendations is in their best interest!*

Recommendation algorithm is *Bayesian Incentive-Compatiable* (BIC) if

\[
\mathbb{E}_{\text{prior}}[\text{reward}(a) - \text{reward}(b) | \text{rec}_t = a] \geq 0
\]

∀round $t$, arms $a, b$

**Goal:** design *BIC* bandit algorithms to maximize performance

Can *BIC* bandit algorithms perform as well as the best bandit algorithms, *BIC* or not?
Exploration vs. exploitation

Algorithm wants to balance exploration & exploitation, can choose suboptimal arms for the sake of new info

Each agent is myopic: does not care to explore, only exploits

… based on what she knows:

- common prior
- recommendation algorithm
- algorithm’s recommendation (& extra info, if any)

Does not see entire feedback from previous agents
Information asymmetry

- Revealing all info to all agents does not work
  
  Then algorithm can only exploit ⇒ not good.
  
  E.g.: can only pick the “prior best” arm.

- So, algorithm needs to reveal less than it knows.
  
  W.l.o.g., reveal only recommended arm, no extra info

Approach: hide *a little exploration* in *lots of exploitation*.

- Each agent gets “exploitation” with high prob,
  “exploration” with low prob, but does not know which
Related work: multi-armed bandits

- Studied in Econ, OR and CS since 1933
- Most related: IID rewards, with or without a prior
  E.g.: Thompson Sampling, Gittins Index, UCB1 (Auer et al.’02).
- **Best arm prediction**: care about learning rate, not total reward
  E.g.: Even-Dar et al.’02, Goel et al.’09, Bubeck et al.’11.
- Bandits with agents/incentives:
  - **Dynamic pricing** (E.g.: Kleinberg & Leighton’03, Besbes & Zeevi’09)
  - **Ad auctions with unknown CTRs** (E.g.: Babaioff et al.’09,’10,’13)
  - **Dynamic auctions** (E.g.: Athey & Segal’13, Bergemann & Valimaki’10)
Related work: BIC exploration in Econ

- Kremer, Mansour, Perry (2014): same model, two arms. Bayesian-optimal algorithm for deterministic rewards, very suboptimal performance for IID rewards
- Connections to some high-profile work in Economics
  Bayesian Persuasion (Kamenica & Gentzkow: Econometrica’11)
  Strategic Experimentation (Bolton & Harris: Econometrica’99, Keller, Rady & Cripps: Econometrica’05)
Outline

✔ Basic model: BIC bandits
☐ Our results for BIC bandits
☐ Algorithms and key ideas
☐ Beyond BIC bandits
☐ Discussion and open questions
How to measure performance?

For the first $t$ rounds:

- Expected total reward of the algorithm $W(t)$
- Ex-post regret $R_{\text{ex}}(t) = t \cdot (\max \mu_a) - W(t)$
- Bayesian regret $R(t) = \mathbb{E}_{\text{prior}}[R_{\text{ex}}(t)]$

Can **BIC** bandit algorithms attain optimal regret?
Our results: optimal regret

BIC algorithm with optimal ex-post regret for constant #arms:

\[ R_{\text{ex}}(T) = O\left( \min\left( \frac{\log T}{\Delta}, \sqrt{T \log T} \right) \right) + c_P \log T \]

For given \((\mu_1, \ldots, \mu_K)\): \(\Delta\) is the gap between best and 2nd-best arm. Optimal for given \(\Delta\).

optimal regret in the worst case

Depends on prior \(\mathcal{P}\). “Price” for BIC.

Conceptually: exploration schedule is \textit{adaptive} to previous observations

Resolve BIC bandit exploration for constant #arms
Our results: detail-free algorithm

Our algorithm is *detail-free*: requires little info about the prior

- \( N > N_0 \), where \( N_0 \) is a constant that depends on the prior
- \( \hat{\mu} \): approx. min prior mean reward

\[
\mu_{\text{min}} = \min_{\text{arms } i} \mathbb{E}_{\text{prior}}[\mu_i]
\]

Extra perks:

- Algorithm does not need to know \( N_0 \) and \( \mu_{\text{min}} \) exactly
- Agents can have different beliefs, if they believe that:
Our results: black-box reduction

Given arbitrary bandit algorithm \( \mathcal{A} \), produce BIC bandit algorithm \( \mathcal{A}' \) with similar performance:

- Bayesian regret increases only by constant factor \( c_P \) (which depends only on the prior \( \mathcal{P} \)).
- Learning rate decreases by factor \( c_P \):
  
  Suppose \( \mathcal{A} \) outputs a prediction \( \phi_t \) in each round \( t \).
  Then \( \mathcal{A}' \) outputs a prediction \( \phi'_t \) distributed as \( \phi[t/c_P] \).

Modular design: use existing \( \mathcal{A} \), inject BIC

can incorporate auxiliary info (e.g., prior);
exploration preferences (e.g., arms to favor)

predict beyond the best arm
(e.g., worst arm)
Outline

✓ Basic model: BIC bandits
✓ Our results for BIC bandits
☐ Algorithms and key ideas
☐ Beyond BIC bandits
☐ Discussion and open questions
How to sample the other arm?

Hide *exploration* in a large pool of *exploitation*

- Pick “exploit arm” via posterior update
- Recommend arm 2 in round chosen u.a.r.

Enough samples of arm 1 ⇒ arm 2 could be the exploit arm!
Agent with rec=arm 2 for exploration does not know it!
Exploration prob. low enough ⇒ follow recommendation.
Black-box reduction from algorithm \( \mathcal{A} \)

2 arms: \( \mathbb{E}_{\text{prior}}[\mu_1 > \mu_2] \)

Pick “exploit arm”

arm 2

the “exploit arm”

arm 1

Sampling stage: sample each arm

Re-compute “exploit arm”

Call \( \mathcal{A} \) once, report back

The “exploit arm”

Simulation stage

phase

time

repeat

Enough initial samples \( \Rightarrow \) any arm could be the exploit arm!

Agent does not know: exploitation or algorithm \( \mathcal{A} \)?

“Algorithm” prob. low enough \( \Rightarrow \) follow recommendation.

Performance: \( \mathbb{E}_{\text{prior}}[\text{reward}] \) of exploit arm \( \geq \) that of \( \mathcal{A} \)
Black-box reduction from algorithm $\mathcal{A}$

If algorithm $\mathcal{A}$ outputs a prediction $\phi_t$ in each round, the reduction outputs the same prediction in all of next phase. Prediction in round $t$ is distributed as $\phi_{\lfloor t/L \rfloor}$, $L =$ phase length.
Black-box reduction from algorithm $\mathcal{A}$

**Sampling stage:** sample each arm

**Simulation stage**

- **Re-compute “exploit arm”**
- **The “exploit arm”**

2 arms: $\mathbb{E}_{\text{prior}}[\mu_1 > \mu_2]$

Pick “exploit arm”

- **arm 1**
- **the “exploit arm”**

arm 2

Call $\mathcal{A}$ once, report back

**phase**

Re-compute “exploit arm”

Sufficient phase length should not grow over time!

(How low should explore prob. be to convince?)

Analysis of incentives should not depend on algorithm $\mathcal{A}$. 
Sampling stage for many arms

Phase $i = 2, 3, \ldots, m$: sample arm $i$

Re-compute $a^*$ via posterior update

Recommend arm $i$ in rounds chosen u.a.r.

Need to make sure that arm $i$ could be the exploit arm!

sample each arms $j < i$ enough times

Exploration prob. low enough $\Rightarrow$ follow recommendation.
The detail-free algorithm

Detail-free ⇒ cannot use Bayesian update
Ex-post regret ⇒ best posterior arm may not suffice

Define “exploit arm” & “elimination condition” via sample averages.
For BIC, connect sample averages to Bayesian posteriors (tricky!).
Enough initial samples ⇒ “Active arms elimination” is BIC
Assumptions on the prior

• Hopeless for some priors
e.g., if $\mu_1$ and $\mu_1 - \mu_2$ are independent.

• Assumption for two arms: for $k$ large enough,

$$\mathbb{P}(\mathbb{E}[\mu_2 - \mu_1 | k \text{ samples of arm 1}] > 0) > 0.$$  

Arm 2 can become “exploit arm” after enough samples of arm 1.

• Necessary for BIC algorithms (to sample arm 2).
  Sufficient for black-box reduction!

• Similar condition for black-box reduction with $> 2$ arms
Includes: independent priors, bounded rewards, full support on $[L,H]$  

Suffices for the detail-free algorithm
Outline

✓ Basic model: BIC bandits
✓ Our results for BIC bandits
✓ Algorithms and key ideas
☐ Beyond BIC bandits
☐ Discussion and open questions
Black-box reduction with contexts

Our black-box reduction “works” in a very general setting.

For each round $t$, algorithm observes context $x_t$, then:

- recommends an arm, and (possibly) makes a prediction
- agent chooses an arm, reports her reward & extra feedback

Distribution of reward & feedback depend on arm & context

- allows (limited) agent heterogeneity
- incorporates three major lines of work on bandits: with contexts, with extra feedback, and with predictions

- e.g., customer profile @Amazon
- e.g., detailed restaurant reviews
Setup & result

Contextual Bayesian regret

\[ R_\Pi(t) = \mathbb{E}_{\text{prior}}[W(t; \pi^*) - W(t; \mathcal{A})] \]

Bayesian incentive-compatibility (BIC):

\[ \mathbb{E}_{\text{prior}}[\mu_{x,a} - \mu_{x,b} \mid x_t = x, \text{rec}_t = a] \geq 0 \]

\forall \text{time } t, \text{context } x, \text{arms } a, b

Policy \( \pi: \{\text{contexts}\} \rightarrow \{\text{arms}\} \)

Fixed set of policies \( \Pi \)

\( \pi^* : \text{best policy in } \Pi \)

Arms \( a \), contexts \( x \).

Expected reward \( \mu_{x,a} \in [0,1] \).

Reduction: bandit algorithm \( \mathcal{A} \xrightarrow{} \text{BIC bandit algorithm } \mathcal{A}' \)

with similar Bayesian regret & prediction quality

Unlike algorithms, our reduction does not depend on:

policy set \( \Pi \), what is extra feedback, or what is predicted
Algorithm

- **Defn**: *arm-rank* of arm $a$ given context $x$ is $i$ iff arm $a$ is $i$-th best given $x$, according to the prior.
- **Key idea**: recommend *arm-ranks instead of arms*.
- Maintain a dataset $\mathcal{D}$ of rank-samples: $(x, i, \text{reward}, \text{feedback})$.

Exploit arm $a_x^*$: best posterior arm for context $x$ given $\mathcal{D}$

**Sampling stage**: sample each arm-rank

**Simulation stage**:
- update $\mathcal{D}$
- Call $\mathcal{A}$
- The “exploit arm” $a_x^*$

**Time**
- repeat
BIC bandit games

In each round, a fresh batch of agents plays a game (possibly noisy payoffs, same game in every round)

- algorithm recommends an action to each agent
  E.g., driving directions on Waze
- … chooses a distribution over action profiles
- solution concept: *Bayesian correlated equilibrium (BCE)*

Which action profiles are “explorable” by a BIC algorithm?

How to explore all of them?

What is the best *BCE* achievable with all explorable info?

How to converge on this *BCE*?
Outline

✓ Basic model: BIC bandits
✓ Our results for BIC bandits
✓ Algorithms and key ideas
✓ Beyond BIC bandits

☐ Discussion and open questions
Connection to medical trials

- Basic design: new drug vs. placebo (blind, randomized)
- “advanced” designs studied & used (adaptive, >2 arms, contexts)
- Medical trials is one of original motivations for bandits

- Patients’ incentives: why participate? why take less known drug?
  Major obstacle, esp. for wide-spread diseases & cheap drugs.
- Medical trial as a BIC recommendation algorithm
  - OK not to give the patients any data from the trial itself
  - Extension to contexts and extra feedback very appropriate!

How to really convince the patients / model their incentives?
Connection to Systems

- System with many settings/parameters (hidden or exposed) your laptop, smartphone, or Facebook feed
- Optimal settings unclear => need for exploration
  - often: settings are hidden, exploration done covertly
- Alternative: expose the settings, let users decide
  - \textit{explore via incentive-compatible recommendations}
    (e.g., the defaults that users can override)
- The version without incentives is understood in theory, but (sort of) open in practice, need to really solve \textit{that} first.
Auxiliary signals

For each agent, algorithm recommends an arm & sends aux. signal

- If algorithm can control whether to send the aux. signal
  - *not sending* is w.l.o.g. if the prior is fully observed & used
  - aux. signal may help for detail-free algorithms
  - cleaner without aux. signals (and we don’t use them)

- If algorithm *is required* to send some aux. signals
  - complicated – e.g., revealing full stats does not work!
  - may help to reveal more info than required
  - what *must* and *can* be revealed may depend on application
(More) open questions

Optimal dependence on the prior?
Better dependence on \#actions?
Action spaces with known structure?
Use exploration that happens anyway?

Fully detail-free algorithms?
Elicit some info from agents?
(ensure they do not lie)

BIC bandit game with succinct game representation:
better regret, running time?

ML