# Bayesian Exploration:<br>Incentivizing Exploration with Heterogeneous Agents

June 29, 2018

## 1 Introduction

### 1.1 Model

The Bayesian exploration consists of $T$ rounds. The participants are $T$ agents and a principal. Each agent only participates in one round.

The state of nature $\omega$ is drawn from a Bayesian prior distribution over a finite state space $\mathit{\Omega}$. We use $\Pr[\omega]$ to denote the probability that state $\omega$ is sampled. We use $\Omega$ as the random variable for the state.

In each round, a new agent comes and its type $\theta$ is sampled from a distribution over a finite type space $\mathit{\Theta}$. The type distribution is independent with the state distribution. We use $\Pr[\theta]$ to denote the probability type $\theta$ is sampled. We use $\Theta$ as the random variable for the type.

For an agent with type $\theta \in \mathit{\Theta}$, it can choose an action $a$ from action space $\mathcal{A}$ where $\mathcal{A}$ is a finite set. The utility of the agent is a deterministic function of its type and action, and the state of nature: $u(\theta, a, \omega)$. We also assume the utility is bounded in $[0, 1]$.

#### 1.1.1 Type Information

In terms of principal's knowledge of the agents' types, we consider three different models:

- **Public types:** Principal knows the types of agents.
- **Private types, communication allowed:** Only the agent knows its own type. The principal asks each agent to report its type.
- **Private types, communication not allowed:** Only the agent knows its own type. Each agent is not allowed to send any message to the principal.

## 2 Bayesian Exploration with Public Types

In this section, we show a BIC scheme for public types (as in Theorem 2.3). We formally state Theorem 2.3 in Section 2.1 and we prove it in Section 2.2, 2.3 and 2.4.

## 2.1 Explorability and Benchmark

In this subsection, we define the benchmark and state our main theorem (Theorem 2.3).

**Definition 2.1.** *An action $a$ of type $\theta$ is eventually-explorable, for a given state $\omega$, if there exists a BIC recommendation policy $\pi$ and some round $t$ such that* $\Pr[\pi^t(\theta) = a] > 0$*. The set of all such actions for state $\omega$ and type $\theta$ is denoted as $\mathcal{A}^{exp}_{\omega,\theta}$.*

**Definition 2.2** (Benchmark)**.** *Define benchmark as*

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

Notice that for a given state $\omega$ and type $\theta$, any BIC recommendation policy can only recommend an action in $\mathcal{A}^{exp}_{\omega,\theta}$. We can simply get the following claim which says that no BIC recommendation policy can get expected per round reward better than the benchmark:

**Claim 2.1.** *Any BIC recommendation policy of $T$ rounds has expected total reward at most $T \cdot OPT$.*

On the other hand, we construct a BIC recommendation policy that nearly achives the benchmark. It's proved in the following subsections.

**Theorem 2.3.** *We have a BIC recommendation policy of $T$ rounds with expected total reward at least $(T - C) \cdot OPT$ for some constant $C$ which does not depend on $T$.*

## 2.2 Single-round Exploration

In this subsection, we consider a single round of the Bayesian exploration. As we only consider one round, we fix the agent's type in this round to be $\theta$.

A signal structure $\mathcal{S}$ can be specified by the signal support $\mathcal{X}$ and a joint distribution on $(\mathcal{X}, \Omega)$.

**Definition 2.4.** *Consider a single-round of Bayesian exploration when the principal has signal $S$ from signal strcture $\mathcal{S}$. An action $a \in \mathcal{A}$ is called signal-explorable, for a given signal $s$, if there exists a single-round BIC recommendation policy $\pi$ such that* $\Pr[\pi(s) = a] > 0$*. The set of all such actions is denoted as $EX_s[\mathcal{S}]$. The signal-explorable set is defined as $EX[\mathcal{S}] = EX_S[\mathcal{S}]$.*

Note that $EX_s[\mathcal{S}]$ is a fixed subset and $EX[\mathcal{S}]$ is a random variable whose realization is determined by the realization of signal $S$.

**Definition 2.5.** *We say random variable $S$ is at least as informative as random variable $S'$ about state $\Omega$ if $I(S'; \Omega | S) = 0$.*

**Lemma 2.6.** *Let $S$ and $S'$ be two random variables and $\mathcal{S}$ and $\mathcal{S}'$ be their signal structures. If $S$ is at least as informative as $S'$ about state $\Omega$ (i.e. $I(S'; \Omega | S) = 0$), then $EX_{s'}[\mathcal{S}'] \subseteq EX_s[\mathcal{S}]$ for all $s', s$ such that* $\Pr[S = s, S' = s'] > 0$.

*Proof.* Consider any BIC scheme $\pi'$ for signal structure $\mathcal{S}'$. We construct $\pi$ for signature structure $\mathcal{S}$ by setting $\Pr[\pi(s) = a] = \sum_{s'} \Pr[\pi'(s') = a] \cdot Pr[S' = s' | S = s]$. It's easy to check $\pi$ is BIC. 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 $EX_{s'}[\mathcal{S}'] \subseteq EX_s[\mathcal{S}]$. □

For a given signal $s \in \mathcal{X}$ and an action $a \in \mathcal{A}$, we solve the following LP to check if $a$ is signal-exlporable given signal $s$:

$$\begin{aligned}
\textbf{maximize} \quad & x_{a,s} \\
\textbf{subject to:} \quad & \sum_{\omega \in \Omega, s' \in \mathcal{X}} \Pr[\omega] \cdot \Pr[s'|\omega] \cdot \big(u(\theta, a', \omega) - u(\theta, a'', \omega)\big) \cdot x_{a',s'} \geq 0 \ \forall a', a'' \in \mathcal{A} \\
& \sum_{a' \in \mathcal{A}} x_{a',s'} = 1, \qquad \forall s' \in \mathcal{X} \\
& x_{a',s'} \geq 0, \qquad \forall s' \in \mathcal{X}, a' \in \mathcal{A}
\end{aligned}$$

Notice that the constraints in the above LP characterize any BIC recommendation scheme. Therefore we get the following claim.

**Claim 2.2.** *For a given signal $s \in \mathcal{X}$, an action $a \in \mathcal{A}$ is signal-explorable if and only if the above LP has a postive solution. When the LP has a positive solution, define $\pi^{a,s}$ as $\Pr[\pi^{a,s}(s') = a'] = x_{a',s'}, \forall a' \in \mathcal{A}, s' \in \mathcal{X}$. Then $\pi^{a,s}$ is a single-round BIC recommendation policy such that* $\Pr[\pi^{a,s}(s) = a] > 0$

**Definition 2.7** (Max-support policy)**.** *Given a signal structure $\mathcal{S}$, a BIC recommendation policy $\pi$ is called the max-support policy if $\forall s \in \mathcal{X}$ and signal-explorable action $a \in \mathcal{A}$ given $s$, $\Pr[\pi(s) = a] > 0$.*

We can get a max-support policy $\pi^{max}$ by averaging over BIC recommendation policies for signal explorable actions.

**Claim 2.3.** *The following $\pi^{max}$ is a BIC and max-support policy.*

$$\pi^{max} = \frac{1}{|\mathcal{X}|} \sum_{s \in \mathcal{X}} \frac{1}{|EX_s[\mathcal{S}]|} \sum_{a \in EX_s[S]} \pi^{a,s}.$$

## 2.3 MaxExplore

In this subsection, we show a subroutine MaxExplore which converts $\pi^{max}$ into a sequence of actions. We are going to assume $L_\theta \geq \max_{a,s:\Pr[\pi^{max}(s)=a] \neq 0} \frac{1}{\Pr[\pi^{max}(s)=a]}$.

**Algorithm 1** Subroutine MaxExplore

```
Input: type $\theta$, signal $S$ and signal strcuture $\mathcal{S}$.
Output: a list of actions $\alpha$
Compute $\pi^{max}$ as stated in Claim 2.3.
Initialize $Res = L_\theta$.
for each action $a \in \mathcal{A}$ do
    Add $\lfloor L_\theta \cdot \Pr[\pi^{max}(S) = a] \rfloor$ copies of action $a$ into list $\alpha$.
    $Res \leftarrow Res - \lfloor L_\theta \cdot \Pr[\pi^{max}(S) = a] \rfloor$.
    $p^{Res}(a) \leftarrow L_\theta \cdot \Pr[\pi^{max}(S) = a] - \lfloor L_\theta \cdot \Pr[\pi^{max}(S) = a] \rfloor$
end for
$p^{Res}(a) \leftarrow p^{Res}(a)/Res$, $\forall a \in \mathcal{A}$.
Sample $Res$ many actions from distribution according to $p^{Res}$ independently and add these
    actions into $\alpha$.
Randomly permute the actions in $\alpha$.
return $\alpha$.
```

Such conversion is exactly the same as [**?**], we have the following two claims.

**Claim 2.4.** *Given type $\theta$ and signal $S$, MaxExplore explores all the signal-explorable actions.*

**Claim 2.5.** *Signals and signal structures can be efficiently computed and represented.*

### 2.4 Main Scheme

In this subsection, we show our main scheme which explores all the eventually-explorable actions and then advises the agents to choose the best actions among eventually-explorable actions.

Algorithm 2 is the main procedure of our scheme. It globally divides rounds into epochs and runs a separate thread for each type (Sub-$\theta$ for type $\theta$). We pick $L_\theta$ to be at least $\max_{a,s:\Pr[\pi(s)=a]\neq 0} \frac{1}{\Pr[\pi(s)=a]}$ for all $\pi$ that might be chosen as $\pi^{max}$ in Algorithm 2.

**Algorithm 2** Main procedure for public types

```
Initial signal $S_1 = \mathcal{S}_1 = \perp$.
Initial epoch count $l = 1$.
for $t = 1$ to $T$ do
    Let $\theta_t$ to be the agent type of current round $t$. Run subroutine Sub-$\theta_t$.
    if every type has finished a complete phase in the current epoch then
        Start a new epoch:
        $l \leftarrow l + 1$.
        Let $S_l$ be the set of type-action-reward triples and $\mathcal{S}_l$ be the signal structure when fixing
            $\theta_1, ..., \theta_t$.
    end if
end for
```

Algorithm 3 describes the thread for type $\theta$. It is activated whenever the agent in the current round has type $\theta$. It divides the rounds of type $\theta$ into phases of fixed length ($L_\theta$). Each phase only uses the information collected before the current epoch starts.

**Algorithm 3** Subroutine for type $\theta$: Sub-$\theta$

```
for each call from Algorithm 2 do
    if previous phase has finished ($i_\theta > L_\theta$) or this is the first call then
        Start a new phase:
        if $l \leq |\mathcal{A}| \cdot |\Theta|$ then
            Use the current $S_l$ and $\mathcal{S}_l$ to compute a list of $L_\theta$ actions $\alpha_\theta \leftarrow$ MaxExplore($\theta, S_l, \mathcal{S}_l$).
        else
            Add $L_\theta$ copies of the best explored action of type $\theta$ according to signal $S$ into action list $\alpha_\theta$.
        end if
        Intialize the index of current phase $i_\theta \leftarrow 1$.
    end if
    Suggest action $\alpha_\theta[i_\theta]$ to the agent.
    $i_\theta \leftarrow i_\theta + 1$.
end for
```

The proof plan is to first give an upper bound on the expected number of rounds of an epoch in Lemma 2.8 and then show in Lemma 2.9 that Algorithm 2 explores all the explorable type-action pair in $|\mathcal{A}| \cdot |\Theta|$ epochs. We use these two lemmas to prove our main theorem (Theorem 2.3) in Corollary 2.10.

First of all, it's easy to check that for each agent, it is Bayesian incentive compatible to follow the recommended action if previous agents all follow the recommended actions. Therefore we have the following claim.

**Claim 2.6.** *The scheme in this section is BIC.*

**Lemma 2.8.** *For any $l > 0$, the expected number of rounds of the first $l$ epochs is at most $l \cdot \sum_{\theta \in \Theta} \frac{2L_\theta}{\Pr[\theta]}$.*

*Proof.* First of all, if a type $\theta$ shows up $2L_\theta$ times, type $\theta$ must have finished a complete phase in these rounds. Therefore, the expected number of rounds of an epoch is at most the expected number of rounds in which each type $\theta$ has shown up at least $2L_\theta$ times. The latter expectation can be easily bounded by $\sum_{\theta \in \Theta} \frac{2L_\theta}{\Pr[\theta]}$. Therefore, the expected number of rounds of the first $l$ epochs is at most $l \cdot \sum_{\theta \in \Theta} \frac{2L_\theta}{\Pr[\theta]}$. □

Notice that in Algorithm 2 the division of phases and epochs depends only on realized types $\theta_1, ..., \theta_T$ as each phase has a fixed length. We then have the following claim.

**Claim 2.7.** *Given $\theta_1, ..., \theta_T$ as the realized types in $T$ rounds, the division of phases and epochs in Algorithm 2 is fixed.*

**Lemma 2.9.** *For any $l > 0$ and a fixed sequence of types $\theta_1, ..., \theta_T$ as the types encountered by Algorithm 2, assume Algorithm 2 has at least $\min(l, |\mathcal{A}| \cdot |\Theta|)$ epochs. For a given state $\omega$, if an action $a$ of type $\theta$ can be explored by a BIC recommendation policy $\pi$ at round $l$ (i.e. $\Pr[\pi^l(\theta) = a] > 0$), then such action is guaranteed to be explored by Algorithm 2 by the end of epoch $\min(l, |\mathcal{A}| \cdot |\Theta|)$.*

*Proof.* We prove this by induction on $l$ for $l \leq |\mathcal{A}| \cdot |\Theta|$. Base case $l = 1$ is trivial by Claim 2.4. Assuming the lemma is correct for $l - 1$, let's prove it's correct for $l$.

Let $S$ be the history of Algorithm 2 by the end of epoch $l-1$. By the definition of Algorithm 2, $S = S_{l-1}|\theta_1, ..., \theta_T$. Let $S'$ be the history of $\pi$ in the first $l-1$ rounds. More precisely, $S' = R, H_1, ..., H_{l-1}$. Here $R$ is the internal randomness of scheme $\pi$ and $H_t = (\Theta_t, A_t, u(\Theta_t, A_t, \Omega))$ is the triple of type, action and reward in round $t$ of scheme $\pi$.

First of all, we have

$$I(S';\Omega|S) = I(R, H_1, ..., H_{l-1};\Omega|S) = I(R;\Omega|S) + I(H_1, ..., H_{l-1};\Omega|S, R) = I(H_1, ..., H_{l-1};\Omega|S, R).$$

By the chain rule of mutual information, we have

$$I(H_1, ..., H_{l-1};\Omega|S, R) = I(H_1;\Omega|S, R) + I(H_2;\Omega|S, R, H_1) + \cdots + I(H_{l-1};\Omega|S, R, H_1, ..., H_{l-2}).$$

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

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

Notice that $A_t$ is a deterministic function of $R, H_1, ..., H_{t-1}, \Theta_t$ because $A_t$ only depend on the current type $\Theta_t$. Also notice that, by induction hypothesis, $u(\Theta_t, A_t, \Omega)$ is a deterministic function of $|S, R, H_1, ..., H_{t-1}, \Theta_t, A_t$. Therefore we have

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

Then we get

$$I(S';\Omega|S) = 0.$$

By Lemma 2.6, we know that $EX[\mathcal{S}'] \subseteq EX[\mathcal{S}]$. For state $\omega$, there exists a signal $s'$ such that $\Pr[S' = s'|\Omega = \omega] > 0$ and $a \in EX_{s'}[\mathcal{S}']$. Now let $s$ be the realized value of $S$ given $\Omega = \omega$, we know that $\Pr[S' = s'|S = s] > 0$, so $a \in EX_s[\mathcal{S}]$. By Claim 2.4, we know that a complete phase of type $\theta$ in epoch $l$ of Algorithm 2 will choose action $a$ at least once.

Now we consider the case when $l > |\mathcal{A}| \cdot |\Theta|$. Define Algorithm $E$ to be the variant of Algorithm 2 in the sense that Algorithm $E$ always tries to run MaxExplore (i.e. in Sub-$\theta$, ignores if condition in step 5 and always runs MaxExplore ). For $l > |\mathcal{A}| \cdot |\Theta|$, the above induction proof still work on Algorithm $E$. Now notice if in some epoch, no unexplored type-action pair is played, Algorithm $E$'s signal won't change and after that Algorithm $E$ won't explore any unexplored type-action pair. As there are $|\mathcal{A}| \cdot |\Theta|$ type-action pairs, Algorithm $E$ only explores unexplored type-action pair in the first $|\mathcal{A}| \cdot |\Theta|$ epochs. Therefore, the first $\min(l, |\mathcal{A}| \cdot |\Theta|)$ epochs of Algorithm 2 explores the same set of type-action pair as the first $l$ epochs of Algorithm $E$. □

**Corollary 2.10** (Restatement of Theorem 2.3)**.** *We have a BIC recommendation policy (Algorithm 2) of $T$ rounds with expected total reward at least $(T - C) \cdot OPT$ for constant $C = |\mathcal{A}| \cdot |\Theta| \cdot \sum_{\theta \in \Theta} \frac{2L_\theta}{\Pr[\theta]}$.*

*Proof.* First of all, by Claim 2.6, Algorithm 2 is BIC.

By Lemma 2.9, for each state $\omega$, Algorithm 2 explores all the eventually explorable actions (i.e. $\mathcal{A}^{exp}_{\omega,\theta}$) for each type $\theta$ by the end of $|\mathcal{A}| \cdot |\Theta|$ epochs. After that, for each type $\theta$, Algorithm 2 will always recommend action $\arg\max_{a\in\mathcal{A}^{exp}_{\omega,\theta}} u(\theta, a, \omega)$. Therefore Algorithm 2 gets reward $OPT$ except rounds in first $|\mathcal{A}| \cdot |\Theta|$ epochs. By Lemma 2.8, we know that the expected number of rounds of the first $|\mathcal{A}| \cdot |\Theta|$ epochs is at most $C$. Therefore, Algorithm 2 has expected total reward at least $(T - C) \cdot OPT$.

□

## 3 Bayesian Exploration with Private Types and Communication

In this section, we show a BIC scheme for private types when communication is allowed. The main idea is to guess the type of each agent and then use the scheme in Section 2. In each round, we also ask the agent to report its type in the end. Since this report has nothing to do with the recommended actions received, the agent has no incentive to misreport its type. Our scheme only uses information collected in rounds when the guessed type equals to the actual type and therefore the expected length of each epoch becomes longer than the one in Section 2 (as shown in Lemma 3.1. Once the scheme explores all the explorable type-action pairs, it just shows agents the best action for each type among all explorable actions.

Algorithm 4 is the main procedure of the scheme. It is very similar to the main procedure in Section 2 (Algorithm 2). The only difference is that the scheme guesses types and only uses information collected in rounds when the scheme guesses the type correctly.

**Algorithm 4** Main procedure for private types

```
Initial signal $S_1 = \mathcal{S}_1 = \perp$.
Initial epoch count $l = 1$.
for $t = 1$ to $T$ do
    The principal receives information $I_t$ about $\theta_t$.
    Sample $\hat{\theta}_t$ from $\Theta$ uniformly. Run subroutine Private-Sub-$\hat{\theta}_t$.
    if every type has finished a complete phase in the current epoch then
        Start a new epoch:
        $l \leftarrow l + 1$.
        Let $S_l$ be the type-action-reward triples in rounds when $\bar{\theta}_\tau = \hat{\theta}_\tau$, and $\mathcal{S}_l$ be the signal structure.
    end if
end for
```

Algorithm 5 describes the thread when the algorithm guesses the type to be $\theta$. It is very similar to Algorithm 3 in Section 2. We use the same $L_\theta$ as Section 2.

**Algorithm 5** Subroutine for type $\theta$: Private-Sub-$\theta$

```
for each call from Algorithm 2 do
    if previous phase has finished ($i_\theta > L_\theta$) or this is the first call then
        Start a new phase:
        if $l \leq |\mathcal{A}| \cdot |\Theta|$ then
            Use the current $S_l$ and $\mathcal{S}_l$ to compute a list of $L_\theta$ actions $\alpha_\theta \leftarrow$ MaxExplore($\theta, S_l, \mathcal{S}_l$).
        else
            Add $L_\theta$ copies of the best explored action of type $\theta$ according to signal $S$ into action list $\alpha_\theta$.
        end if
        Intialize the index of current phase $i_\theta \leftarrow 1$.
    end if
    Recommend ($\alpha_\theta[i_\theta]$ for type $\theta$ and for other types, recommend the best actions given the agents' prior and the fact that ($\alpha_\theta[i_\theta]$ is recommended for type $\theta$ .
    Agent reports its type $\bar{\theta}$.
    if $\bar{\theta} = \theta$ then
        $i_\theta \leftarrow i_\theta + 1$.
    end if
end for
```

For similar reasons as Claim 2.6, we have the following claim.

**Claim 3.1.** *The scheme in this section is BIC.*

Finally, we prove the main theorem of this section (Theorem 3.3). It is based on Lemma 3.1 and Lemma 3.2.

**Lemma 3.1.** *For any $l > 0$, the expected number of rounds of the first $l$ epochs is at most $l \cdot \sum_{\theta\in\Theta} \frac{2 \cdot L_\theta \cdot |\Theta|}{\Pr[\theta]}$.*

*Proof.* First of all, if a type $\theta$ is guessed correctly $2L_\theta$ times, type $\theta$ must have finished a complete phase in these rounds. Therefore, the expected number of rounds of an epoch is at most the expected number of rounds in which each type $\theta$ is guessed correctly at least $2L_\theta$ times. The latter expectation can be easily bounded by $\sum_{\theta\in\Theta} \frac{2L_\theta \cdot |\Theta|}{\Pr[\theta]}$. Therefore, the expected number of rounds of the first $l$ epochs is at most $l \cdot \sum_{\theta\in\Theta} \frac{2L_\theta \cdot |\Theta|}{\Pr[\theta]}$. □

**Lemma 3.2.** *For each state $\omega$, Algorithm 4 explores all the eventually explorable actions (i.e. $\mathcal{A}^{exp}_{\omega,\theta}$) for each type $\theta$ by the end of $|\mathcal{A}| \cdot |\Theta|$ epochs.*

*Proof.* Fix state $\omega$. Consider any type sequence $\theta_1, \theta_2, ..., \theta_T$ that are the types encountered by Algorithm 4 in each round. And consider another type sequence $\bar{\theta}_1, \bar{\theta}_2, ..., \bar{\theta}_T$ that are the guesses by Algorithm 4. Assume Algorithm 4 finished the $(|\mathcal{A}| \cdot |\Theta|)$-th epoch after round $t$. Let $r_1 < r_2 < \cdots < r_\tau = t$ be the rounds that Algorithm 4 guesses the type correctly (i.e. $\theta = \bar{\theta}$) in the first $t$ rounds. Assume Algorithm 4 uses randomness $R$ for MaxExplore. Now consider Algorithm 2 also uses the same randomness $R$ for MaxExplore when facing types $\theta_{r_1}, \theta_{r_2}, ..., \theta_{r_\tau}$ in each round. By the definition of Algorithm 4 and Algorithm 2, it's not hard to check that the $i$-th round of Algorithm 2 and the $r_i$-th round of Algorithm 4 play the same action.

Since Algorithm 4 finishes $|\mathcal{A}| \cdot |\Theta|$ epochs at round $t$, Algorithm 2 also finishes $|\mathcal{A}| \cdot |\Theta|$ epochs at round $\tau$. By Lemma 2.9, we know Algorithm 2 has explored all the eventually explorable actions by the end of round $\tau$. Therefore, Algorithm 4 has explored all the eventually explorable actions by the end of round $t$ (i.e. the end of $|\mathcal{A}| \cdot |\Theta|$ epochs). □

**Theorem 3.3.** *We have a BIC recommendation policy of $T$ rounds with expected total reward at least $(T - C) \cdot OPT$ for some constant $C = |\mathcal{A}| \cdot |\Theta| \cdot \sum_{\theta \in \Theta} \frac{2L_\theta \cdot |\Theta|}{\Pr[\theta]}$.*

*Proof.* First of all, by Claim 3.1, Algorithm 4 is BIC.

By Lemma 3.2, for each state $\omega$, Algorithm 4 explores all the eventually explorable actions (i.e. $\mathcal{A}^{exp}_{\omega,\theta}$) for each type $\theta$ by the end of $|\mathcal{A}| \cdot |\Theta|$ epochs. After that, for each type $\theta$, Algorithm 4 will always recommend action $\max_{a \in \mathcal{A}^{exp}_{\omega,\theta}} u(\theta, a, \omega)$. Therefore Algorithm 4 gets reward $OPT$ except rounds in first $|\mathcal{A}| \cdot |\Theta|$ epochs. By Lemma 3.1, we know that the expected number of rounds of the first $|\mathcal{A}| \cdot |\Theta|$ epochs is at most $C$. Therefore, Algorithm 4 has expected total reward at least $(T - C) \cdot OPT$.

□

# 4 Bayesian Exploration with Private Types and No Communication

In this section, we show a $\delta$-BIC scheme when types are private and no communication is allowed (as in Theorem 4.4). We formally state Theorem 4.4 in Section 4.1 and we prove it in Section 4.2, 4.3, 4.4 and 4.5.

## 4.1 Explorability and Benchmark

In this subsection, we define the benchmark and state our main theorem (Theorem 4.4).

**Definition 4.1.** *A menu $m : \Theta \to \mathcal{A}$ is a mapping from the type space $\Theta$ to the action space $\mathcal{A}$. We use $\mathcal{M}$ to denote the set of all menus.*

**Claim 4.1.** *Each single round of a BIC scheme can be considered as a distribution of menus.*

**Definition 4.2.** *A menu $m$ is eventually-explorable, for a given state $\omega$, if there exists a BIC recommendation policy $\pi$ and some round $t$ such that $\Pr[\pi^t = m] > 0$. The set of all such menus is denoted as $\mathcal{M}^{exp}_\omega$.*

**Definition 4.3** (Benchmark)**.** *Define benchmark as*

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

**Claim 4.2.** *Any BIC recommendation policy of $T$ rounds has expected total reward at most $T \cdot OPT$.*

**Theorem 4.4.** *For any $\delta > 0$, we have a $\delta$-BIC recommendation policy of $T$ rounds with expected total reward at least $(T - C \cdot \log(T)) \cdot OPT$ for some constant $C$ which does not depend on $T$.*

## 4.2 Single-round Exploration

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

**Definition 4.5.** *Consider a single-round of Bayesian exploration when the principal has signal $S$ from signal structure $\mathcal{S}$. A menu $m \in \mathcal{M}$ is called signal-explorable, for a given signal $s$, if there exists a single-round $\delta$-BIC recommendation policy $\pi$ such that $\Pr[\pi(s) = m] > 0$. The set of all such actions is denoted as $EX_s^\delta[\mathcal{S}]$. The signal-explorable set is defined as $EX^\delta[\mathcal{S}] = EX_S^\delta[\mathcal{S}]$. We omit $\delta$ in $EX^\delta[\mathcal{S}]$ when $\delta = 0$.*

**Definition 4.6.** *We say random variable $S$ is $\alpha$-approximately informative as random variable $S'$ about state $\Omega$ if $I(S'; \Omega|S) = \alpha$.*

**Lemma 4.7** (Approximate Information Monotonicity)**.** *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 $\Omega$ (i.e. $I(S'; \Omega|S) \leq \delta^2/8$), then $EX_{s'}[\mathcal{S}'] \subseteq EX_s^\delta[\mathcal{S}]$ for all $s', s$ such that $\Pr[S = s, S' = s'] > 0$.*

*Proof.* We have

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

By Pinsker's inequality, we have

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

Consider any $\delta$-BIC scheme $\pi'$ for signal structure $\mathcal{S}'$. We construct $\pi$ 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 $\pi$ is BIC. For any $m, m' \in \mathcal{M}$ and $\theta \in \mathit{\Theta}$,

$$
\begin{aligned}
&\sum_{\omega,s} \Pr[\Omega = \omega] \cdot \Pr[S = s|\Omega = \omega] \cdot \big(u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)\big) \cdot \Pr[\pi(s) = m] \\
=& \sum_{\omega,s,s'} \Pr[\Omega = \omega, S = s] \cdot \Pr[S' = s'|S = s] \cdot \Pr[\pi'(s') = m] \cdot \big(u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)\big) \\
\geq& \sum_{\omega,s,s'} \Pr[\Omega = \omega, S = s, S' = s'] \cdot \Pr[\pi'(s') = m] \cdot \big(u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)\big) \\
&- 2 \cdot \sum_{\omega,s,s'} |\Pr[\Omega = \omega, S = s] \cdot \Pr[S' = s'|S = s] - \Pr[\Omega = \omega, S = s, S' = s']| \\
=& \sum_{\omega,s'} \Pr[\Omega = \omega, S' = s'] \cdot \Pr[\pi'(s') = m] \cdot \big(u(\theta, m(\theta), \omega) - u(\theta, m'(\theta), \omega)\big) \\
&- 2 \cdot \sum_{s} \Pr[S = s] \cdot \sum_{s',\omega} \big|\Pr[S' = s', \Omega = \omega|S = s] - \Pr[S' = s'|S = s] \cdot \Pr[\Omega = \omega|S = s]\big| \\
\geq& \ \delta - 2 \cdot \frac{\delta}{2} \\
=& \ 0
\end{aligned}
$$

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

For a given signal $s \in \mathcal{X}$ and a menu $m \in \mathcal{M}$, we solve the following LP to check if $m$ is $\delta$-signal-exlporable given signal $s$:

**maximize** $x_{m,s}$

**subject to:**

$$\sum_{\omega \in \mathit{\Omega}, s' \in \mathcal{X}} \Pr[\omega] \cdot \Pr[s'|\omega] \cdot \big(u(\theta, m'(\theta), \omega) - u(\theta, m''(\theta), \omega) + \delta\big) \cdot x_{m',s'} \geq 0 \ \forall m', m'' \in \mathcal{M}, \theta \in \mathit{\Theta}$$

$$\sum_{m' \in \mathcal{M}} x_{m',s'} = 1, \qquad \forall s' \in \mathcal{X}$$

$$x_{m',s'} \geq 0, \qquad \forall s' \in \mathcal{X}, m' \in \mathcal{M}$$

As the above LP constraints characterize all BIC recommendation schemes, we have the following claim.

**Claim 4.3.** *For a given signal $s \in \mathcal{X}$, a menu $m \in \mathcal{M}$ is $\delta$-signal-explorable if and only if the above LP has a positive solution. When the LP has a positive solution, define $\pi^{m,s}$ as $\Pr[\pi^{m,s}(s') = m'] = x_{m',s'}, \forall m' \in \mathcal{M}, s' \in \mathcal{X}$. Then $\pi^{m,s}$ is a single-round $\delta$-BIC recommendation policy such that* $\Pr[\pi^{m,s}(s) = m] > 0$

**Definition 4.8** (Max-support policy). *Given a signal structure $\mathcal{S}$, a recommendation policy $\pi$ is called the $\delta$-max-support policy if $\forall s \in \mathcal{X}$ and $\delta$-signal-explorable menu $m \in \mathcal{M}$ given $s$,* $\Pr[\pi(s) = m] > 0$.

By Claim 4.3, we have the following claim.

**Claim 4.4.** *The following $\pi^{max}$ is a $\delta$-BIC and $\delta$-max-support policy. Here $\mathcal{M}'$ is the set of menus with positive solutions in the LP mentioned in Claim 4.3.*

$$\pi^{max} = \frac{1}{|\mathcal{X}|} \sum_{s \in \mathcal{X}} \frac{1}{|\mathcal{M}'|} \sum_{m \in \mathcal{M}'} \pi^{m,s}.$$

### 4.3 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 the distribution of the samples. $D_m$ is random variable depending on the state $\Omega$. For a fixed state $\omega$, we use $D_m(\omega)$ to denote the distribution of the samples of menu $m$.

**Lemma 4.9.** *For any $\alpha > 0$, we can compute $\Delta_m$ which is a function of $B_m = O\left(\ln\left(\frac{1}{\gamma}\right)\right)$ samples of menu $m$ such that for any state $\omega$,*

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

*Proof.* Let $U$ be the union of the support of $D_m(\omega)$ for all $\omega \in \mathit{\Omega}$. 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 $\delta_m$ be small enough such that for all $\omega, \omega'$ 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 $\Delta_m$ as following: Take $B_m = \frac{2}{\delta_m^2} \ln\left(\frac{2|U|}{\gamma}\right)$ samples and set $\hat{q}(u)$ as the empirical frequency of seeing $u$. And set $\Delta_m$ 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 $\omega$ exists, $\Delta_m$ will be unique. If no $\omega$ satisfies this, just pick $\Delta_m$ to be an arbitrary $D_m(\omega)$.

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

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

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 $\Delta_m = D_m(\omega)$. □

### 4.4 MaxExplore

In this subsection, we are going to assume $L \geq \max_{m,s} \frac{B_m}{\Pr[\pi^{max}(s) = m]}$.

**Algorithm 6** Subroutine MaxExplore

```
**Input:** signal $S$, signal structure $\mathcal{S}$.
**Output:** a list of menus $\mu$
Compute $\pi^{max}$ as stated in Claim 4.4.
Initialize $Res = L$.
**for** each menu $m \in \mathcal{M}$ **do**
    Add $\lfloor L \cdot \Pr[\pi^{max}(S) = m] \rfloor$ copies of menu $m$ into list $\mu$.
    $Res \leftarrow Res - \lfloor L \cdot \Pr[\pi^{max}(S) = m] \rfloor$.
    $p^{Res}(a) \leftarrow L \cdot \Pr[\pi^{max}(S) = m] - \lfloor L \cdot \Pr[\pi^{max}(S) = m] \rfloor$
**end for**
$p^{Res}(m) \leftarrow p^{Res}(m)/Res$, $\forall m \in \mathcal{M}$.
Sample $Res$ many actions from distribution according to $p^{Res}$ independently and add these actions into $\mu$.
Randomly permute the actions in $\mu$.
**return** $\mu$.
```

**Claim 4.5.** *Given signal $S$, MaxExplore is $\delta$-BIC and explores each $\delta$-signal-explorable menu $m$ at least $B_m$ times.*

## 4.5 Main Scheme

In this subsection, we show our main scheme which explores all the eventually-explorable actions and then recommends the agents the best menu given all history. We pick $L$ to be at least $\max_{m,s:\Pr[\pi(s)=m]>0} \frac{B_m}{\Pr[\pi(s)=m]}$ for all $\pi$ that might be chosen as $\pi^{max}$ by Algorithm 7.

**Algorithm 7** Main procedure for private types

```
Initial signal $S_1 = \mathcal{S}_l = \perp$.
Initial phase count $l = 1$.
for $t = 1$ to $T$ do
  if $t \equiv 1 \pmod{L}$ then
    Start a new phase:
    For each explored menu $m$ in the previous phase, use $B_m$ samples to compute $\Delta_m$ stated in Lemma 4.9 with $\gamma = \min\left(\frac{\delta^2}{16|\mathcal{M}|\log(|\mathit{\Omega}|)}, \left(\frac{\delta^2}{32|M|}\right)^2, \frac{1}{T|\mathcal{M}|}\right)$.
    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 $|\mathit{\Omega}|$.
    Set $S_l$ to be the collection of $\Delta_m$'s for all explored menu $m$.
    if $l \leq |\mathcal{M}|$ then
      Use the current $S_l$ and $\mathcal{S}_l$ to compute a list of $L$ actions $\mu \leftarrow \text{MaxExplore}(S_l, \mathcal{S}_l)$.
    else
      Set menu list $\mu$ to be $L$ copies of the best explored menu according to all history.
    end if
  end if
  Suggest menu $\mu[(t-1) \mod L + 1]$ to the agent.
end for
```

**Claim 4.6.** *Algorithm 7 is $\delta$-BIC.*

**Lemma 4.10.** *For any $l > 0$, assume Algorithm 7 has at least $\min(l, |\mathcal{M}|)$ phases. For a given state $\omega$, if a menu $m$ can be explored by a BIC recommendation policy $\pi$ at round $l$ (i.e. $\Pr[\pi^l = m] > 0$), then such menu is guaranteed to be explored $B_m$ times by Algorithm 7 by the end of phase $\min(l, |\mathcal{M}|)$.*

*Proof.* The proof is similar to Lemma 2.9. We prove by induction on $l$ for $l \leq |\mathcal{M}|$.

Let $S$ be the signal of Algorithm 7 in phase $l$. Let $S'$ be the history of $\pi$ in the first $l-1$ rounds. More precisely, $S' = R, H_1, ..., H_{l-1}$. Here $R$ is the internal randomness of scheme $\pi$ and $H_t = (M_t, A_t, u(\Theta_t, M_t(\Theta_t), \Omega))$ is the menu and the action-reward pair in round $t$ of scheme $\pi$.

Let $\mathcal{M}'$ to be the set of menus explored in the first $l-1$ phases of Algorithm 7. By the induction hypothesis, we have $\forall t \in [l-1]$, $M_t \subseteq \mathcal{M}'$.

First of all, we have

$$I(S';\Omega|S) = I(R, H_1, ..., H_{l-1};\Omega|S) = I(R;\Omega|S) + I(H_1, ..., H_{l-1};\Omega|S, R) = I(H_1, ..., H_{l-1};\Omega|S, R).$$

By the chain rule of mutual information, we have

$$I(H_1, ..., H_{l-1};\Omega|S, R) = I(H_1;\Omega|S, R) + I(H_2;\Omega|S, R, H_1) + \cdots + I(H_{l-1};\Omega|S, R, H_1, ..., H_{l-2}).$$

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

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

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 $(A_t, u(\Theta_t, M_t(\Theta_t), \Omega))$ is independent with $\Omega$ given $D_{M_t}$.

Then we have

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

The last step comes from the fact that $I(D_m; (S \backslash \Delta_m), R, H_1, ..., H_{t-1} | \Omega, \Delta_m, M_t = m) = 0$. By Lemma 4.9, we know that $\Pr[D_m \neq \Delta_m | M_t = m] \leq \gamma$. By Fano's inequality, we have

$$H(D_m | \Delta_m, M_t = m) \leq H(\gamma) + \gamma \log(|\Omega| - 1) \leq 2\sqrt{\gamma} + \gamma \log(|\Omega| - 1) \leq \frac{\delta^2}{16|\mathcal{M}|} + \frac{\delta^2}{16|\mathcal{M}|} = \frac{\delta^2}{8|\mathcal{M}|}.$$

Therefore we have

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

Then we get

$$I(S'; \Omega | S) \leq \delta^2/8.$$

By Lemma 4.7, we know that $EX_{s'}[\mathcal{S}'] \subseteq EX_s^\delta[\mathcal{S}]$. By Claim 4.5, we know that phase $l$ will explore menu $m$ at least $B_m$ times.

When $l > |\mathcal{M}|$, the proof follows from the same argument as the last paragraph of the proof of Lemma 2.9. □

**Corollary 4.11** (Restatement of Theorem 4.4)**.** *For any $\delta > 0$, we have a $\delta$-BIC recommendation policy of $T$ rounds with expected total reward at least $(T - C \cdot \log(T)) \cdot OPT$ for some constant $C$ which does not depend on $T$.*

*Proof.* First of all, by Claim 4.6, Algorithm 7 is $\delta$-BIC.

By Lemma 4.10, for each state $\omega$, Algorithm 7 explores all the eventually explorable menus (i.e. $\mathcal{M}_\omega^{exp}$) for by the end of $|\mathcal{M}|$ phases.

After that, if the algorithm just pick the best explored menu according to all $\Delta_m$'s, by Lemma 4.9 and $\gamma \leq \frac{1}{T|\mathcal{M}|}$, for a fixed state $\omega$, we know that with probability $1 - 1/T$, the algorithm chooses menu $\arg\max_{m\in\mathcal{M}^{exp}_{\omega}} \sum_{\theta\in\Theta} \Pr[\theta]\cdot u(\theta, m(\theta), \omega)$. And since Algorithm 7 chooses the best menu according to all history, it should at least get $(1-1/T)\cdot \max_{m\in\mathcal{M}^{exp}_{\omega}} \sum_{\theta\in\Theta} \Pr[\theta]\cdot u(\theta, m(\theta), \omega)$ in expectation.

We know that the expected number of rounds of the first $|\mathcal{M}|$ is $|\mathcal{M}|\cdot L = O(\ln(T))$. Therefore, Algorithm 7 has expected total reward at least $T\cdot OPT - T\cdot(1/T) - O(\ln(T)) = T\cdot OPT - O(\ln(T))$. □

# 5 Do More Types Help Exploration?

In this section, we discuss whether having more types helps exploration. In particular, we compare our original instance with a less diverse instance. The less diverse instance has the same utility function $u$, state space $\Omega$ and probability distribution over states. The less diverse instance has a smaller set of types $\Theta' \subsetneq \Theta$ and the probability of each type $\theta \in \Theta'$ is arbitrary.

In Definition 2.1 in Section 2, we define $\mathcal{A}^{exp}_{\omega,\theta}$ as the set of eventually-explorable actions for a given state $\omega$ and type $\theta$. We use $A'^{exp}_{\omega,\theta}$ to denote the set of eventually-explorable actions for the less diverse instance.

**Claim 5.1.** *For any $\omega \in \Omega, \theta \in \Theta'$, $A'^{exp}_{\omega,\theta} \subseteq \mathcal{A}^{exp}_{\omega,\theta}$.*

*Proof.* The proof simply follows from Lemma 2.9. It's easy to check that the lemma still works even if $\pi$ is a BIC scheme for the less diverse instance. This implies that for a given state $\omega$, Algorithm 2 for the original instance explores all actions in $A'^{exp}_{\omega,\theta}$ for all $\theta \in \Theta'$. And therefore $A'^{exp}_{\omega,\theta} \subseteq \mathcal{A}^{exp}_{\omega,\theta}$. □

From Section 2 and 3, we know that when types are public or types are private and communication is allowed, we have a BIC scheme to explore all eventually explorable actions. Therefore in these two cases, more types do help exploration.

Finally we are going to look at the case when types are private and communication is not allowed. First of all, more types can help in some situations. For example, if types have disjoint set of actions, then this case is the same as the case when type are private and communication is allowed. And therefore more types help exploration in this situation.

On the other hand, we show in the following example that more types can hurt exploration when types are private and communication allowed.

**Example 5.1.** *$\Omega = \{0,1\}$, $\Theta = \{0,1\}$ and $\mathcal{A} = \{0,1\}$. $\Pr[\omega = 0] = \Pr[\omega = 1] = 1/2$ and $\Pr[\theta = 0] = \Pr[\theta = 1] = 1/2$. We define $u(\theta, a, \omega)$ 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:** $u(\theta, a, \omega)$ when $\omega = 0$ or 1.

In this example, it is easy to check that action 0 is preferred by both types without any information about the state. On the other hand, samples from action 0 does not convey any information about the state. Therefore, the set of eventually-explorable actions for both types and both states is $\{0\}$. Now consider a less diverse instance in which only type 0 appears. After one agent in that type chooses action 0, the state is reviewed to the principal. When the state $\omega = 0$, action 1 can be recommended to future agents.