Bayesian Exploration with Heterogeneous Agents

Anonymous Author(s)

ABSTRACT

It is common in recommendation systems that users both consume and produce information as they make strategic choices under uncertainty. While a social planner would balance “exploration" and “exploitation" using a multi-armed bandit algorithm, usersâĂŹ incentives may tilt this balance in favor of exploitation. We consider Bayesian Exploration: a simple model in which the recommendation system (the “principal") controls the information flow to the users (the “agents") and strives to incentivize exploration via information asymmetry. A single round of this model is a version of a well-known “Bayesian Persuasion game" from [19]. We allow heterogeneous users, relaxing a major assumption from prior work that users have the same preferences from one time step to another. The goal is now to learn the best *personalized* recommendations. One particular challenge is that it may be impossible to incentivize some of the user types to take some of the actions, no matter what the principal does or how much time she has. We consider several versions of the model, depending on whether and when the user types are reported to the principal, and design a near-optimal “recommendation policy" for each version. We also investigate how the model choice and the diversity of user types impact the set of actions that can possibly be “explored” by each type.

ACM Reference Format:
Anonymous Author(s). 2018. Bayesian Exploration with Heterogeneous Agents. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

[jm: Reviewer 4: the authors interchange between l and ℓ. Please commit to one of the two and use this consistently.] Recommendation systems are ubiquitous in online markets. Well-known examples include recommendations for movies (*e.g.,* Netflix), products (*e.g.,* Amazon), and restaurants (*e.g.,* Yelp). High-quality recommendations are a crucial part of the value proposition of these businesses. A typical recommendation system encourages its users to submit evaluations and/or reviews of their experiences, and aggregates this feedback in order to provide better recommendations to future users. Thus, each user plays a dual rule: she consumes information from the previous users (indirectly, via recommendations), and produces new information (*e.g.,* a review) that benefits future users. This dual role creates a tension between exploration, exploitation, and users' incentives.

A social planner – a hypothetical entity that controls users for the sake of common good – would balance “exploration" vs. “exploitation", *i.e.,* trying out insufficiently known alternatives for the sake of acquiring new information vs. making myopic decisions based on this information. Designing algorithms to trade off these two objectives is a well-researched subject in machine learning and operations research. However, user incentives with respect to exploration are misaligned with those of society. To the extent that a given user decides to “explore", she experiences all the downside of this decision (a potentially suboptimal experience), whereas the upside (improved recommendations) is spread over many users in the future. Absent an adequate mechanism to compensate for the risks of exploration, self-interested users have incentives to exploit. This may result in outcomes that are very suboptimal. First, it may take much longer to arrive at good recommendations. Second, the recommendations may consistently suffer from selection bias in the data: *e.g.,* ratings of a particular movie may mostly come from people who like this type or genre. Third, in some natural but idealized models (*e.g.,* [23, 24]), there are simple examples when optimal recommendations are never found because the corresponding actions are never taken.

Thus, we have a problem of *incentivizing exploration.* Relying on monetary incentives or the natural human propensity for exploration can be financially and technologically infeasible, and/or introduce selection bias. A recent line of work, started by [23], instead strives to incentivize exploration by taking advantage of the inherent *information asymmetry* – the fact that the recommendation system has more information than a typical user – by restricting the information revealed to the agents. These papers posit a simple model, termed *Bayesian Exploration* in [25]. The recommendation system is a “principal" that interacts with a stream of self-interested “agents" arriving one by one. Each agent needs to make a decision: take an action from a given set of alternatives. The principal issues a recommendation, and observes the outcome, but cannot direct the agent to take a particular action. The problem is to design a “recommendation policy" for the principal that learns over time to make good recommendations *and* ensures that the agents are incentivized to follow this recommendation. A single round of this model is a version of a well-known "Bayesian Persuasion game" [19].

Our scope. We depart from prior work on Bayesian Exploration in that we allow the agents to have different preferences from one another. The preferences of an agent are encapsulated in her *type*, *e.g.,* vegan vs meat-lover. When an agent takes a particular action, the outcome depends on the action itself (*e.g.,* the selection of restaurant), the "state" of the world (*e.g.,* the qualities of the restaurants), and the type of the agent. The state is persistent (does not change over time), but initially not known; a Bayesian prior on the state is common knowledge. In each round, the agent type is drawn independently from a fixed and known distribution. The principal strives to learn the best possible recommendation for each agent type.

We consider three models, depending on whether and when the agent type is revealed to the principal: the type is revealed immediately after the agent arrives (*public types*), the type is revealed only after the principal issues a recommendation (*reported types*),[1] and the type is never revealed (*private types*). We design a near-optimal recommendation policy for each modeling choice. The "benchmark" that our policies compete with is the "best-in-hindsight" policy: a recommendation policy whose Bayesian-expected reward is optimal for a particular problem instance. [jm: Reviewer 2: what is a particular problem instance?]

Explorability and public types. A distinctive feature of Bayesian Exploration is that it may be impossible to incentivize some agent types to take some actions, no matter what the principal does or how much time she has. For a more precise terminology, a given type-action pair is *explorable* if this agent type takes this action under some recommendation policy in some round with positive probability. This action is also called *explorable* for this type. Thus: some type-action pairs might not be explorable. Moreover, one may need to explore to find out which pairs are explorable.

Recommendation policies cannot do better than the "best explorable action" for a particular agent type: an explorable action with a largest reward in the realized state. Our recommendation policy for public types matches this benchmark in the long run, i.e., after a "constant" number of rounds (which depends on the problem instance but not the time horizon). While it is easy to prove that such a policy exists, the challenge is to provide it as an explicit procedure.

Our policy focuses on exploring all explorable type-action pairs. Exploration needs to proceed gradually, whereby exploring one action may enable the policy to explore another. In fact, exploring some action for one type may enable the policy to explore some action for another type. Accordingly, our policy proceeds in phases: in each phase, we explore all actions for each type that can be explored "immediately", with information available in the beginning of the phase.

An important building block is the analysis of the single-round game. We use information theory to characterize how much state-relevant information the principal has. In particular, we prove a version of *information-monotonicity*: essentially, the set of all explorable type-action pairs can only increase if the principal has more information.

[1]Reported types may arise if the principal asks agents to report the type after the recommendation is issued, *e.g.,* in a survey. While the agents are allowed to misreport their respective types, they have no incentives to do that.

Beyond public types. For private or reported types, recommending one particular action to the current agent is not very meaningful because the agents' type is not yet known to the principal. Instead, one can recommend a *menu*: a mapping from agent types to actions. Analogous to the case of public types, we focus on explorable menus and gradually explore all such menus, eventually matching the Bayesian-expected reward of the best explorable menu. One difficulty is that exploring a given menu does not immediately reveal the reward of a particular type-action pair (because multiple types could map to the same action). Consequently, even keeping track of what the policy knows is now non-trivial. The analysis of the single-round game becomes more involved, as one needs to argue about "approximate information-monotonicity". To handle these issues, our recommendation policy satisfies only a relaxed version of incentive-compatibility.

Our policy for reported types does much better: we show that in the long run it matches our benchmark for public types. This may seem counterintuitive because "reported types" are completely useless to the principal in the single-round game (whereas public types are very useful). Essentially, we reduce the problem to the public types case, at the cost of a much longer exploration.

Comparative statics for explorability. We study how the set of all explorable type-action pairs (*explorable set*) is affected by the model choice and the diversity of types. First, we fix an arbitrary problem instance, and we find that the explorable set stays the same if we transition from public types to reported types [jm: Reviewer 2: not a valid sentence], and can only decrease if we transition from reported types to private types. We provide a concrete example when the latter transition makes a huge difference. Second, let us vary the distribution $\mathcal{D}$ of agent types. For public types (and therefore also for reported types), we find that the explorable set is determined by the support set of $\mathcal{D}$. Further, if we increase the support set, then the explorable set can only increase. In other words, *diversity of agent types helps exploration*. We provide a concrete example when the explorable set increases very substantially even if the support set increases by a single type. However, for private types the picture is quite different: we provide an example when *diversity hurts*, in the same sense as above.

Related work. The problem of Bayesian Exploration was introduced in [23]. The special case of homogenous agents has been largely resolved, in terms of the optimal policy for two actions [23], explorability [25], and regret minimization for stochastic utilities [24]. [24] also consider an extension to public types, under a very strong assumption geared to ensure that all type-action pairs are explorable. [25] allows several agents to arrive in each round and play a game. Agents can have different types, but the tuple of types stays the same from one round to another (and is known to the principal). [5] enrich the model to allow agents to observe recommendations of their "friends" in a known social network.

Several other papers study related, but technically different models. [8] consider a basic setting of Bayesian Exploration, but with time-discounted utilities. [14] allow monetary incentives. [11] posit a continuous information flow and a continuum of agents. [22] propose a mechanism to coordinate costly exploration decisions in social learning. [27] consider a "full-revelation" recommendation system, and show that (under some substantial assumptions) agent heterogeneity leads to exploration. Scenarios with long-lived, exploring agents and no principal to coordinate them have been studied in [9, 20] under the name *strategic experimentation*.

Exploration-exploitation tradeoff received much attention over the past decades, usually under the rubric of "multi-armed bandits"; see [10, 16] for background. Absent incentives, Bayesian Exploration with public types is a well-studied problem of "contextual bandits" (with deterministic rewards and a Bayesian prior). A single round of Bayesian Exploration is a version of the Bayesian Persuasion game [19], where the signal observed by the principal is distinct from the state. Exploration-exploitation problems with incentives issues arise in several other scenarios: dynamic pricing [4, 7, 21], dynamic auctions [1, 6, 18], pay-per-click ad auctions [2, 3, 13], and human computation [15, 17, 28].

2 MODEL AND PRELIMINARIES

[jm: Reviewer 2: The model description is not clear. In particular, it did not describe what the principal knows about the states, agents' reward functions etc., and what the agents know. Note that it is crucial to be clear about who knows what since the information asymmetry is essential to the model. Is a new state drawn at every round t or is the state of nature the same across all rounds? I think for the model to make sense, it should be the latter case. Then, the questions are: (1) how does the principal's belief about the state of nature evolve? (2) in the benchmark OPT_{pub}s rightly above Section 3, why you need to take expectation over w? Why not just the particular w realized at the beginning of the game?

Then at each single round, what do principal and agents know about S?]

Bayesian exploration is a game between a principal and T agents who arrive one by one, with agent t arriving in round t. Each agent t has a type $\theta_t \in \Theta$, drawn independently from a fixed distribution, and an action space $\mathcal{A}$. There is uncertainty, captured by a latent "state of nature" $\omega \in \Omega$, henceforth simply the *state*, drawn from a Bayesian prior at the beginning of time and fixed across rounds. The reward $r_t = u(\theta_t, a_t, \omega)$ is determined by the type θ_t, the action a_t, and the state ω, for some fixed, deterministic function $u : \Theta \times \mathcal{A} \times \Omega \to [0, 1]$. We assume that the sets $\mathcal{A}$, Θ and Ω are finite. We use ω_0 as the random variable for the state, and write $\Pr[\omega]$ for $\Pr[\omega_0 = \omega]$. Similarly, we write $\Pr[\theta]$ for $\Pr[\theta_t = \theta]$. **NSI: do we assume r is bounded or anything like that?** [jm: Reward is defined as the utility which is defined to be in [0, 1].] An *instance* of the Bayesian exploration game consists of T, the type distribution, the action space, the prior over the state, and the reward function.

A solution to the Bayesian exploration game is a randomized online algorithm π termed "recommendation policy" which, at each round t, maps the instance, as defined above, and current history, defined below, to a message m_t which, in general, is an arbitrary bit string of length polynomial in the size of the instance. The *history* includes the rewards and agent type information obtained in the past rounds and current round up until the recommendation is issued.**NSI: is the following footnote required?**[2] We consider three model variants, depending on whether and when the principal learns the agent's type: the type is revealed immediately after the agent arrives (*public types*), the type is revealed only after the principal issues a recommendation (*reported types*), the type is not revealed (*private types*). Hence the history at round t is $\{(r_1, \theta_1), \ldots, (r_{t-1}, \theta_{t-1}), r_t\}$ for public types, $\{(r_1, \theta_1), \ldots, (r_{t-1}, \theta_{t-1})\}$ for reported types, and $\{r_1, \ldots, r_{t-1}\}$ for private types.

The goal of each agent t is to choose an action a_t so as to maximize her Bayesian-expected reward given the instance, her type θ_t, and the policy π. The goal of the principal is to choose a policy π that maximizes (Bayesian-expected) total reward, *i.e.,* $\mathbb{E}[\sum_{t=1}^{T} r_t]$, where the expectation is over the randomness in the instance and the policy.[3]

Bayesian-incentive compatibility. For public types, we assume that the message in each round is an action, *i.e.,* $m_t \in \mathcal{A}$, and the recommendation policy π is *Bayesian incentive-compatible* (*BIC*). This is without loss of generality, by a suitable version of Myerson's "revelation principle".

The BIC condition is stated as follows. Let $\mathcal{E}_{t-1}$ be the event that the agents have followed principal's recommendations up to (but not including) round t. For any type θ and round t, let $\pi^t(\theta)$ be the action recommended by π in round t. The recommendation policy is BIC if for each round t, type θ, and and any two actions a, a' such that $\Pr[\pi^t(\theta) = a | \mathcal{E}_{t-1}] > 0$ it holds that

$$\mathbb{E}\left[\, u(\theta, a, \omega) - u(\theta, a', \omega) \mid \pi^t(\theta) = a, \mathcal{E}_{t-1} \,\right] \geq 0. \tag{1}$$

(The expectation is over the realized state ω and the randomness in the policy, thus capturing the uncertainty in the agent's information.)

For reported types and private types, we restrict each message to be a *menu*, i.e., a mapping from types to actions, and assume that the recommendation policy satisfies a similar but technically different BIC condition. Again, this is without loss of generality by revelation principle. To state the BIC condition, let π^t be the menu recommended in round t. The recommendation policy π is BIC if for each round t, type θ, and any two menus m, m' such that $\Pr[\pi^t = m | \mathcal{E}_{t-1}] > 0$, we have

$$\mathbb{E}\left[\, u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega) \mid \pi^t = m, \mathcal{E}_{t-1} \,\right] \geq 0. \tag{2}$$

(Again, the expectation is over the realized state ω and the randomness in the policy.) **Explorability and benchmarks.** For public types, a type-action pair $(\theta, a) \in \Theta \times \mathcal{A}$ is called *eventually-explorable* in state ω if there is some BIC recommendation policy that, for T large enough, eventually recommends this action to this agent type with positive probability. Then action a is called *eventually-explorable* for type θ and state ω. The set of all such actions is denoted $\mathcal{A}_{\omega,\theta}$. Likewise, for private types, a menu is called *eventually-explorable* in state ω if there is some BIC recommendation policy that eventually recommends this menu with positive probability. The set of all such menus is denoted $\mathcal{M}_\omega$.

[2]Formally, for randomized algorithms, we also assume the history contains the random seed and so can simulate the algorithm on prior rounds given the history.

[3]Note, the principal must commit to the policy, given only the instance. The policy, however, includes the history and thus can adapt recommendations to inferences about the state based on the history. See Example 3.2.

Our benchmark is the best eventually-explorable recommendation:

$$\mathtt{OPT}_{\mathtt{pub}} = \sum_{\theta\in\Theta,\omega\in\Omega} \Pr[\omega]\cdot\Pr[\theta]\cdot \max_{a\in\mathcal{A}_{\omega,\theta}} u(\theta,a,\omega).$$

(for public types: actions)

$$\mathtt{OPT}_{\mathtt{pri}} = \sum_{\omega\in\Omega} \Pr[\omega]\cdot \max_{m\in\mathcal{M}_\omega} \sum_{\theta\in\Theta} \Pr[\theta]\cdot u(\theta,m(\theta),\omega)$$

(for private types: menus).

We have $\mathtt{OPT}_{\mathtt{pub}} \geq \mathtt{OPT}_{\mathtt{pri}}$, essentially because any BIC policy for private types can be simulated as a BIC policy for public types. We provide an example (Example 3.2) when $\mathtt{OPT}_{\mathtt{pub}} > \mathtt{OPT}_{\mathtt{pri}}$.

3 COMPARATIVE STATICS FOR EXPLORABILITY

The eventually-explorable type-action pairs are affected by the model choice and the diversity of types. All else equal, settings with greater potential exploration have greater total expected reward in both the benchmark and approximation guarantee. Our first result shows that models with public or reported types can explore (strictly) more actions for each type than models with private types. Thus more type information (strictly) improves outcomes. Our second result shows that **NSI: add something here.**

Explorability and the model choice. Fix an instance of Bayesian exploration. We will show in Section 4.3 that the eventually-explorable set of type-action pairs in a given state of the world is the same for public and reported types. Let $\mathcal{A}^{\mathtt{pub}}_\omega$ be the set of all type-action pairs $\{(\theta,a)|a\in\mathcal{A}_{\omega,\theta}\}$ be the eventually-explorable type-action pairs in state ω with public (equivalently, reported) types. Similarly, let $\mathcal{A}^{\mathtt{pri}}_\omega$ be the set of all type-action pairs $(\theta,m(\theta))$ that appear in some eventually-explorable menu $m\in\mathcal{M}_\omega$ in state ω with private types.

It is fairly easy to argue that $\mathcal{A}^{\mathtt{pri}}_\omega \subseteq \mathcal{A}^{\mathtt{pub}}_\omega$. The idea in the proof is that one can simulate any BIC recommendation policy for private types with a BIC recommendation policy for public types; we omit the details. **NSI: might want to include it if there's time.**

Claim 3.1. *Any type-action pair eventually-explorable with private types is also eventually-explorable with public types:* $\mathcal{A}^{\mathtt{pri}}_\omega \subseteq \mathcal{A}^{\mathtt{pub}}_\omega$.

Interestingly, as the following example shows, $\mathcal{A}^{\mathtt{pri}}_\omega$ can in fact be a *strict* subset of $\mathcal{A}^{\mathtt{pub}}_\omega$.

Example 3.2. There are two states, two types and two actions: $\Omega=\Theta=\mathcal{A}=\{0,1\}$. States and types are drawn uniformly at random: $\Pr[\omega=0]=\Pr[\theta=0]=\frac{1}{2}$. Rewards are defined in the following table:

	$a=0$	$a=1$
$\theta=0$	$u=3$	$u=4$
$\theta=1$	$u=2$	$u=0$

	$a=0$	$a=1$
$\theta=0$	$u=2$	$u=0$
$\theta=1$	$u=3$	$u=4$

Table 1: Rewards $u(\theta,a,\omega)$ when $\omega=0$ and $\omega=1$.

Action 0 is preferred by both types when there is no information beyond the prior about the state. Thus in the first round, the principal must recommend action 0 in order for the policy to be BIC. Hence type-action pairs $\{(0,0),(1,0)\}$ are eventually-explorable in all models.

In the second round, the principal knows the reward of the first-round agent. When types are public or reported, the reward together with the type is sufficient information for the principal to learn the state. Moving forward, the principal can now recommend the higher-reward action for each type (either directly or, in the case of reported types, through a menu). Thus, type-action pair $(0,1)$ is eventually-explorable when $\omega=0$ and, similarly, type-action pair $(1,1)$ is eventually-explorable when $\omega=1$.

For private types, samples from the first-round menu (which, as argued above, must recommend action 0 for both types) do not convey any information about the state, as they have the same distribution in both states. Therefore, action 1 is not eventually-explorable, for either type and either state. Thus:

Claim 3.3. *In Example 3.2,* $\mathcal{A}^{\mathtt{pri}}_\omega$ *is a strict subset of* $\mathcal{A}^{\mathtt{pub}}_\omega$.

Explorability and diversity of agent types. NSI: this section needs editing. Let us discuss whether and how the type distribution affects the explorable set. Fix an instance of Bayesian Exploration with type distribution $\mathcal{D}$. Let us see how the explorable set changes if we change $\mathcal{D}$ to some other distribution $\mathcal{D}'$.

Let $\mathcal{A}_\omega$ and $\mathcal{A}'_\omega$ be the corresponding explorable sets, for each state ω. Let $\mathtt{support}(\mathcal{D})$ be the support set of distribution $\mathcal{D}$, *i.e.,* the set of all feasible agent types according to $\mathcal{D}$.

For public types, we show that the explorable set is determined by the support set of $\mathcal{D}$, and can only increase if the support set increases:

Claim 3.4. *Consider Bayesian Exploration with public types. Then:*
(a) if $\mathtt{support}(\mathcal{D})=\mathtt{support}(\mathcal{D}')$ *then* $\mathcal{A}_\omega=\mathcal{A}'_\omega$.
(b) if $\mathtt{support}(\mathcal{D})\subset\mathtt{support}(\mathcal{D}')$ *then* $\mathcal{A}_\omega\subseteq\mathcal{A}'_\omega$.

Proof Sketch. We follow the steps of the proof of Lemma 4.7. The idea is that the proof carries through even if π is a BIC recommendation policy for the less diverse problem instance, *i.e.,* an instance with type distribution $\mathcal{D}'$ such that $\mathtt{support}(\mathcal{D}')\subset\mathtt{support}(\mathcal{D})$. Then for a given state ω, Algorithm 2 for the original distribution $\mathcal{D}$ instance explores all type-action pairs in $\mathcal{A}'_\omega$. □

The conclusions in Claim 3.4 apply to reported types, too. This is because explorable sets are the same for public and reported types.

For private types, the situation is more complicated. More types can help for some problem instances. For example, if different types have disjoint sets of available actions (more formally: say, disjoint sets of actions with positive rewards) then we are essentially back to the case of reported types, and the conclusions in Claim 3.4 apply. On the other hand, we can use Example 3.2 to show that more types can hurt explorability when types are private. Recall that in this example, for private types only action 0 can be recommended. Now consider a less diverse instance in which only type 0 appears. After one agent in that type chooses action 0, the state is revealed to the principal. For example, when the state $\omega=0$, action 1 can be recommended to future agents. This shows that, in this example, explorable set increases when we have fewer types.

4 BAYESIAN EXPLORATION WITH PUBLIC TYPES

In this section, we develop our recommendation policy for public types. Throughout, $\texttt{OPT} = \texttt{OPT}_{\text{pub}}$.

Theorem 4.1. *Consider an arbitrary instance of Bayesian Exploration with public types. There exists a BIC recommendation policy with expected total reward at least $(T - C) \cdot \texttt{OPT}$, for some constant C that depends on the problem instance but not on T. This policy explores all type-action pairs that are eventually-explorable for a given state.*

4.1 A single round of Bayesian Exploration

Let us analyze a single round of the Bayesian exploration. Throughout, let θ be the agent's type.

Signal and explorability. We posit that the principal receives a signal S before the round starts, where S is a random variable. The *signal structure* of S, denoted as $\mathcal{S}$, consists of the support set $\mathcal{X}$ and a joint distribution of (S, ω_0). Here S represents the history of the previous rounds and the internal random seed, and the signal structure represents information about S available to the current agent. However, for this subsection it is more lucid to consider an arbitrary signal structure.

A single-round recommendation policy π is simply a randomized mapping from signal realizations to actions. The policy is called *Bayesian-incentive compatible* (*BIC*) for signal S if for each type θ and any two actions a, a' such that $\Pr[\pi(S) = a] > 0$ it holds that

$$\mathbb{E}\left[\; u(\theta, a, \omega) - u(\theta, a', \omega) \mid \pi(S) = a \;\right] \geq 0. \tag{3}$$

[jm: Reviewer 2 asked what this expectation is over.]

Definition 4.2. Consider a single-round of Bayesian exploration when the principal receives signal S with signal structure $\mathcal{S}$. An action $a \in \mathcal{A}$ is called signal-explorable, for a given type θ and realized signal s, if there exists a single-round BIC recommendation policy π such that $\Pr[\pi(s) = a] > 0$. The set of all such actions is denoted as $\texttt{EX}_s[\mathcal{S}]$. The signal-explorable set is defined as $\texttt{EX}[\mathcal{S}] = \texttt{EX}_S[\mathcal{S}]$.

[jm: Reviewer 2: The notation about $s, S, \mathcal{S}$ are confusing. Some places say "realized state s" (e.g., in definition 4.2 and several other places) and some places say "realized state S" (e.g., in Algorithm 1, Claim 4.6 and several other places).]

Note that $\texttt{EX}[\mathcal{S}]$ is a random subset of actions whose realization is determined by s.

Information-monotonicity. We compare the information content of two signals using the notion of conditional mutual information (see Appendix A for definition and background). Essentially, we show that a more informative signal leads to the same or larger explorable set.

Definition 4.3. We say that signal S *is at least as informative* as signal S' if $I(S'; \omega_0 \mid S) = 0$.

Intuitively, the condition $I(S'; \omega_0 | S) = 0$ means if one is given random variable S, one can learn no further information from S' about ω_0. Note that this condition depends not only on the signal structures of the two signals, but also on their joint distribution.

Lemma 4.4. *Let S, S' be two signals with signal structures $\mathcal{S}, \mathcal{S}'$. If S is at least as informative as S', then $\texttt{EX}_{s'}[\mathcal{S}'] \subseteq \texttt{EX}_s[\mathcal{S}]$ for all s', s such that $\Pr[S = s, S' = s'] > 0$.*

Proof. Consider any BIC recommendation policy π' for signal structure $\mathcal{S}'$. We construct π for signal structure $\mathcal{S}$ by setting $\Pr[\pi(s) = a] = \sum_{s'} \Pr[\pi'(s') = a] \cdot Pr[S' = s' \mid S = s]$. Notice that $I(S'; \omega_0 \mid S) = 0$ implies S' and ω_0 are independent given S, i.e $\Pr[S' = s' \mid S = s] {\cdot} \Pr[\omega_0 = \omega \mid S = s] = \Pr[S' = s', \omega_0 = \omega \mid S = s]$ for all s, s', ω. Therefore, for all s' and ω,

$$\begin{aligned}
&\sum_s \Pr[S' = s' \mid S = s] \cdot \Pr[\omega_0 = \omega, S = s] \\
&= \textstyle\sum_s \Pr[S' = s' \mid S = s] \cdot \Pr[\omega_0 = \omega \mid S = s] \cdot \Pr[S = s] \\
&= \textstyle\sum_s \Pr[S' = s', \omega_0 = \omega \mid S = s] \cdot \Pr[S = s] \\
&= \textstyle\sum_s \Pr[S = s, S' = s', \omega_0 = \omega] \\
&= \Pr[\omega_0 = \omega, S' = s'].
\end{aligned}$$

Therefore π' being BIC implies that π is also BIC. Indeed, for any $a, a' \in \mathcal{A}$ and $\theta \in \Theta$, by plugging in the definition of π,

$$\begin{aligned}
&\textstyle\sum_{\omega,s}\ \Pr[\omega_0 = \omega, S = s] \cdot (u(\theta, a', \omega) - u(\theta, a, \omega)) \cdot \Pr[\pi(s) = a] \\
&= \textstyle\sum_{\omega,s'}\ \Pr[\omega_0 = \omega, S' = s'] \cdot (u(\theta, a', \omega) - u(\theta, a, \omega)) \cdot \Pr[\pi'(s') = a] \\
&\geq 0.
\end{aligned}$$

Finally, for any s', s, a such that $Pr[S' = s', S = s] > 0$ and $\Pr[\pi'(s') = a] > 0$, we have $\Pr[\pi(s) = a] > 0$. This implies $\texttt{EX}_{s'}[\mathcal{S}'] \subseteq \texttt{EX}_s[\mathcal{S}]$. □

Max-Support Policy. We can solve the following LP to check whether a particular action $a_0 \in \mathcal{A}$ is signal-explorable given a particular realized signal $s_0 \in \mathcal{X}$. In this LP, we represent a policy π as a set of numbers $x_{a,s} = \Pr[\pi(s) = a]$, for each action $a \in \mathcal{A}$ and each feasible signal $s \in \mathcal{X}$.

maximize x_{a_0,s_0}
subject to:

$$\begin{aligned}
&\textstyle\sum_{\omega \in \Omega, s \in \mathcal{X}}\ \Pr[\omega] \cdot \Pr[s \mid \omega] \cdot \\
&\big(u(\theta, a, \omega) - u(\theta, a', \omega)\big) \cdot x_{a,s} \geq 0 \ \forall a, a' \in \mathcal{A} \\
&\textstyle\sum_{a \in \mathcal{A}}\ x_{a,s} = 1, && \forall s \in \mathcal{X} \\
&x_{a,s} \geq 0, && \forall s \in \mathcal{X}, a \in \mathcal{A}
\end{aligned}$$

Since the constraints in this LP characterize any BIC recommendation policy, it follows that action a_0 is signal-explorable given realized signal s_0 if and only if the LP has a positive solution. If such solution exists, define recommendation policy $\pi = \pi^{a_0, s_0}$ by setting $\Pr[\pi(s) = a] = x_{a,s}$ for all $a \in \mathcal{A}, s \in \mathcal{X}$. Then this is a BIC recommendation policy such that $\Pr[\pi(s_0) = a_0] > 0$.

Definition 4.5. Given a signal structure $\mathcal{S}$, a BIC recommendation policy π is called *max-support* if $\forall s \in \mathcal{X}$ and signal-explorable action $a \in \mathcal{A}$ given s, $\Pr[\pi(s) = a] > 0$.

It is easy to see that we obtain max-support recommendation policy by averaging the $\pi^{a,s}$ policies defined above. Specifically, the following policy is BIC and max-support:

$$\pi^{\max} = \frac{1}{|\mathcal{X}|}\sum_{s\in\mathcal{X}} \frac{1}{|\mathsf{EX}_s[\mathcal{S}]|}\sum_{a\in\mathsf{EX}_s[\mathcal{S}]} \pi^{a,s}. \tag{4}$$

Maximal Exploration. We design a subroutine `MaxExplore` which outputs a sequence of actions with two properties: it includes every signal-explorable action at least once, and the marginal distribution at each location is $\pi^{\max}$. The length of this sequence, denoted L_θ, should satisfy

$$L_\theta \geq \max_{(a,s)\in\mathcal{A}\times\mathcal{X} \text{ with } \Pr[\pi^{\max}(s)=a]\neq 0} \frac{1}{\Pr[\pi^{\max}(s)=a]}. \tag{5}$$

This step is essentially from [26]; [jm: Reviewer 2 mentioned that we cite your EC paper twice. I know Alex wants to show that this is from your working paper which is not the version for EC. But it might confuse the reviewer. Not sure if we should do this.] we provide the details below for the sake of completeness. The idea is to put $C_a = L_\theta \cdot \Pr[\pi^{\max}(S) = a]$ copies of each action a into a sequence of length L_θ and randomly permute the sequence. [jm: Reviewer 2: Equation (5) used $\pi^{max}(s)$, but rightly after $\pi^{max}(S)$ is used. JM: reviewer's confusion about realized vs random values continue.] However, C_a might not be an integer, and in particular may be smaller than 1. The latter issue is resolved by making L_θ sufficiently large. For the former issue, we first put $\lfloor C_a \rfloor$ copies of each action a into the sequence, and then sample the remaining $L_\theta - \sum_a \lfloor C_a \rfloor$ actions according to distribution $p^{\mathrm{res}}(a) = \frac{C_a - \lfloor C_a \rfloor}{L_\theta - \sum_a \lfloor C_a \rfloor}$. For details, see Algorithm 1.

Algorithm 1 Subroutine MaxExplore

```
 1: Input: type $\theta$, realized signal $S$ and signal structure $\mathcal{S}$.
 2: Output: a list of actions $\alpha$
 3: Compute $\pi^{\max}$ as per (4)
 4: Initialize $Res = L_\theta$.
 5: for each action $a \in \mathcal{A}$ do
 6:     $C_a \leftarrow L_\theta \cdot \Pr[\pi^{\max}(S) = a]$
 7:     Add $\lfloor C_a \rfloor$ copies of action $a$ into list $\alpha$.
 8:     $Res \leftarrow Res - \lfloor C_a \rfloor$.
 9:     $p^{\mathrm{res}}(a) \leftarrow C_a - \lfloor C_a \rfloor$
10: $p^{\mathrm{res}}(a) \leftarrow p^{\mathrm{res}}(a)/Res$, $\forall a \in \mathcal{A}$.
11: Sample $Res$ many actions from distribution according to $p^{\mathrm{res}}$
    independently and add these actions into $\alpha$.
12: Randomly permute the actions in $\alpha$.
13: return $\alpha$.
```

Claim 4.6. *Given type θ and realized signal S, MaxExplore outputs a sequence of L_θ actions. Each action in the sequence marginally distributed as $\pi^{\max}$. For any action a such that $\Pr[\pi^{\max} = a] > 0$, a shows up in the sequence at least once with probability exactly 1. MaxExplore runs in time polynomial in L_θ, $|\mathcal{A}|$, $|\Omega|$ and $|\mathcal{X}|$ (size of the support of the signal).*

[jm: Reviewer 2: In the definition of Maximal Exploration, you mentioned"the marginal distribution at each location is π^max". Whose marginal distribution do you mean? ActionsâĂŹ? But π^{max} is a randomized map from S to actions, how could it be a marginal distribution of actions? Do you mean $\pi^{max}(S)$?

Algorithm 2 Main procedure for public types

```
 1: Initialization: signal $S_1 = \mathcal{S}_1 = \perp$, phase count $\ell = 1$, index
    $i_\theta = 0$ for each type $\theta \in \mathbf{\Theta}$.
 2: for rounds $t = 1$ to $T$ do
 3:     if $\ell \leq |\mathcal{A}| \cdot |\mathbf{\Theta}|$ then
 4:         {Exploration} Call thread thread($\theta_t$).
 5:         if every type $\theta$ has finished $L_\theta$ rounds in the current phase
            ($i_\theta \geq L_\theta$) then
 6:             Start a new phase: $\ell \leftarrow \ell + 1$.
 7:             Let $S_\ell$ be the signal for phase $\ell$: the set of all observed
                type-action-reward triples.
 8:             Let $\mathcal{S}_\ell$ be the signal structure for $S_\ell$ given the realized
                type sequence $(\theta_1, \ldots, \theta_t)$.
 9:     else
10:         {Exploitation} Recommend the best explored action for
            agent type $\theta_t$.
```

JM: Marginal distribution means is the distribution of one action not the distribution of joint actions.]

4.2 Main Recommendation Policy

Algorithm 2 is the main procedure of our recommendation policy. It consists of two parts: *exploration*, which explores all the eventually-explorable actions, and *exploitation*, which simply recommends the best explored action for a given type. The exploration part proceeds in phases. In each phase ℓ, each type θ gets a sequence of L_θ actions from MaxExplore using the data collected before this phase starts. The phase ends when every agent type θ has finished L_θ rounds. We pick parameter L_θ large enough so that the condition (5) is satisfied for all phases ℓ and all possible signals $S = S_\ell$. (Note that L_θ is finite because there are only finitely many such signals.) After $|\mathcal{A}| \cdot |\mathbf{\Theta}|$ phases, our recommendation policy enters the exploitation part. See Algorithm 2 for details.

There is a separate thread for each type θ, denoted thread(θ), which is called whenever an agent of this type shows up; see Algorithm 3. In a given phase ℓ, it recommends the L_θ actions computed by MaxExplore, then switches to the best explored action. The thread only uses the information collected before the current phase starts: the signal S_ℓ and signal structure $\mathcal{S}_\ell$.

Algorithm 3 Thread for agent type θ: thread(θ)

```
1: if this is the first call of thread($\theta$) of the current phase then
2:     Compute a list of $L_\theta$ actions $\alpha_\theta \leftarrow$ MaxExplore($\theta, S_\ell, \mathcal{S}_\ell$).
3:     Initialize the index of type $\theta$: $i_\theta \leftarrow 0$.
4: $i_\theta \leftarrow i_\theta + 1$.
5: if $i_\theta \leq L_\theta$ then
6:     Recommend action $\alpha_\theta[i_\theta]$.
7: else
8:     Recommend the best explored action of type $\theta$.
```

The BIC property follows easily from Claim 4.6. The key argument is that Algorithm 2 explores all eventually-explorable type-action pairs.

[jm: Reviewer 2: I did not follow the the statement and proof of Lemma 4.7. Is a phase equivalent to a round? If so, then why within each phase the algorithm interacts with agents for multiple rounds? If not, then why the lemma refers to both phase l and round l? Also π is the policy at round l, why it has history which is denoted by S'? The definition of H_t seems to imply that π is played every roundâĂę

JM: A phase is not a round. The answer to the question is that we are comparing our policy with l phases with other poilcy with l rounds.]

Lemma 4.7. *Fix phase $\ell > 0$ and the sequence of agent types $\theta_1, ..., \theta_T$. Assume Algorithm 2 has been running for at least $\min(l, |\mathcal{A}| \cdot |\Theta|)$ phases. For a given state ω, if type-action pair (θ, a) can be explored by some BIC recommendation policy π at round ℓ with positive probability, then such action is explored by Algorithm 2 by the end of phase $\min(l, |\mathcal{A}| \cdot |\Theta|)$ with probability 1.*

Proof. We prove this by induction on ℓ for $\ell \leq |\mathcal{A}| \cdot |\Theta|$. Base case $\ell = 1$ is trivial by Claim 4.6. Assuming the lemma is correct for $\ell - 1$, let's prove it's correct for ℓ.

Let $S = S_l$ be the signal of Algorithm 2 by the end of phase $\ell - 1$. Let S' be the history of π in the first $\ell - 1$ rounds. More precisely, $S' = (R, H_1, ..., H_{l-1})$, where R is the internal randomness of policy π, and $H_t = (\Theta_t, A_t, u(\Theta_t, A_t, \omega_0))$ is the type-action-reward triple in round t of policy π.

The proof plan is as follows. We first show that $I(S'; \omega_0|S) = 0$. Informally, this means the information collected in the first $l - 1$ phases of Algorithm 2 contains all the information S' has about the state w_0. After that, we will use the information monotonicity lemma to show that phase l of Algorithm 2 explores all the action-type pairs π might explore in round l.

First of all, we have

$$\begin{aligned} I(S'; \omega_0 \mid S) &= I(R, H_1, ..., H_{l-1}; \omega_0 \mid S) \\ &= I(R; \omega_0 \mid S) + I(H_1, ..., H_{l-1}; \omega_0 \mid S, R) \\ &= I(H_1, ..., H_{l-1}; \omega_0 \mid S, R). \end{aligned}$$

By the chain rule of mutual information, we have

$$\begin{aligned} &I(H_1, ..., H_{l-1}; \omega_0 \mid S, R) \\ &= I(H_1; \omega_0 \mid S, R) + \cdots + I(H_{l-1}; \omega_0 \mid S, R, H_1, ..., H_{l-2}). \end{aligned}$$

For all $t \in [l - 1]$, we have

$$\begin{aligned} &I(H_t; \omega_0 \mid S, R, H_1, ..., H_{t-1}) \\ =&I(\Theta_t, A_t, u(\Theta_t, A_t, \omega_0); \omega_0 \mid S, R, H_1, ..., H_{t-1}) \\ =&I(\Theta_t; \omega_0 \mid S, R, H_1, ..., H_{t-1}) \\ &+ I(A_t, u(\Theta_t, A_t, \omega_0); \omega_0 \mid S, R, H_1, ..., H_{t-1}, \Theta_t) \\ =&I(A_t, u(\Theta_t, A_t, \omega_0); \omega_0 \mid S, R, H_1, ..., H_{t-1}, \Theta_t). \end{aligned}$$

Notice that the suggested action A_t is a deterministic function of randomness of the recommendation policy R, history of previous rounds $H_1, ..., H_{t-1}$ and type in the current round Θ_t. Also notice that, by induction hypothesis, $u(\Theta_t, A_t, \omega_0)$ is a deterministic function of $S, R, H_1, ..., H_{t-1}, \Theta_t, A_t$. Therefore we have

$$I(H_t; \omega_0 \mid S, R, H_1, ..., H_{t-1}) = 0, \qquad \forall t \in [l - 1].$$

Then we get $I(S'; \omega_0 \mid S) = 0$.

By Lemma 4.4, we know that $\mathsf{EX}[\mathcal{S}'] \subseteq \mathsf{EX}[\mathcal{S}]$. For state ω, there exists a signal s' such that $\Pr[S' = s' \mid \omega_0 = \omega] > 0$ and $a \in \mathsf{EX}_{s'}[\mathcal{S}']$. Now let s be the realized value of S given $\omega_0 = \omega$, we know that $\Pr[S' = s' \mid S = s] > 0$, so $a \in \mathsf{EX}_s[\mathcal{S}]$. By Claim 4.6, we know that at least one agent of type θ in phase ℓ of Algorithm 2 will choose action a.

Now we consider the case when $\ell > |\mathcal{A}| \cdot |\Theta|$. Define Algorithm E to be the variant of Algorithm 2 such that it only does exploration (removing the if-condition and exploitation in Algorithm 2). For $\ell > |\mathcal{A}| \cdot |\Theta|$, the above induction proof still work for Algorithm E, i.e. for a given state ω, if an action a of type θ can be explored by a BIC recommendation policy π at round ℓ, then such action is guaranteed to be explored by Algorithm E by the end of phase ℓ. Now we are going to argue that Algorithm E won't explore any new action-type pairs after phase $|\mathcal{A}| \cdot |\Theta|$. Call a phase exploring if in that phase Algorithm E explores at least one new action-type pair. As there are $|\mathcal{A}| \cdot |\Theta|$ type-action pairs, Algorithm E can have at most $|\mathcal{A}| \cdot |\Theta|$ exploring phases. On the other hand, once Algorithm E has a phase that is not exploring, because the signal stays the same after that phase, all phases afterwards are not exploring. Therefore Algorithm E does not have any exploring phases after phase $|\mathcal{A}| \cdot |\Theta|$. For $\ell > |\mathcal{A}| \cdot |\Theta|$, the first $|\mathcal{A}| \cdot |\Theta|$ phases of Algorithm 2 explores the same set of type-action pairs as the first ℓ phases of Algorithm E. □

Proof of Theorem 4.1. Algorithm 2 is BIC by Claim 4.6. By Lemma 4.7, Algorithm 2 explores all the eventually-explorable type-actions pairs after $|\mathcal{A}| \cdot |\Theta|$ phases. After that, for each agent type θ, Algorithm 2 always recommends the best explored action: $\arg\max_{a \in \mathcal{A}_{\omega,\theta}} u(\theta, a, \omega)$. Therefore Algorithm 2 gets reward OPT except rounds in the first $|\mathcal{A}| \cdot |\Theta|$ phases. It remains to prove that the expected number of rounds in exploration does not depend on the time horizon T. Let N_ℓ be the duration of phase ℓ. Recall that the phase ends as soon as each type has shown up at least L_θ times. It follows that $\mathbb{E}[N_\ell] \leq \sum_{\theta \in \Theta} \frac{L_\theta}{\Pr[\theta]}$. So, one can take $C = |\mathcal{A}| \cdot |\Theta| \cdot \sum_{\theta \in \Theta} \frac{L_\theta}{\Pr[\theta]}$. □

[jm: Above Section 3.3, I also did not follow the proof of Theorem 3.1. It claims that "Algorithm 2 always recommends the best explored action" after some phases, why? Is this a high-probability guarantee or 100 percent guarantee? It seems to me that this should have been a high-probability guarantee since there is a strictly positive probability that the agent comes at each round has exactly the same agent type and in this case you can never learn the other agent's best action. If so, what's the algorithm's success probability? If not, why the above is not a concern? At the end, I also did not see why C depends linearly on the expectation of N_l (the duration of phase l). Do we need to guarantee that each type has shown up at least L_θ times FOR SURE, which may never happen if you get unlucky?

JM: It seems the reviewer does not understand we are proving expected regret bound. The example mentioned by the reviewer will stop us from getting enough phases. But once we have enough phases, it is 100 percent guarantee.]

4.3 Extension to Reported Types

Let us sketch how to extend our ideas for public types to handle the case of reported types. We'd like to simulate the recommendation policy for public types, call it π_{pub}. We simulate it separately for the exploration part and the exploitation part. The exploitation part is fairly easy: we provide a menu that recommends the best explored action for each agent types. In the exploration part, in each round t we guess the agent type to be $\hat{\theta}_t$, with equal probability among all types. [jm: Reviewer 2: why not guess a type according to its probability? This seems more intuitive to me. JM: notice our goal is to explore type certain rounds. Guessing a type according to its probability will make rare types even more rare to appear and be guessed correctly. And this is against us.] The idea is to simulate π_{pub} only in *lucky rounds* when we guess correctly, *i.e.,* $\hat{\theta}_t = \theta_t$. Thus, in each round t we simulate the ℓ_t-th round of π_{pub}, where ℓ_t is the number of lucky rounds before round t.

In each round t of exploration, we suggest the following menu. For type $\hat{\theta}_t$, we recommend the same action as π_{pub} would recommend for this type in the ℓ_t-th round, namely $\hat{a}_t = \pi^{\ell_t}_{\text{pub}}(\hat{\theta}_t)$. For any other type, we recommend the action which has the best expected reward given the "common knowledge" (information available before round 1) and the action $\hat{a}_t$. This is to ensure that in a lucky round, the menu does not convey any information beyond action $\hat{a}_t$. When we receive the reported type, we can check whether our guess was correct. If so, we input the type-action-reward triple back to π_{pub}. Else, we ignore this round, as if it never happened.

Thus, our recommendation policy eventually explores the same type-action pairs as π_{pub}. The expected number of rounds increases by the factor of $|\Theta|$. Thus, we have the following theorem.

Theorem 4.8. *Consider Bayesian Exploration with reported types. There exists a BIC recommendation policy whose expected total reward is at least* $(T - C) \cdot \mathtt{OPT}_{\text{pub}}$, *for some constant C that depends on the problem instance but not on T. This policy explores all type-action pairs that are eventually-explorable for a given state in the case of public types.*

5 BAYESIAN EXPLORATION WITH PRIVATE TYPES

Our recommendation policy for private types satisfies a relaxed version of the BIC property, called *δ-BIC*, where the right-hand side in (2) is $-\delta$ for some fixed $\delta > 0$. We assume a more permissive behavioral model in which agents follow recommendations of such policy.

The main result is as follows. (Throughout this section, $\mathtt{OPT} = \mathtt{OPT}_{\mathtt{pri}}$.)

Theorem 5.1. *Consider Bayesian Exploration with private types, and fix $\delta > 0$. There exists a δ-BIC recommendation policy with expected total reward at least* $(T - C \log T) \cdot \mathtt{OPT}$, *where C depends on the problem instance but not on time horizon T.*

Our recommendation policy and proofs have a similar structure as the ones for public types. The recommendation policy proceeds in phases: in each phase, it explores all menus that can be explored given the information collected so far. The crucial step in the proof is to show that:

(P1) the first l phases of our recommendation policy explore all the menus that could be possibly explored by the first l rounds of any BIC recommendation policy.

The new difficulty for private types comes from the fact that we are exploring menus instead of type-actions pairs, and we do not learn the reward of a particular type-action pair immediately. This is because a recommended menu may map several different types to the chosen action, so knowing the latter does not immediately reveal the agent's type. Moreover, the full "outcome" of a particular menu is a distribution over action-reward pairs, it is, in general, impossible to learn this outcome exactly in any finite number of rounds. Because of these issues, we cannot obtain Property (P1) exactly. Instead, we achieve an approximate version of this property, as long as we explore each menu enough times in each phase.

We then show that this approximate version of (P1) suffices to guarantee explorability, if we relax the incentives property of our policy from BIC to δ-BIC, for any fixed $\delta > 0$. In particular, we prove an approximate version of the information-monotonicity lemma (Lemma 4.4) which (given the approximate version of (P1)) ensures that our recommendation policy can explore all the menus that could be possibly explored by the first l rounds of any BIC recommendation policy.

5.1 Single-round Exploration

In this subsection, we consider a single round of the Bayesian exploration.

Definition 5.2. Consider a single-round of Bayesian exploration when the principal has signal S from signal structure $\mathcal{S}$. For any $\delta \geq 0$, a menu $m \in \mathcal{M}$ is called δ-signal-explorable, for a given signal s, if there exists a single-round δ-BIC recommendation policy π such that $\Pr[\pi(s) = m] > 0$. The set of all such menus is denoted as $\mathtt{EX}^{\delta}_{s}[\mathcal{S}]$. The δ-signal-explorable set is defined as $\mathtt{EX}^{\delta}[\mathcal{S}] = \mathtt{EX}^{\delta}_{S}[\mathcal{S}]$. We omit δ in $\mathtt{EX}^{\delta}[\mathcal{S}]$ when $\delta = 0$.

Approximate Information Monotonicity. In the following definition, we define a way to compare two signals approximately.

Definition 5.3. Let S and S' be two random variables. We say random variable S is α-approximately informative as random variable S' about state ω_0 if $I(S'; \omega_0 | S) = \alpha$.

Lemma 5.4. *Let S and S' be two random variables and $\mathcal{S}$ and $\mathcal{S}'$ be their signal structures. If S is $(\delta^2/8)$-approximately informative as S' about state ω_0 (i.e. $I(S'; \omega_0 | S) \leq \delta^2/8$), then $\mathtt{EX}_{s'}[\mathcal{S}'] \subseteq \mathtt{EX}^{\delta}_{s}[\mathcal{S}]$ for all s', s such that $\Pr[S = s, S' = s'] > 0$.*

Proof. We have

$$\sum_{s} \Pr[S = s] \cdot \mathbf{D}_{\text{KL}} \left(S'\omega_0 | S = s \| (S' | S = s) \times (\omega_0 | S = s) \right)$$
$$= I(S'; \omega_0 | S) \leq \delta^2/8.$$

By Pinsker's inequality, we have

$$
\begin{aligned}
&\sum_{s',\omega} \left|\Pr[S'=s',\omega_0=\omega|S=s]-\Pr[S'=s'|S=s]\cdot\Pr[\omega_0=\omega|S=s]\right| \\
&\cdot \sum_s \Pr[S=s] \\
\le& \sum_s \Pr[S=s]\cdot\sqrt{2\ln(2)\cdot \mathbf{D}_{\mathrm{KL}}\left(S'\omega_0|S=s \| (S'|S=s)\times(\omega_0|S=s)\right)} \\
\le& \sum_s \Pr[S=s]\cdot\sqrt{2\mathbf{D}_{\mathrm{KL}}\left(S'\omega_0|S=s \| (S'|S=s)\times(\omega_0|S=s)\right)} \\
\le& \sqrt{2\sum_s \Pr[S=s]\cdot \mathbf{D}_{\mathrm{KL}}\left(S'\omega_0|S=s \| (S'|S=s)\times(\omega_0|S=s)\right)} \\
\le& \delta/2.
\end{aligned}
$$

Consider any BIC recommendation policy π' for signal structure $\mathcal{S}'$. We construct π for signature structure $\mathcal{S}$ by setting $\Pr[\pi(s)=m]=\sum_{s'}\Pr[\pi'(s')=m]\cdot Pr[S'=s'|S=s]$.

Now we check π is δ-BIC. For any $m,m'\in\mathcal{M}$ and $\theta\in\mathbf{\Theta}$,

$$
\begin{aligned}
&\sum_{\omega,s}\Pr[\omega_0=\omega]\cdot\Pr[S=s|\omega_0=\omega] \\
&\cdot\left(u(\theta,m(\theta),\omega)-u(\theta,m'(\theta),\omega)\right)\cdot\Pr[\pi(s)=m] \\
=& \sum_{\omega,s,s'}\Pr[\omega_0=\omega,S=s]\cdot\Pr[S'=s'|S=s]\cdot\Pr[\pi'(s')=m] \\
&\cdot\left(u(\theta,m(\theta),\omega)-u(\theta,m'(\theta),\omega)\right) \\
\ge& \sum_{\omega,s,s'}\Pr[\omega_0=\omega,S=s,S'=s']\cdot\Pr[\pi'(s')=m] \\
&\cdot\left(u(\theta,m(\theta),\omega)-u(\theta,m'(\theta),\omega)\right) \\
&-2\cdot\sum_{\omega,s,s'}|\Pr[\omega_0=\omega,S=s]\cdot\Pr[S'=s'|S=s] \\
&-\Pr[\omega_0=\omega,S=s,S'=s']| \\
=& \sum_{\omega,s'}\Pr[\omega_0=\omega,S'=s']\cdot\Pr[\pi'(s')=m] \\
&\cdot\left(u(\theta,m(\theta),\omega)-u(\theta,m'(\theta),\omega)\right) \\
&-2\cdot\sum_s\Pr[S=s]\cdot\sum_{s',\omega}|\Pr[S'=s',\omega_0=\omega|S=s] \\
&-\Pr[S'=s'|S=s]\cdot\Pr[\omega_0=\omega|S=s]| \\
\ge& 0-2\cdot\frac{\delta}{2} \\
=& -\delta
\end{aligned}
$$

We also have for any s',s,m such that $Pr[S'=s',S=s]>0$ and $\Pr[\pi'(s')=m]>0$, we have $\Pr[\pi(s)=m]>0$. This implies $\mathsf{EX}_{s'}[\mathcal{S}']\subseteq\mathsf{EX}_s^{\delta}[\mathcal{S}]$. □

Max-Support Policy. We can solve the following LP to check whether a particular menu $m_0\in\mathcal{A}$ is signal-explorable given a particular realized signal $s_0\in\mathcal{X}$. In this LP, we represent a policy π as a set of numbers $x_{m,s}=\Pr[\pi(s)=m]$, for each menu $m\in\mathcal{M}$ and each feasible signal $s\in\mathcal{X}$.

maximize x_{m_0,s_0}
subject to:

$$
\begin{aligned}
&\sum_{\omega\in\Omega,s\in\mathcal{X}}\Pr[\omega]\cdot\Pr[s|\omega]\cdot\left(u(\theta,m(\theta),\omega)-u(\theta,m'(\theta),\omega)+\delta\right) \\
&\cdot x_{m,s'}\ge 0 && \forall m,m'\in\mathcal{M},\theta\in\mathbf{\Theta} \\
&\sum_{m\in\mathcal{M}}x_{m,s}=1, && \forall s\in\mathcal{X} \\
&x_{m,s}\ge 0, && \forall s\in\mathcal{X},m\in\mathcal{M}
\end{aligned}
$$

Since the constraints in this LP characterize any δ-BIC recommendation policy, it follows that menu m_0 is δ-signal-explorable given realized signal s_0 if and only if the LP has a positive solution. If such solution exists, define recommendation policy $\pi=\pi^{m_0,s_0}$ by setting $\Pr[\pi(s)=m]=x_{m,s}$ for all $m\in\mathcal{M},s\in\mathcal{X}$. Then this is a δ-BIC recommendation policy such that $\Pr[\pi(s_0)=m_0]>0$.

Definition 5.5. Given a signal structure $\mathcal{S}$, a recommendation policy π is called the δ-max-support policy if $\forall s\in\mathcal{X}$ and δ-signal-explorable menu $m\in\mathcal{M}$ given s, $\Pr[\pi(s)=m]>0$.

It is easy to see that we obtain δ-max-support recommendation policy by averaging the $\pi^{m,s}$ policies define above. Specifically, the following policy is a δ-BIC and δ-max-support policy.

$$
\pi^{max}=\frac{1}{|\mathcal{X}|}\sum_{s\in\mathcal{X}}\frac{1}{|\mathsf{EX}_s^{\delta}[\mathcal{S}]|}\sum_{m\in\mathsf{EX}_s^{\delta}[\mathcal{S}]}\pi^{m,s}. \tag{6}
$$

Maximal Exploration. Let us design a subroutine, called MaxExplore, which outputs a sequence of L menus. We are going to assume $L\ge\max_{m,s}\frac{B_m(\gamma_0)}{\Pr[\pi^{max}(s)=m]}$. γ_0 is defined in Algorithm 5 of Section 5.2. B_m is defined in Lemma 5.7.

The goal of this subroutine MaxExplore is to make sure that for any signal-explorable menu m, m shows up at least $B_m(\gamma_0)$ times in the sequence with probability exactly 1. On the other hand, we want that the menu of each specific location in the sequence has marginal distribution same as π^{max}.

Algorithm 4 Subroutine MaxExplore

```
1:  Input: signal $S$, signal structure $\mathcal{S}$.
2:  Output: a list of menus $\mu$
3:  Compute $\pi^{max}$ as per (6).
4:  Initialize $Res = L$.
5:  for each menu $m \in \mathcal{M}$ do
6:      $C_m \leftarrow L \cdot \Pr[\pi^{max}(S) = m]$.
7:      Add $\lfloor C_m \rfloor$ copies of menu $m$ into list $\mu$.
8:      $Res \leftarrow Res - \lfloor C_m \rfloor$.
9:      $p^{Res}(m) \leftarrow C_m - \lfloor C_m \rfloor$
10: $p^{Res}(m) \leftarrow p^{Res}(m)/Res, \forall m \in \mathcal{M}$.
11: Sample $Res$ many menus from distribution according to $p^{Res}$
    independently and add these menus into $\mu$.
12: Randomly permute the menus in $\mu$.
13: return $\mu$.
```

Similarly as the MaxExplore in Section 4, we have the following claim.

Claim 5.6. *Given realized signal S, MaxExplore outputs a sequence of L menus. Each menu in the sequence marginally distributed as π^{max}. For any menu m such that $\Pr[\pi^{max} = m] > 0$, m shows up in the sequence at least $B_m(\gamma_0)$ times with probability exactly 1. MaxExplore runs in time polynomial in L, $|\mathcal{M}|$, $|\Omega|$, $|\mathcal{X}|$ (size of the support of the signal).*

Menu Exploration. Given a menu m, a action-reward pair will be revealed to the algorithm after the round. Assuming the agent is following the menu, such action-reward pair is called a sample of the menu m. We use D_m to denote the distribution of the samples. D_m is a random variable depending on the state ω_0. For a fixed state ω, we use $D_m(\omega)$ to denote the distribution of the samples of menu m.

Lemma 5.7. *For any $\gamma > 0$, we can compute Δ_m which is a function of $B_m(\gamma) = O\left(\ln\left(\frac{1}{\gamma}\right)\right)$ samples of menu m such that for any state ω,*

$$\Pr[\Delta_m \neq D_m(\omega) | \omega_0 = \omega] \leq \gamma.$$

Proof. Let U be the union of the support of $D_m(\omega)$ for all $\omega \in \Omega$. For each $u \in U$ (u is just a sample of the menu), define $q(u,\omega) = \Pr_{v\sim D_m(\omega)}[v = u]$. Let δ_m be small enough such that for all ω, ω' with $D_m(\omega) \neq D_m(\omega')$, there exists $u \in U$, such that $|q(u,\omega) - q(u,\omega')| > \delta_m$.

Now we compute Δ_m as following: Take $B_m(\gamma) = \frac{2}{\delta_m^2}\ln\left(\frac{2|U|}{\gamma}\right)$ samples and set $\hat{q}(u)$ as the empirical frequency of seeing u. And set Δ_m to be some $D_m(\omega)$ such that for all $u \in U$, $|q(u,\omega) - \hat{q}(u)| \leq \delta_m/2$. Notice that if such ω exists, Δ_m will be unique. If no ω satisfies this, just pick Δ_m to be an arbitrary $D_m(\omega)$.

Now let's analyze $\Pr[\Delta_m \neq D_m(\omega)]$. Let's fix the state $\omega_0 = \omega$. By Chernoff bound, for each $u \in U$,

$$\Pr[|q(u,\omega) - \hat{q}(u)| > \delta_m/2] \leq 2\exp\left(-2\cdot\left(\frac{\delta_m}{2}\right)^2 \cdot B_m(\gamma)\right) \leq \frac{\gamma}{|U|}.$$

By union bound, with probability at least $1-\gamma$, we have for all $u \in U$, $|q(u,\omega) - \hat{q}(u)| \leq \delta_m/2$. This implies $\Delta_m = D_m(\omega)$. □

5.2 Main Recommendation Policy

In this subsection, we develop our main recommendation policy, Algorithm 5 (see pseudo-code), which explores all the eventually-explorable menus and then recommends the agents the best menu given all history. We pick L to be at least $\max_{m,s:\Pr[\pi(s)=m]>0} \frac{B_m(\gamma_0)}{\Pr[\pi(s)=m]}$ for all π that might be chosen as π^{max} by Algorithm 5.

First of all, it's easy to check by Claim 5.6 that for each agent, it is δ-BIC to follow the recommended action if previous agents all follow the recommended actions. Therefore we have the following claim.

Claim 5.8. *Algorithm 5 is δ-BIC.*

Lemma 5.9. *For any $l > 0$, assume Algorithm 5 has at least $\min(l, |\mathcal{M}|)$ phases. For a given state ω, if a menu m can be explored by a BIC recommendation policy π at round l (i.e. $\Pr[\pi^l = m] > 0$), then such menu is guaranteed to be explored B_m times by Algorithm 5 by the end of phase $\min(l, |\mathcal{M}|)$.*

Algorithm 5 Main procedure for private types

```
1:  Initial signal $S_1 = \mathcal{S}_1 = \perp$.
2:  Set $\gamma_1 = \min\left(\frac{\delta^2}{16|\mathcal{M}|\log(|\Omega|)}, \left(\frac{\delta^2}{32|M|}\right)^2\right)$ and $\gamma_2 = \frac{1}{T|\mathcal{M}|}$ and $\gamma_0 = \min(\gamma_1, \gamma_2)$.
3:  Initial phase count $l = 1$.
4:  for $t = 1$ to $T$ do
5:      if $l \leq |\mathcal{M}|$ then
6:          Exploration:
7:          if $t \equiv 1 \pmod L$ then
8:              Start a new phase:
9:              Use the current $S_l$ and $\mathcal{S}_l$ to compute a list of $L$ menus $\mu \leftarrow$ MaxExplore($S_l, \mathcal{S}_l$).
10:         Suggest menu $\mu[(t-1) \mod L + 1]$ to the agent.
11:         if $t \equiv 0 \pmod L$ then
12:             End of a phase:
13:             For each explored menu $m$ in the previous phase, use $B_m(\gamma_1)$ samples to compute $\Delta_m$ stated in Lemma 5.7.
14:             If there does not exist a state $w$ which is consistent with $\Delta_m$ ($\Delta_m = D_m(\omega)$) for all explored menu $m$, pick an arbitrary state $\omega$ and set $\Delta_m \leftarrow D_m(\omega)$ for all explored menu $m$. This step just make sure the number of signals is bounded by $|\Omega|$.
15:             $l \leftarrow l + 1$.
16:             Set $S_l$ to be the collection of $\Delta_m$'s for all explored menu $m$.
17:     else
18:         Exploitation:
19:         If this is the first exploitation round, for each explored menu $m$ in the exploration, use $B_m(\gamma_2)$ samples to compute $\Delta_m$ stated in Lemma 5.7. Set $S_l$ to be the collection of $\Delta_m$'s for all explored menu $m$.
20:         Suggest the menu which consists of the best action of each type conditioned on $S_l$ and the prior.
```

Proof. The proof is similar to Lemma 4.7. We prove by induction on l for $l \leq |\mathcal{M}|$.

Let S be the signal of Algorithm 5 in phase l. Let S' be the history of π in the first $l-1$ rounds. More precisely, $S' = R, H_1, ..., H_{l-1}$. Here R is the internal randomness of π and
$H_t = (M_t, A_t, u(\Theta_t, M_t(\Theta_t), \omega_0))$ is the menu and the action-reward pair in round t of π.

Let $\mathcal{M}'$ to be the set of menus explored in the first $l-1$ phases of Algorithm 5. By the induction hypothesis, we have $\forall t \in [l-1]$, $M_t \subseteq \mathcal{M}'$.

First of all, we have

$$\begin{aligned}&I(S';\omega_0|S) = I(R, H_1, ..., H_{l-1};\omega_0|S)\\ =&I(R;\omega_0|S) + I(H_1, ..., H_{l-1};\omega_0|S, R) = I(H_1, ..., H_{l-1};\omega_0|S, R).\end{aligned}$$

By the chain rule of mutual information, we have

$$\begin{aligned}&I(H_1, ..., H_{l-1};\omega_0|S, R)\\ =&I(H_1;\omega_0|S, R) + I(H_2;\omega_0|S, R, H_1) + \cdots + I(H_{l-1};\omega_0|S, R, H_1, ..., H_{l-2}).\end{aligned}$$

For all $t \in [l-1]$, we have

$$\begin{aligned}&I(H_t;\omega_0|S,R,H_1,...,H_{t-1})\\ =&I(M_t,A_t,u(\Theta_t,M_t(\Theta_t),\omega_0);\omega_0|S,R,H_1,...,H_{t-1})\\ =&I(A_t,u(\Theta_t,M_t(\Theta_t),\omega_0);\omega_0|S,R,H_1,...,H_{t-1},M_t)\\ \le&I(D_{M_t};\omega_0|S,R,H_1,...,H_{t-1},M_t).\end{aligned}$$

The second last step comes from the fact that M_t is a deterministic function of $R, H_1, ..., H_{t-1}$. The last step comes from the fact that $(A_t, u(\Theta_t, M_t(\Theta_t), \omega_0))$ is independent with ω_0 given D_{M_t}.

Then we have

$$\begin{aligned}&I(D_{M_t};\omega_0|S,R,H_1,...,H_{t-1},M_t)\\ =&\sum_{m\in\mathcal{M}'}\Pr[M_t=m]\cdot I(D_m;\omega_0|S,R,H_1,...,H_{t-1},M_t=m)\\ \le&\sum_{m\in M'}\Pr[M_t=m]\cdot I(D_m;\omega_0|\Delta_m,M_t=m).\\ \le&\sum_{m\in M'}\Pr[M_t=m]\cdot H(D_m|\Delta_m,M_t=m).\end{aligned}$$

The last step comes from the fact that $I(D_m; (S\backslash\Delta_m), R, H_1, ..., H_{t-1}|\omega_0, \Delta_m, M_t = m) = 0$. By Lemma 5.7, we know that $\Pr[D_m \neq \Delta_m|M_t = m] \le \gamma_1$. By Fano's inequality, we have

$$\begin{aligned}&H(D_m|\Delta_m,M_t=m)\le H(\gamma_1)+\gamma_1\log(|\Omega|-1)\\ &\le 2\sqrt{\gamma_1}+\gamma_1\log(|\Omega|-1)\le\frac{\delta^2}{16|\mathcal{M}|}+\frac{\delta^2}{16|\mathcal{M}|}=\frac{\delta^2}{8|\mathcal{M}|}.\end{aligned}$$

Therefore we have

$$I(H_t;\omega_0|S,R,H_1,...,H_{t-1})\le\frac{\delta^2}{8|\mathcal{M}|},\forall t\in[l-1].$$

Then we get

$$I(S';\omega_0|S)\le\delta^2/8.$$

By Lemma 5.4, we know that $\mathsf{EX}_{s'}[\mathcal{S}'] \subseteq \mathsf{EX}_s^\delta[\mathcal{S}]$. By Claim 5.6, we know that phase l will explore menu m at least $B_m(\gamma_0)$ times.

When $l > |\mathcal{M}|$, the proof follows from the same argument as the last paragraph of the proof of Lemma 4.7. □

Corollary 5.10 (Restatement of Theorem 5.1). *For any $\delta > 0$, we have a δ-BIC recommendation policy of T rounds with expected total reward at least $(T - C \cdot \log(T)) \cdot \mathsf{OPT}$ for some constant C which does not depend on T.*

Proof. First of all, by Claim 5.8, Algorithm 5 is δ-BIC.

By Lemma 5.9, for each state ω, Algorithm 5 explores all the eventually-explorable menus (i.e. $\mathcal{M}_\omega$) by the end of $|\mathcal{M}|$ phases.

After that, by Lemma 5.7 and $\gamma_2 = \frac{1}{T|\mathcal{M}|}$, for a fixed state ω, we know that with probability $1-1/T$, $\delta_m = D_m$ for all $m \in \mathcal{M}_\omega$. In this case, the agent of type θ gets expected reward at least $u(\theta, m^*(\theta), \omega)$ where menu $m^* = \arg\max_{m\in\mathcal{M}_\omega} \sum_{\theta\in\Theta} \Pr[\theta] \cdot u(\theta, m(\theta), \omega)$. Taking average over types, the expected reward per round should be at least $(1 - 1/T) \cdot \max_{m\in\mathcal{M}_\omega} \sum_{\theta\in\Theta} \Pr[\theta] \cdot u(\theta, m(\theta), \omega)$.

We know that the expected number of rounds of the first $|\mathcal{M}|$ phases is $|\mathcal{M}| \cdot L = O(\ln(T))$. Therefore, Algorithm 5 has expected total reward at least $T \cdot \mathsf{OPT} - T \cdot (1/T) - O(\ln(T)) = T \cdot \mathsf{OPT} - O(\ln(T))$. □

A BASICS OF INFORMATION THEORY

We briefly review some standard facts and definitions from information theory which are used in proofs. For a more detailed introduction, see [12]. Throughout, X, Y, Z, W are random variables that take values in an arbitrary domain (not necessarily $\mathbb{R}$).

Entropy. The fundamental notion is *entropy* of a random variable. In particular, if X has finite support, its entropy is defined as

$$H(X)=-\textstyle\sum_x p(x)\cdot\log p(x),\quad\text{where } p(x)=\Pr[X=x].$$

(Throughout this paper, we use log to refer to the base 2 logarithm and use ln to refer to the natural logarithm.) If X is drawn from Bernoulli distribution with $\mathbb{E}[X] = p$, then

$$H(p)=-(p\log p+(1-p)(\log(1-p)).$$

The conditional entropy of X given event E is the entropy of the conditional distribution $(X|E)$:

$$H(X|E)=-\textstyle\sum_x p(x)\cdot\log p(x),\quad\text{where } p(x)=\Pr[X=x|E].$$

The *conditional entropy* of X given Y is

$$H(X|Y):=\mathbb{E}_y[H(X|Y=y)]=\textstyle\sum_y\Pr[Y=y]\cdot H(X|Y=y).$$

Note that $H(X|Y) = H(X)$ if X and Y are independent.

We are sometimes interested in the entropy of a tuple of random variables, such as (X, Y, Z). To simplify notation, we will write $H(X, Y, Z)$ instead $H((X, Y, Z))$, and similarly in other information-theoretic notation. With this ado, we can formulate the *Chain Rule* for entropy:

$$H(X,Y)=H(X)+H(Y|X).\tag{7}$$

We also use the following fundamental fact about entropy:

Lemma A.1 (Fano's Inequality). *Let $X, Y, \hat{X}$ be random variables such that $\hat{X}$ is a deterministic function of Y. (Informally, $\hat{X}$ is an approximate version of X derived from signal Y.) Let $E = \{\hat{X} \neq X\}$ be the "error event". Then*

$$H(X|Y)\le H(E)+\Pr[E]\cdot(\log(|\mathcal{X}|-1),$$

where $\mathcal{X}$ denotes the support set of X.

Mutual information. The *mutual information* between X and Y is

$$I(X;Y):=H(X)-H(X|Y)=H(Y)-H(Y|X).$$

The *conditional mutual information* between X and Y given Z is

$$I(X;Y|Z):=H(X|Z)-H(X|Y,Z)=H(Y|Z)-H(Y|X,Z).$$

Note that $I(X; Y|Z) = I(X; Y)$ if X and Z are conditionally independent given Y, and Y and Z are conditionally independent given X.

Some of the fundamental properties of conditional mutual information are as follows:

$$I(X,Y;Z|W)=I(X;Z|W)+I(Y;Z|W,X)\tag{8}$$

$$I(X;Y|Z)\ge I(X;Y|Z,W)\qquad\text{if } I(Y;W|X,Z)=0\tag{9}$$

$$I(X;Y|Z)\le I(X;Y|Z,W)\qquad\text{if } I(Y;W|Z)=0\tag{10}$$

KL-divergence. The *Kullback-Leibler divergence* (a.k.a., *KL-divergence*) between random variables X and Y is defined as

$$\mathbf{D}_{\mathsf{KL}}(X\|Y) = \sum_x \Pr[X=x]\cdot\log\left(\frac{\Pr[X=x]}{\Pr[Y=x]}\right).$$

Note that the definition is not symmetric, in the sense that in general $\mathbf{D}_{\mathsf{KL}}(X\|Y) \neq \mathbf{D}_{\mathsf{KL}}(Y\|X)$.

KL-divergence can be related to conditional mutual information as follows:

$$\begin{aligned} I(X;Y|Z) &= \mathbb{E}_{x,z}\left[\ \mathbf{D}_{\mathsf{KL}}((Y|X=x,Z=z)\|(Y|Z=z))\ \right] \\ &= \sum_{x,z}\Pr[X=x,Z=z]\ \ \mathbf{D}_{\mathsf{KL}}((Y|X=x,Z=z)\|(Y|Z=z)). \end{aligned} \tag{11}$$

Here $(Y|E)$ denotes the conditional distribution of Y given event E.

We also use *Pinsker Inequality*:

$$\sum_x |\Pr[X=x]-\Pr[Y=x]| \leq \sqrt{2\ln(2)\,\mathbf{D}_{\mathsf{KL}}(X\|Y)}. \tag{12}$$

REFERENCES

[1] Susan Athey and Ilya Segal. 2013. An Efficient Dynamic Mechanism. *Econometrica* 81, 6 (Nov. 2013), 2463–2485. A preliminary version has been available as a working paper since 2007.

[2] Moshe Babaioff, Robert Kleinberg, and Aleksandrs Slivkins. 2015. Truthful Mechanisms with Implicit Payment Computation. *J. of the ACM* 62, 2 (2015), 10. Subsumes the conference papers in *ACM EC 2010* and *ACM EC 2013*.

[3] Moshe Babaioff, Yogeshwer Sharma, and Aleksandrs Slivkins. 2014. Characterizing Truthful Multi-armed Bandit Mechanisms. *SIAM J. on Computing (SICOMP)* 43, 1 (2014), 194–230. Preliminary version in *10th ACM EC*, 2009.

[4] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2018. Bandits with Knapsacks. *J. of the ACM* 65, 3 (2018). Preliminary version in *FOCS 2013*.

[5] Gal Bahar, Rann Smorodinsky, and Moshe Tennenholtz. 2016. Economic Recommendation Systems. In *16th ACM Conf. on Electronic Commerce (EC)*.

[6] Dirk Bergemann and Juuso Välimäki. 2010. The Dynamic Pivot Mechanism. *Econometrica* 78, 2 (2010), 771–789. Preliminary versions have been available since 2006, as *Cowles Foundation Discussion Papers* #1584 (2006), #1616 (2007) and #1672(2008).

[7] Omar Besbes and Assaf Zeevi. 2009. Dynamic Pricing Without Knowing the Demand Function: Risk Bounds and Near-Optimal Algorithms. *Operations Research* 57 (2009), 1407–1420. Issue 6.

[8] Kostas Bimpikis, Yiangos Papanastasiou, and Nicos Savva. 2018. Crowdsourcing Exploration. *Management Science* (2018). Forthcoming. Published online as *Articles in Advance* in April 2017.

[9] Patrick Bolton and Christopher Harris. 1999. Strategic Experimentation. *Econometrica* 67, 2 (1999), 349–374.

[10] Sébastien Bubeck and Nicolo Cesa-Bianchi. 2012. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning* 5, 1 (2012).

[11] Yeon-Koo Che and Johannes Hörner. 2018. Optimal design for social learning. *Quarterly Journal of Economics* (2018). Forthcoming. First published draft: 2013.

[12] I. Csiszar and J. Körner. 2011. *Information theory: coding theorems for discrete memoryless systems.* Cambridge University Press.

[13] Nikhil Devanur and Sham M. Kakade. 2009. The Price of Truthfulness for Pay-Per-Click Auctions. In *10th ACM Conf. on Electronic Commerce (EC)*. 99–106.

[14] Peter Frazier, David Kempe, Jon M. Kleinberg, and Robert Kleinberg. 2014. Incentivizing exploration. In *ACM Conf. on Economics and Computation (ACM EC)*. 5–22.

[15] Arpita Ghosh and Patrick Hummel. 2013. Learning and incentives in user-generated content: multi-armed bandits with endogenous arms. In *Innovations in Theoretical Computer Science Conf. (ITCS)*. 233–246.

[16] John Gittins, Kevin Glazebrook, and Richard Weber. 2011. *Multi-Armed Bandit Allocation Indices.* John Wiley & Sons.

[17] Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. 2016. Adaptive Contract Design for Crowdsourcing Markets: Bandit Algorithms for Repeated Principal-Agent Problems. *J. of Artificial Intelligence Research* 55 (2016), 317–359. Preliminary version appeared in *ACM EC 2014*.

[18] Sham M. Kakade, Ilan Lobel, and Hamid Nazerzadeh. 2013. Optimal Dynamic Mechanism Design and the Virtual-Pivot Mechanism. *Operations Research* 61, 4 (2013), 837–854.

[19] Emir Kamenica and Matthew Gentzkow. 2011. Bayesian Persuasion. *American Economic Review* 101, 6 (2011), 2590–2615.

[20] Godfrey Keller, Sven Rady, and Martin Cripps. 2005. Strategic Experimentation with Exponential Bandits. *Econometrica* 73, 1 (2005), 39–68.

[21] Robert D. Kleinberg and Frank T. Leighton. 2003. The Value of Knowing a Demand Curve: Bounds on Regret for Online Posted-Price Auctions. In *IEEE Symp. on Foundations of Computer Science (FOCS)*.

[22] Robert D. Kleinberg, Bo Waggoner, and E. Glen Weyl. 2016. Descending Price Optimally Coordinates Search. Working paper. Preliminary version in *ACM EC 2016*.

[23] Ilan Kremer, Yishay Mansour, and Motty Perry. 2014. Implementing the âĂIJWisdom of the CrowdâĂİ. *J. of Political Economy* 122 (2014), 988–1012. Issue 5. Preliminary version in *ACM EC 2014*.

[24] Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. 2015. Bayesian Incentive-Compatible Bandit Exploration. In *15th ACM Conf. on Economics and Computation (ACM EC)*.

[25] Yishay Mansour, Aleksandrs Slivkins, Vasilis Syrgkanis, and Steven Wu. 2016. Bayesian Exploration: Incentivizing Exploration in Bayesian Games. In *16th ACM Conf. on Economics and Computation (ACM EC)*.

[26] Yishay Mansour, Aleksandrs Slivkins, Vasilis Syrgkanis, and Steven Wu. 2018. Bayesian Exploration: Incentivizing Exploration in Bayesian Games. Working paper. Available at `https://arxiv.org/abs/1602.07570`. Preliminary version in *ACM EC 2016*.

[27] Sven Schmit and Carlos Riquelme. 2018. Human Interaction with Recommendation Systems. In *Intl. Conf. on Artificial Intelligence and Statistics (AISTATS)*. 862–870.

[28] Adish Singla and Andreas Krause. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *22nd Intl. World Wide Web Conf. (WWW)*. 1167–1178.