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, users' 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 [**?**]. 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.

Bayesian Exploration with Heterogeneous Agents

Nicole [ni: HATES GITHUB] Immorlica Jieming Mao
Aleksandrs Slivkins Zhiwei Steven Wu

November 4, 2018

1 Introduction

Recommendation systems are ubiquitous in online markets (*e.g.,* Netflix for movies, Amazon for products, Yelp for restaurants, etc.), high-quality recommendations being a crucial part of their value proposition. A typical recommendation system encourages its users to submit feedback on their experiences, and aggregates this feedback in order to provide better recommendations in the future. 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" of insufficiently known alternatives and "exploitation" of the information acquired so far. Designing algorithms to trade off these two objectives is a well-researched subject in machine learning and operations research. However, a given user who decides to "explore" typically suffers all the downside of this decision, whereas the upside (improved recommendations) is spread over many users in the future. Therefore, users' incentives are skewed in favor of exploitation. As a result, observations may be collected at a slower rate, and suffer from selection bias (*e.g.,* ratings of a particular movie may mostly come from people who like this type of movies). Moreover, in some natural but idealized examples (*e.g.,* [**?**, **?**]) optimal recommendations are never found because they are never explored.

Thus, we have a problem of *incentivizing exploration*. Providing monetary incentives can be financially or technologically unfeasible, and relying on voluntary exploration can lead to selection biases. A recent line of work, started by [**?**], relies on the inherent *information asymmetry* between the recommendation system and a user. These papers posit a simple model, termed *Bayesian Exploration* in [**?**]. 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" [**?**].

Our scope. We depart from prior work on Bayesian Exploration in that we allow agents with heterogenous preferences. 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. [●] as deleted here

Explorability. 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. The set of explorable pairs is interesting in its own right as they bound the welfare of a setting. 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.

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 find that for each problem instance the explorable set stays the same if we transition from public types to reported types, and can only become smaller if we transition from reported types to private types. We provide a concrete example when the latter transition makes a huge difference. Second, we 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 make the support set larger, then the explorable set can only become larger. In other words, *diversity of agent types helps exploration.* We provide a concrete example when the explorable set

[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.

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. Intuitively, with private types, diversity muddles the information available to the principal making it harder to learn about the state of the world, whereas for public types diversity helps the principal refine her belief about the state.

Our techniques. As a warm-up, we first develop a recommendation policy for public types. In the long run, our policy matches the benchmark of "best explorable action". [•] 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. Our policy proceeds in phases: in each phase, we explore all actions for each type that can be explored using information available at the start of the phase.

as deleted here

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*: the set of all explorable type-action pairs can only increase if the principal has more information.

As our main contribution, we develop a policy for private types. In this model, recommending one particular action to the current agent is not very meaningful because the agents' type is not 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.

In the reported types model, we face a similar issue, but achieve a much stronger result: we design a policy which matches our public-types benchmark in the long run. 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.

Related work. The problem of Bayesian Exploration was introduced in [?]. The special case of homogenous agents has been largely resolved, in terms of the optimal policy for two actions [?], explorability [?], and regret minimization for stochastic utilities [?]. [?] also consider an extension to public types, under a very strong assumption geared to ensure that all type-action pairs are explorable. [?] 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). [?] 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. [?] consider a basic setting of Bayesian Exploration, but with time-discounted utilities. [?] allow monetary incentives. [?] posit a continuous information flow and a continuum of agents. [?] propose a mechanism to coordinate costly exploration decisions in social learning. [?] 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 [?, ?] under the name *strategic experimentation*.

Exploration-exploitation tradeoff received much attention over the past decades, usually under the rubric of "multi-armed bandits"; see [?, ?] 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 [?], 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 [?, ?, ?], dynamic auctions [?, ?, ?], pay-per-click ad auctions [?, ?, ?], and human computation [?, ?, ?].

2 Model and Preliminaries

Bayesian Exploration is a game between a principal and T agents. The game consists of T rounds. Each round $t \in [T]$ proceeds as follows: a new agent t arrives, receives a message m_t from the principal, chooses an action a_t from a fixed action space $\mathcal{A}$, and collects a reward $r_t \in [0,1]$ that is immediately observed by the principal. Each agent t has a *type* $\theta_t \in \mathbf{\Theta}$, drawn independently from a fixed distribution $\mathcal{D}$, and an *action space* $\mathcal{A}$ (same for all agents). There is uncertainty, captured by a "state of nature" $\omega \in \mathbf{\Omega}$, henceforth simply the *state*, drawn from a Bayesian prior $\mathcal{P}$ at the beginning of time and fixed across rounds. The *reward* $r_t = u(\theta_t, a_t, \omega) \in [0,1]$ of agent t is determined by its type θ_t of agent t, the action $a_t \in \mathcal{A}$ chosen this agent, and the state ω, for some fixed and deterministic *reward function* $u : \mathbf{\Theta} \times \mathcal{A} \times \mathbf{\Omega} \to [0,1]$. Principal's messages m_t are generated according to a randomized online algorithm π termed "recommendation policy". Thus, an *instance* of Bayesian Exploration consists of the time horizon T, the sets $\mathcal{A}, \mathbf{\Theta}, \mathbf{\Omega}$, the type distribution $\mathcal{D}$, the prior $\mathcal{P}$, and the reward function u.

The knowledge structure is as follows. The type distribution $\mathcal{D}$, the Bayesian prior $\mathcal{P}$, the reward function u, and the recommendation policy are common knowledge. Each agent t knows her own type θ_t, and observes nothing else except the message m_t. 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*).

Let H_t denote the *history* observed by the principal at round t, immediately before it chooses its message m_t. Hence, it equals $\{(r_1,\theta_1),\ldots,(r_{t-1},\theta_{t-1}),r_t\}$ 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.[2] Formally, this is the input to the recommendation policy in each round t. Borrowing terminology from the Bayesian Persuasion literature, we will often refer to the history as the *signal.* We denote the set of all possible histories (signals) at time t by $\mathcal{H}_t$.

The recommendation policy π, the type distribution $\mathcal{D}$, the state distribution $\mathcal{P}$, and the reward function u induce a joint distribution $\mathcal{D}(\mathbf{\Omega},\mathcal{H}_t)$ over states and histories, henceforth called the *signal structure* at round t. Note that it is known to agent t.

We are ready to state agents' decision model. Each agent t, given the message m_t, chooses an action a_t so as to maximize her *Bayesian-expected reward*

$$\mathbb{E}[r_t] \equiv \mathop{\mathbb{E}}_{(\omega,H_t)\sim\mathcal{D}(\mathbf{\Omega},\mathcal{H}_t)} \left[u(\theta_t,a_t,\omega)\ \mid m_t\sim\pi(H_t)\right]. \tag{1}$$

Given the instance of Bayesian Exploration, the goal of the principal is to choose a policy π that maximizes (Bayesian-expected) *total* reward, *i.e.,* $\sum_{t=1}^{T}\mathbb{E}[r_t]$.[3]

We assume that the sets $\mathcal{A}$, $\mathbf{\Theta}$ and $\mathbf{\Omega}$ 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 $\Pr[\theta_t=\theta]$.

Bayesian-incentive compatibility. For public types, we assume the message m_t in each round is a recommended action $a\in\mathcal{A}$ which, for convenience, we sometimes write as $m_t(\theta_t)$. For private and reported types, we assume that the message m_t in each round is a *menu* mapping types to actions, i.e., $m_t:\mathbf{\Theta}\to\mathcal{A}$. We further assume π is Bayesian incentive-compatible.

Definition 2.1. Let $\mathcal{E}_t$ be the event that the agents have followed principal's recommendations before round t, *i.e.,* $a_s=m_s(\theta_s)$ for all rounds $s<t$. The recommendation policy π is *Bayesian incentive compatible* (*BIC*) if for all rounds t and messages m such that

$$\Pr_{(\omega,H_t)\sim\mathcal{D}(\mathbf{\Omega},\mathcal{H}_t)}[m=\pi(H_t)\ \mid\ \mathcal{E}_t]>0,$$

it holds that for all types θ and arms a,

$$\mathbb{E}\left[\ u(\theta,m(\theta),\omega)-u(\theta,a,\omega)\ \mid\ m_t=m,\mathcal{E}_t\ \right]\geq 0, \tag{2}$$

where the expectation is over $(\omega,H_t)\sim\mathcal{D}(\mathbf{\Omega},\mathcal{H}_t)$.

[2]For randomized policies, the history also contains policy's random seed in each round.

[3]While the principal must commit to the policy given only the problem instance, the policy itself observes the history and thus can adapt recommendations to inferences about the state based on the history. See Example 3.2.

The above assumptions are without loss of generality, by a suitable version of Myerson's "revelation principle".

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 $\mathcal{M}_\omega$.

Our benchmark is the best eventually-explorable recommendation for each type. For public and private types, resp., this is

$$\texttt{OPT}_{\texttt{pub}} = \sum_{\theta \in \mathbf{\Theta}, \omega \in \mathbf{\Omega}} \Pr[\omega] \cdot \Pr[\theta] \cdot \max_{a \in \mathcal{A}_{\omega,\theta}} u(\theta, a, \omega). \tag{3}$$

$$\texttt{OPT}_{\texttt{pri}} = \sum_{\omega \in \mathbf{\Omega}} \Pr[\omega] \cdot \max_{m \in \mathcal{M}_\omega} \sum_{\theta \in \mathbf{\Theta}} \Pr[\theta] \cdot u(\theta, m(\theta), \omega). \tag{4}$$

We have $\texttt{OPT}_{\texttt{pub}} \geq \texttt{OPT}_{\texttt{pri}}$, 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 $\texttt{OPT}_{\texttt{pub}} > \texttt{OPT}_{\texttt{pri}}$.

3 Comparative Statics

We discuss how the set of all eventually-explorable type-action pairs (*explorable set*) is affected by the model choice and the diversity of types. The explorable set is all information that can possibly be learned in the public-types model. All else equal, settings with larger explorable set have greater or equal total expected reward, both in benchmark (3) and in our approximation guarantees. For private types, the exploration set provides an "upper bound" on the information available to the principal, because the principal does not directly observe the agent types. [●]

as deleted here

Explorability and the model choice. Fix an instance of Bayesian exploration. Let $\mathcal{A}^{\texttt{pub}}_\omega$ and $\mathcal{A}^{\texttt{pri}}_\omega$ be the explorable set for a given state ω, for public and private types, respectively.[4] We will show in Section 4.3 that the explorable set for reported types is $\mathcal{A}^{\texttt{pub}}_\omega$, too.

Claim 3.1. $\mathcal{A}^{\texttt{pri}}_\omega \subseteq \mathcal{A}^{\texttt{pub}}_\omega$.

The idea is that one can simulate any BIC recommendation policy for private types with a BIC recommendation policy for public types; we omit the details.

Interestingly, $\mathcal{A}^{\texttt{pri}}_\omega$ can in fact be a *strict* subset of $\mathcal{A}^{\texttt{pub}}_\omega$:

[4]Equivalently, $\mathcal{A}^{\texttt{pri}}_\omega$ is 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.

Example 3.2. There are two states, two types and two actions: $\mathbf{\Omega} = \mathbf{\Theta} = \mathcal{A} = \{0, 1\}$. 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 $\omega = 1$.

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

Proof. Action 0 is preferred by both types initially. 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. $\square$

Explorability and diversity of agent types. Fix an instance of Bayesian exploration with type distribution $\mathcal{D}$. We consider how the explorable set changes if we modify the type distribution $\mathcal{D}$ in this instance to some other distribution $\mathcal{D}'$. Let $\mathcal{A}_\omega$ and $\mathcal{A}'_\omega$ be the corresponding explorable sets, for each state ω.

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

[as: restored the old version of the claim because I like it better and it takes less space.]

Claim 3.4. *Consider Bayesian Exploration with public types. Then:*

(a) if $\texttt{support}(\mathcal{D}) = \texttt{support}(\mathcal{D}')$ then $\mathcal{A}_\omega = \mathcal{A}'_\omega$.

(b) if $\texttt{support}(\mathcal{D}) \subset \texttt{support}(\mathcal{D}')$ then $\mathcal{A}_\omega \subseteq \mathcal{A}'_\omega$.

Proof Sketch. Consider public types (the case of reported types then follows by arguments in Section 4.3). Let π be a BIC recommendation policy for the instance with type distribution $\mathcal{D}$ and suppose π eventually explores type-action pairs $\mathcal{A}_\omega$ for this instance and state ω. Consider the instance with type distribution $\mathcal{D}'$. Extend π to a policy π' as follows: let T' be the subsequence of T

for which $\mathcal{D}(\theta_t) > 0$. If $t \notin T'$, then recommend the action a that maximizes agent t's Bayesian-expected reward. If $t \in T'$, then consider the sub-history $H \equiv H_t^{T'}$ restricted to T' and recommend action $a \sim \pi(H)$. Then π' is BIC for the instance with type distribution $\mathcal{D}'$. Furthermore, π' eventually explores the same set of type-action pairs $\mathcal{A}_\omega$ for this modified instance as well (and possibly more) as every history that occurs with positive probability in the original instance occurs as a sub-history in the modified instance with positive probability as well. **NSI: check this.** □

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 Public Types

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

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

Signal and explorability. We first analyze what actions can be explored by a BIC policy in a single round t of Bayesian exploration for public types, as a function of the history. Throughout, we suppress θ and t from our notation. Let S be a random variable equal to the history at round t (referred to as a *signal* throughout this section), s be a realization of S, and $\mathcal{S} = \mathcal{D}(\Omega, \mathcal{H})$ be the signal structure. Note different policies induce different histories and hence different signal structures. Thus it will be important to be explicit about the signal structure throughout this section.

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 realized signal s* if there exists a 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*, denoted $\texttt{EX}[\mathcal{S}]$, is the random subset of actions $\texttt{EX}_S[\mathcal{S}]$.

[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).]

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 $I(S'; \omega \mid S) = 0$.

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,

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

Therefore π' being BIC implies that π is also BIC. Indeed, for any $a, a' \in \mathcal{A}$ and $\theta \in \mathbf{\Theta}$, 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 $s \in \mathcal{X}$.

$$\begin{array}{ll}
\textbf{maximize } x_{a_0,s_0} & \\
\textbf{subject to:} & \\
\sum_{\omega\in\mathbf{\Omega},s\in\mathcal{X}} \Pr[\omega]\cdot\Pr[s\mid\omega]\cdot & \\
(u(\theta,a,\omega)-u(\theta,a',\omega))\cdot x_{a,s} \geq 0 & \forall a,a'\in\mathcal{A} \\
\sum_{a\in\mathcal{A}} x_{a,s} = 1, & \forall s\in\mathcal{X} \\
x_{a,s} \geq 0, & \forall s\in\mathcal{X}, a\in\mathcal{A}
\end{array}$$

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:

$$\pi^{\max} = \frac{1}{|\mathcal{X}|}\sum_{s\in\mathcal{X}}\frac{1}{|\mathtt{EX}_s[\mathcal{S}]|}\sum_{a\in\mathtt{EX}_s[\mathcal{S}]}\pi^{a,s}. \tag{5}$$

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

$$L_\theta \geq \max_{(a,s)\in\mathcal{A}\times\mathcal{X} \text{ with } \Pr[\pi^{\max}(s)=a]\neq 0}\frac{1}{\Pr[\pi^{\max}(s)=a]}. \tag{6}$$

This step is essentially from [?]; 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 $p^{\mathtt{res}}(a) = \frac{C_a - \lfloor C_a \rfloor}{L_\theta - \sum_a \lfloor C_a \rfloor}$. 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 (5)
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:      $p^{\mathtt{res}}(a) \leftarrow C_a - \lfloor C_a \rfloor$
10: $p^{\mathtt{res}}(a) \leftarrow p^{\mathtt{res}}(a)/Res$, $\forall a \in \mathcal{A}$.
11: Sample $Res$ many actions from distribution according to $p^{\mathtt{res}}$ 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, $|\mathcal{A}|$, $|\mathbf{\Omega}|$ 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? Actions? But π^{max} is a randomized map from S to actions, how could it be a marginal distribution of actions? Do you mean $\pi^{max}(S)$?

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 (6) is satisfied for all phases l and all possible signals $S = S_l$. (Note that L_θ is finite because there

Algorithm 2 Main procedure for public types

```
1: Initialization: signal $S_1 = \mathcal{S}_1 = \perp$, phase count $l = 1$, index $i_\theta = 0$ for each
   type $\theta \in \mathbf{\Theta}$.
2: for rounds $t = 1$ to $T$ do
3:   if $l \leq |\mathcal{A}| \cdot |\mathbf{\Theta}|$ 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: $l \leftarrow l + 1$.
7:       Let $S_l$ be the signal for phase $l$: the set of all observed type-action-
         reward triples.
8:       Let $\mathcal{S}_l$ be the signal structure for $S_l$ given the realized type sequence
         $(\theta_1, ..., \theta_t)$.
9:   else
10:    {Exploitation} Recommend the best explored action for agent type $\theta_t$.
```

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 $\mathcal{S}_l$.

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($\theta, S_l, \mathcal{S}_l$).
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 round

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 $l > 0$ and the sequence of agent types $\theta_1, ..., \theta_T$. Assume Algorithm 2 has been running for at least* $\min(l, |\mathcal{A}| \cdot |\mathbf{\Theta}|)$ *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* $\min(l, |\mathcal{A}| \cdot |\mathbf{\Theta}|)$ *with probability* 1.

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

Let $S = S_l$ be the signal of Algorithm 2 by the end of phase $l-1$. Let S' be the history of π in the first $l-1$ 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

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

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

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

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 $\texttt{EX}[\mathcal{S}'] \subseteq \texttt{EX}[\mathcal{S}]$. For state ω, there exists a signal s' such that $\Pr[S' = s' \mid \omega_0 = \omega] > 0$ and $a \in \texttt{EX}_{s'}[\mathcal{S}']$. Now let s be the realized value of S given $\omega_0 = \omega$, we know that $\Pr[S' = s' \mid S = s] > 0$, so $a \in \texttt{EX}_s[\mathcal{S}]$. 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 consider the case when $l > |\mathcal{A}| \cdot |\mathbf{\Theta}|$. Define ALG to be the variant of Algorithm 2 such that it only does exploration (removing the if-condition and exploitation in Algorithm 2). For $l > |\mathcal{A}| \cdot |\mathbf{\Theta}|$, the above induction proof still work for ALG, 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 ALG by the end of phase l. Now we are going to argue that ALG won't explore any new action-type pairs after phase $|\mathcal{A}| \cdot |\mathbf{\Theta}|$. Call a phase exploring if in that phase ALG explores at least one new action-type pair. As there are $|\mathcal{A}| \cdot |\mathbf{\Theta}|$ type-action pairs, ALG can have at most $|\mathcal{A}| \cdot |\mathbf{\Theta}|$ exploring phases. On the other hand, once ALG has a phase that is not exploring, because the signal stays the same after that phase, all phases afterwards are not exploring. So, ALG does not have any exploring phases after phase $|\mathcal{A}| \cdot |\mathbf{\Theta}|$. For $l > |\mathcal{A}| \cdot |\mathbf{\Theta}|$, the first $|\mathcal{A}| \cdot |\mathbf{\Theta}|$ phases of Algorithm 2 explores the same set of type-action pairs as the first l phases of ALG. □

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 $|\mathcal{A}| \cdot |\mathbf{\Theta}|$ 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 $|\mathcal{A}| \cdot |\mathbf{\Theta}|$ 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 $\mathbb{E}[N_l] \leq \sum_{\theta \in \mathbf{\Theta}} \frac{L_\theta}{\Pr[\theta]}$. So, one can take $C = |\mathcal{A}| \cdot |\mathbf{\Theta}| \cdot \sum_{\theta \in \mathbf{\Theta}} \frac{L_\theta}{\Pr[\theta]}$. □

[jm: Reviewer 2: 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 $\pi_{\texttt{pub}}$. 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 $\pi_{\texttt{pub}}$ 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 $\pi_{\texttt{pub}}$, 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 $\pi_{\texttt{pub}}$ would recommend for this type in the l_t-th round, namely $\hat{a}_t = \pi^{l_t}_{\texttt{pub}}(\hat{\theta}_t)$. 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 $\pi_{\texttt{pub}}$. Else, we ignore this round, as if it never happened.

Thus, our recommendation policy eventually explores the same type-action pairs as $\pi_{\texttt{pub}}$. The expected number of rounds increases by the factor of $|\boldsymbol{\Theta}|$. 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* $(T - C) \cdot \texttt{OPT}_{\texttt{pub}}$*, 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 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 obey such policy.

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

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* $(T - C \log T) \cdot \texttt{OPT}$*, where* C *depends on the problem instance but not on time horizon* T*.*

[●] 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: as deleted here

(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 $\mathtt{EX}_s^\delta[\mathcal{S}]$. The δ-signal-explorable set is defined as $\mathtt{EX}^\delta[\mathcal{S}] = \mathtt{EX}_S^\delta[\mathcal{S}]$. 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 $\mathtt{EX}_{s'}[\mathcal{S}'] \subseteq \mathtt{EX}_s^\delta[\mathcal{S}]$ for all s', s such that* $\Pr[S = s, S' = s'] > 0$.

Proof. For each signal realization s, denote

$$D_s = \mathbf{D}_{\text{KL}} \left(((S', \omega_0) \mid S = s) \quad \| \quad (S'|S = s) \times (\omega_0 \mid S = s) \right).$$

We have $\sum_s \Pr[S = s] \cdot D = I(S'; \omega_0|S) \leq \delta^2/8$.

By Pinsker's inequality, we have [as: Jieming: you said the next eqn is not quite right!]

$$\begin{aligned}
&\sum_{s',\omega} |\Pr[S' = s', \omega_0 = \omega|S = s] - \Pr[S' = s'|S = s] \cdot \Pr[\omega_0 = \omega|S = s]| \\
&\qquad \cdot \sum_s \Pr[S = s] \\
&\quad \leq \textstyle\sum_s \Pr[S = s] \cdot \sqrt{2\ln(2) \cdot D_s} \\
&\quad \leq \sqrt{2 \textstyle\sum_s \Pr[S = s] \cdot D_s} \leq \delta/2.
\end{aligned}$$

Consider any BIC recommendation policy π' for signal structure $\mathcal{S}'$. We construct π for signature structure $\mathcal{S}$ by setting

$$\Pr[\pi(s) = m] = \textstyle\sum_{s'} \Pr[\pi'(s') = m] \cdot Pr[S' = s'|S = s].$$

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

$$\begin{aligned}
&\sum_{\omega,s} \Pr[\omega_0 = \omega] \cdot \Pr[S = s|\omega_0 = \omega] \\
&\qquad \cdot (u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)) \cdot \Pr[\pi(s) = m] \\
&\quad = \sum_{\omega,s,s'} \Pr[\omega_0 = \omega, S = s] \cdot \Pr[S' = s'|S = s] \cdot \Pr[\pi'(s') = m] \\
&\qquad\quad \cdot (u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)) \\
&\quad \geq \sum_{\omega,s,s'} \Pr[\omega_0 = \omega, S = s, S' = s'] \cdot \Pr[\pi'(s') = m] \\
&\qquad\qquad \cdot (u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)) \\
&\qquad\qquad - 2 \cdot \sum_{\omega,s,s'} |\Pr[\omega_0 = \omega, S = s] \cdot \Pr[S' = s'|S = s] \\
&\qquad\qquad - \Pr[\omega_0 = \omega, S = s, S' = s']| \\
&\quad = \sum_{\omega,s'} \Pr[\omega_0 = \omega, S' = s'] \cdot \Pr[\pi'(s') = m] \\
&\qquad\qquad \cdot (u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)) \\
&\qquad\qquad - 2 \cdot \sum_s \Pr[S = s] \cdot \sum_{s',\omega} |\Pr[S' = s', \omega_0 = \omega|S = s] \\
&\qquad\qquad - \Pr[S' = s'|S = s] \cdot \Pr[\omega_0 = \omega|S = s]| \\
&\quad \geq 0 - 2 \cdot t\frac{\delta}{2} = \; -\delta.
\end{aligned}$$

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 $\mathtt{EX}_{s'}[\mathcal{S}'] \subseteq \mathtt{EX}_s^\delta[\mathcal{S}]$. □

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 $s \in \mathcal{X}$.

$$\begin{aligned}
&\textbf{maximize } x_{m_0,s_0}\\
&\textbf{subject to:}\\
&\sum_{\omega\in\mathbf{\Omega}, s\in\mathcal{X}} \Pr[\omega]\cdot\Pr[s|\omega] \ \cdot (u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega) + \delta)\\
&\qquad\qquad \cdot x_{m,s'} \geq 0 \qquad \forall m, m' \in \mathcal{M}, \theta \in \mathbf{\Theta}\\
&\sum_{m\in\mathcal{M}} x_{m,s} = 1, \qquad\qquad \forall s \in \mathcal{X}\\
&x_{m,s} \geq 0, \qquad\qquad\qquad \forall s \in \mathcal{X}, m \in \mathcal{M}
\end{aligned}$$

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.

$$\pi^{max} = \frac{1}{|\mathcal{X}|}\sum_{s\in\mathcal{X}} \frac{1}{|\mathtt{EX}_s^\delta[\mathcal{S}]|} \sum_{m\in\mathtt{EX}_s^\delta[\mathcal{S}]} \pi^{m,s}. \tag{7}$$

Maximal Exploration. Let us design a subroutine, called MaxExplore, which outputs a sequence of L menus. We are going to assume $L \geq \max_{m,s} \frac{B_m(\gamma_0)}{\Pr[\pi^{max}(s)=m]}$. 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 (7).
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 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, $|\mathcal{M}|$, $|\mathbf{\Omega}|$, $|\mathcal{X}|$ (size of the support of the signal).*

Menu Exploration. Given a menu m, an 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. [●] For a fixed state ω, let $D_m(\omega)$ denote the distribution of the samples of menu m. as deleted here

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 $\omega \in \mathbf{\Omega}$. 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 $u \in U$,

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

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.

It is 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 $\min(l, |\mathcal{M}|)$.*

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 $\forall t \in [l-1]$, $M_t \subseteq \mathcal{M}'$. Then:

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

By the chain rule of mutual information, we have

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

Algorithm 5 Main procedure for private types

```
1: Initial signal $S_1 = \mathcal{S}_1 = \perp$.
2: Set $\gamma_1 = \min\left(\frac{\delta^2}{16|\mathcal{M}|\log(|\mathbf{\Omega}|)}, \left(\frac{\delta^2}{32|M|}\right)^2\right)$ 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 $|\mathbf{\Omega}|$.
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.
```

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

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

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

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

The last step comes from the fact that

$$I(D_m;(S\backslash\Delta_m),R,H_1,...,H_{t-1} \mid \omega_0,\Delta_m,M_t=m)=0.$$

By Lemma 5.7, we know that $\Pr[D_m \neq \Delta_m \mid M_t = m] \le \gamma_1$. By Fano's inequality, we have

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

Therefore we have

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

Then we get $I(S';\omega_0 \mid S)\le \delta^2/8$.

By Lemma 5.4, we know that $\mathtt{EX}_{s'}[\mathcal{S}'] \subseteq \mathtt{EX}^{\delta}_{s}[\mathcal{S}]$. 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. □

[•] as deleted here

Proof of Theorem 5.1. 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 $m^* = \arg\max_{m\in\mathcal{M}_\omega}\sum_{\theta\in\mathbf{\Theta}}\Pr[\theta]\cdot u(\theta,m(\theta),\omega)$. Taking average over types, the expected reward per round should be at least $(1-1/T)\cdot\max_{m\in\mathcal{M}_\omega}\sum_{\theta\in\mathbf{\Theta}}\Pr[\theta]\cdot u(\theta,m(\theta),\omega)$.

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 $T\cdot\mathtt{OPT}-T\cdot(1/T)-O(\ln(T)) = T\cdot\mathtt{OPT}-O(\ln(T))$. □

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 [?]. 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

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

(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 $(X|E)$:

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

The *conditional entropy* of X given Y is

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

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 write $H(X, Y, Z)$ instead of $H((X, Y, Z))$, and similarly in other information-theoretic notation. Now, we formulate the *Chain Rule* for entropy:

$$H(X, Y) = H(X) + H(Y|X). \tag{8}$$

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.*[5] *Let $E = \{\hat{X} \neq X\}$ be the "error event". Then, letting $\mathcal{X}$ denote the support set of X,*

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

Mutual info. 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, Z are conditionally independent given Y, and Y, Z are conditionally independent given X.

[5]Informally, $\hat{X}$ is an approximate version of X derived from signal Y.

Important properties of conditional mutual information are:

$$I(X, Y; Z|W) = I(X; Z|W) + I(Y; Z|W, X) \tag{9}$$
$$I(X; Y|Z) \geq I(X; Y|Z, W) \qquad \text{if } I(Y; W|X, Z) = 0 \tag{10}$$
$$I(X; Y|Z) \leq I(X; Y|Z, W) \qquad \text{if } I(Y; W|Z) = 0 \tag{11}$$

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

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

Note that the definition is not symmetric, in the sense that in general $\mathbf{D}_{\mathtt{KL}}(X\|Y) \neq \mathbf{D}_{\mathtt{KL}}(Y\|X)$. KL-divergence can be related to conditional mutual information as follows:

$$\begin{aligned} I(X; Y|Z) &= \mathbb{E}_{x,z}\left[\ \mathbf{D}_{\mathtt{KL}}((Y|X = x, Z = z)\|(Y|Z = z))\ \right] \\ &= \sum_{x,z} \Pr[X = x, Z = z]\ \ \mathbf{D}_{\mathtt{KL}}((Y|X = x, Z = z)\|(Y|Z = z)). \end{aligned} \tag{12}$$

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

We also use *Pinsker Inequality*:

$$\textstyle\sum_x |\Pr[X = x] - \Pr[Y = x]| \leq \sqrt{2\ln(2)\,\mathbf{D}_{\mathtt{KL}}(X\|Y)}. \tag{13}$$