Decentralized Exploration in Multi-Armed Bandits

Raphaël Féraud, Réda Alami, Romain Laroche
raphael.feraud@orange.com, reda.alami@orange.com, Romain.Laroche@microsoft.com

June 2019

Outline

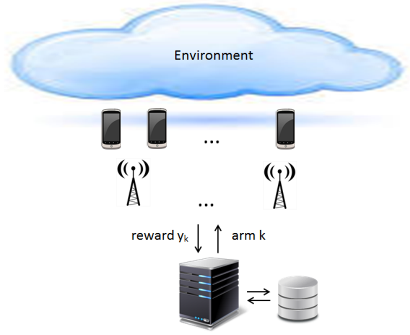
Sequential A/B testing use cases

Most of digital applications perform sequential A/B testing in order to optimize their audience. For instance, Orange web portal performs marketing optimization for promoting services:
If I would like to promote Orange TV which banner is the best ?
Should I push on Games of Thrones or on Sports?

(Centralized) Exploration Problem

Definition 1 (ϵ-optimal arm)

An arm $k \in \mathcal{K}$ is said to be ϵ-optimal, if $\mu_k \geqslant \mu_{k*} - \epsilon$, where $k^* = \arg\max_{k \in \mathcal{K}} \mu_k, \epsilon \in (0, 1]$, and μ_k is the mean reward of arm k.



Centralized approach:

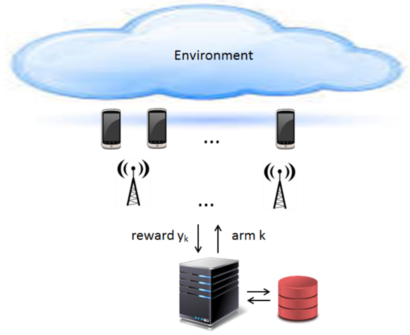
The click stream of users is gathered and processed by a Best Arm Identification algorithm to choose with high probability an ϵ-optimal arm.

Do we really need to gather billions of logs containing private user's information for handling sequential A/B testing use cases ?

(Centralized) Exploration Problem

Definition 1 (ϵ-optimal arm)

An arm $k \in \mathcal{K}$ is said to be ϵ-optimal, if $\mu_k \geqslant \mu_{k*} - \epsilon$, where $k^* = \arg\max_{k \in \mathcal{K}} \mu_k, \epsilon \in (0, 1]$, and μ_k is the mean reward of arm k.



Centralized approach:

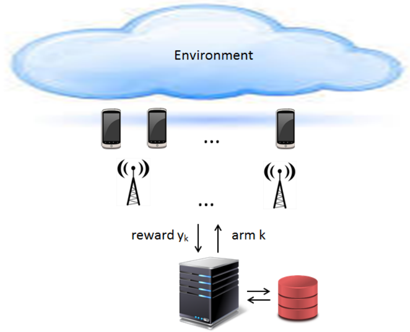
The click stream of users is gathered and processed by a Best Arm Identification algorithm to choose with high probability an ϵ-optimal arm.

Do we really need to gather billions of logs containing private user's information for handling sequential A/B testing use cases ?

(Centralized) Exploration Problem

Definition 1 (ϵ-optimal arm)

An arm $k \in \mathcal{K}$ is said to be ϵ-optimal, if $\mu_k \geqslant \mu_{k*} - \epsilon$, where $k^* = \arg\max_{k \in \mathcal{K}} \mu_k, \epsilon \in (0, 1]$, and μ_k is the mean reward of arm k.



Centralized approach:

The click stream of users is gathered and processed by a Best Arm Identification algorithm to choose with high probability an ϵ-optimal arm.

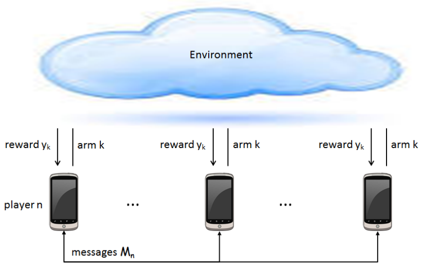
Do we really need to gather billions of logs containing private user's information for handling sequential A/B testing use cases ?

Outline

Problem setting

Definition 2 (message)

A message is a random variable, that is sent by player *n* to other players.



1. When the event *"player n is active"* occurs, player *n* reads the messages received from other players
2. Player *n* chooses an arm to play.
3. The reward of the played arm is revealed to player *n*.
4. Player *n* may update its set of arms and/or send a message to the other players.

Goal

Designing an algorithm that samples effectively to find an ϵ-optimal arm for each player, while ensuring privacy and minimizing the number of messages.

Privacy guarantee

We define the privacy level as the information about the preferred arms of a player, that an adversary could infer by intercepting the messages of this player.

Definition 3 ((ϵ, η)-private).

The decentralized algorithm $\mathcal{A}$ is (ϵ, η)-private for finding an ϵ-approximation of the best arm, if for any player n, $\nexists \eta_1$, $0 < \eta_1 < \eta < 1$ such that an adversary, that knows $\mathcal{M}_n$, the set of messages of player n, and the algorithm $\mathcal{A}$, can infer what arm is an ϵ-approximation of the best arm for player n with a probability at least $1 - \eta_1$:

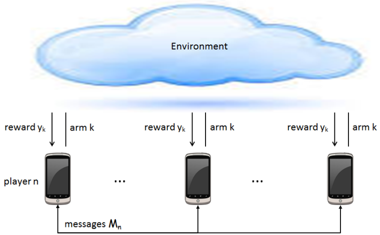
$$\forall n \in \mathcal{N}, \quad \forall l^n \in \{1, ..., L\}, \quad \mathbb{P}\left(\mathcal{K}^n(l^n) \subseteq \mathcal{K}_\epsilon | \mathcal{M}_n, \mathcal{A}\right) \geqslant 1 - \eta_1,$$

where $\mathcal{K}_\epsilon$ is the set of ϵ-optimal arms, and $\mathcal{K}^n$ is the set of arms of player n, and l^n is the number of times where $\mathcal{K}^n$ has been updated, and $L \leqslant K$.

$1 - \eta$ is the confidence level associated to the decision of the adversary: the higher η, the higher the privacy protection.

Outline

Decentralized Elimination: the principle

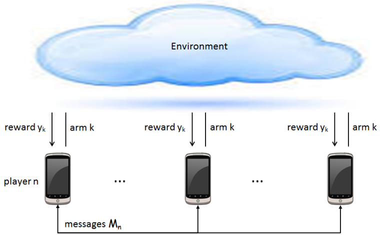


An Arm Selection Subroutine is run on each player. The players exchange the indexes of arms that they eliminate with a high probability of failure η. The high probability of failure insures privacy of messages. When enough players vote for the elimination of an arm, it is eliminated for all players.

Why does it work ?

When $M \leqslant N$ players independently eliminate an arm with a probability of failure η, then the probability of failure of the group of M voting players is $\delta = \eta^M$.

Decentralized Elimination: the principle



An Arm Selection Subroutine is run on each player. The players exchange the indexes of arms that they eliminate with a high probability of failure η. The high probability of failure insures privacy of messages. When enough players vote for the elimination of an arm, it is eliminated for all players.

Why does it work ?

When $M \leqslant N$ players independently eliminate an arm with a probability of failure η, then the probability of failure of the group of M voting players is $\delta = \eta^M$.

Decentralized Elimination: a generic algorithm

Definition 4 (Arm Selection Subroutine)

An ArmSelection subroutine takes as parameters an approximation factor ϵ, a failure probability η, and a set of remaining arm $\mathcal{K}^n(l^n)$, where l^n is the number of times $\mathcal{K}^n$ has been updated. It samples a remaining arm in $\mathcal{K}^n(l^n)$ and returns the set of eliminated arms $\overline{\mathcal{K}^n}(l^n)$.
An ArmSelection subroutine satisfies Properties 1 and 2.

Property 1 (remaining ϵ-optimal arm)

$$\forall l^n \in \{1, ..., L\}, \mathcal{K}^n\left(l^n\right) \subset \mathcal{K}^n\left(l^n - 1\right),$$
$$\mathbb{P}\left(\{\mathcal{K}^n(l^n) \cap \mathcal{K}_\epsilon = \emptyset\} | \mathcal{H}_{t^n}, \mathcal{K}^n\left(l^n - 1\right) \cap \mathcal{K}_\epsilon \neq \emptyset\right) \leqslant \eta \times f(l^n),$$
where $0 \leqslant f(l^n) \leqslant 1$ and $\sum_{l^n} f(l^n) = 1,$ and $\mathcal{H}_{t^n}$ is the interaction history.

Property 2 (finite sample complexity)

$$\exists t^n \geqslant 1, \forall \eta \in (0,1), \forall \epsilon \in (0,1], \mathbb{P}\left(\{\mathcal{K}^n(L) \subset \mathcal{K}_\epsilon\} | \mathcal{H}_{t^n}\right) \geqslant 1 - \eta.$$

Decentralized Elimination: a generic algorithm

Definition 4 (Arm Selection Subroutine)

An ArmSelection subroutine takes as parameters an approximation factor ϵ, a failure probability η, and a set of remaining arm $\mathcal{K}^n(l^n)$, where l^n is the number of times $\mathcal{K}^n$ has been updated. It samples a remaining arm in $\mathcal{K}^n(l^n)$ and returns the set of eliminated arms $\overline{\mathcal{K}^n}(l^n)$.
An ArmSelection subroutine satisfies Properties 1 and 2.

Property 1 (remaining ϵ-optimal arm)

$$\forall l^n \in \{1, ..., L\}, \mathcal{K}^n\left(l^n\right) \subset \mathcal{K}^n\left(l^n - 1\right),$$
$$\mathbb{P}\left(\{\mathcal{K}^n(l^n) \cap \mathcal{K}_\epsilon = \varnothing\} | \mathcal{H}_{t^n}, \mathcal{K}^n\left(l^n - 1\right) \cap \mathcal{K}_\epsilon \neq \varnothing\right) \leqslant \eta \times f(l^n),$$

where $0 \leqslant f(l^n) \leqslant 1$ and $\sum_{l^n} f(l^n) = 1,$ and $\mathcal{H}_{t^n}$ is the interaction history.

Property 2 (finite sample complexity)

$$\exists t^n \geqslant 1, \forall \eta \in (0,1), \forall \epsilon \in (0,1], \mathbb{P}\left(\{\mathcal{K}^n(L) \subset \mathcal{K}_\epsilon\} | \mathcal{H}_{t^n}\right) \geqslant 1 - \eta.$$

Decentralized Elimination: a generic algorithm

Definition 4 (Arm Selection Subroutine)

An ArmSelection subroutine takes as parameters an approximation factor ϵ, a failure probability η, and a set of remaining arm $\mathcal{K}^n(l^n)$, where l^n is the number of times $\mathcal{K}^n$ has been updated. It samples a remaining arm in $\mathcal{K}^n(l^n)$ and returns the set of eliminated arms $\overline{\mathcal{K}^n}(l^n)$.
An ArmSelection subroutine satisfies Properties 1 and 2.

Property 1 (remaining ϵ-optimal arm)

$$\forall l^n \in \{1, ..., L\}, \mathcal{K}^n\left(l^n\right) \subset \mathcal{K}^n\left(l^n - 1\right),$$
$$\mathbb{P}\left(\{\mathcal{K}^n(l^n) \cap \mathcal{K}_\epsilon = \emptyset\} | \mathcal{H}_{t^n}, \mathcal{K}^n\left(l^n - 1\right) \cap \mathcal{K}_\epsilon \neq \emptyset\right) \leqslant \eta \times f(l^n),$$
$$\text{where } 0 \leqslant f(l^n) \leqslant 1 \text{ and } \sum_{l^n} f(l^n) = 1, \text{ and } \mathcal{H}_{t^n} \text{ is the interaction history.}$$

Property 2 (finite sample complexity)

$$\exists t^n \geqslant 1, \forall \eta \in (0,1), \forall \epsilon \in (0,1], \mathbb{P}\left(\{\mathcal{K}^n(L) \subset \mathcal{K}_\epsilon\} | \mathcal{H}_{t^n}\right) \geqslant 1 - \eta.$$

Analysis of Decentralized Elimination: privacy

Theorem 1

Using any ArmSelection *subroutine,* DECENTRALIZED ELIMINATION *is an* (ϵ, η)*-private algorithm, that finds an* ϵ*-optimal arm with a failure probability* $\delta \leqslant \eta^{\lfloor \frac{\log \delta}{\log \eta} \rfloor}$ *and that exchanges at most* $\lfloor \frac{\log \delta}{\log \eta} \rfloor K - 1$ *messages.*

Comment 1

Theorem 1 provides the number of players needed to find an ϵ-optimal arm with high probability while insuring privacy: $M = \lfloor \frac{\log \delta}{\log \eta} \rfloor$.

Comment 2

The communication cost depends only on the problem parameters: the privacy constraint η, the probability of failure δ, the number of actions, and notably not on the number of samples.

Analysis of Decentralized Elimination: privacy

Theorem 1

Using any ArmSelection *subroutine,* DECENTRALIZED ELIMINATION *is an* (ϵ, η)*-private algorithm, that finds an* ϵ*-optimal arm with a failure probability* $\delta \leqslant \eta^{\lfloor \frac{\log \delta}{\log \eta} \rfloor}$ *and that exchanges at most* $\lfloor \frac{\log \delta}{\log \eta} \rfloor K - 1$ *messages.*

Comment 1

Theorem 1 provides the number of players needed to find an ϵ-optimal arm with high probability while insuring privacy: $M = \lfloor \frac{\log \delta}{\log \eta} \rfloor$.

Comment 2

The communication cost depends only on the problem parameters: the privacy constraint η, the probability of failure δ, the number of actions, and notably not on the number of samples.

Analysis of Decentralized Elimination: privacy

Theorem 1

Using any ArmSelection *subroutine,* DECENTRALIZED ELIMINATION *is an* (ϵ, η)*-private algorithm, that finds an* ϵ*-optimal arm with a failure probability* $\delta \leqslant \eta^{\lfloor \frac{\log \delta}{\log \eta} \rfloor}$ *and that exchanges at most* $\lfloor \frac{\log \delta}{\log \eta} \rfloor K - 1$ *messages.*

Comment 1

Theorem 1 provides the number of players needed to find an ϵ-optimal arm with high probability while insuring privacy: $M = \lfloor \frac{\log \delta}{\log \eta} \rfloor$.

Comment 2

The communication cost depends only on the problem parameters: the privacy constraint η, the probability of failure δ, the number of actions, and notably not on the number of samples.

Analysis of Decentralized Elimination: sample complexity

Let $T_{P_\mathbf{y}}$ be the number of samples in $P_\mathbf{y}$ needed by the ArmSelection subroutine to find an ϵ-optimal arm with high probability.

Theorem 2

Using any ArmSelection *subroutine, with a probability of failure a little bit higher than* $\eta^{\lfloor \frac{\log \delta}{\log \eta} \rfloor}$ DECENTRALIZED ELIMINATION *stops after:*

$$\mathcal{O}\left(\frac{1}{p_*}\left(T_{P_\mathbf{y}} + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right)\right) \text{ samples in } P_{x,\mathbf{y}},$$

where $p^* = \min_{n \in \mathcal{N}_M} P_x(x = n)$ *be the probability of the least frequent voting player.*

Theorem 2 states that the penalty coming from the privacy and the communication cost constraints is mainly depending on the probability of the least frequent voting player.

Analysis of Decentralized Elimination: illustration

We consider the case where the distribution of players is uniform, and where a optimal arm selection subroutine is used. With a failure probability at most $\delta = \eta^N$ the number of sample in $P_{x,\mathbf{y}}$ needed by DECENTRALIZED ELIMINATION to find an ϵ-optimal arm is:

$$\mathcal{O}\left(\frac{K}{\epsilon^2}\log\frac{1}{\delta} + N\sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right) \text{ samples in } P_{x,\mathbf{y}}.$$

In comparison to a optimal centralized algorithm, which communicates all the messages and does not provide privacy protection guarantee, in the case of uniform distribution of players, the sample complexity of DECENTRALIZED ELIMINATION suffers from a penalty that is linear with respect to the number of players.

Outline

Experiments: setting

- Problem 1: Uniform distribution of players. 10 arms. The optimal arm has a mean reward $\mu_1 = 0.7$, the second one $\mu_2 = 0.5$, the third one $\mu_3 = 0.3$, and the others have a mean reward of 0.1.
- Problem 2: 50% of players generates 80% of events. Same arms.

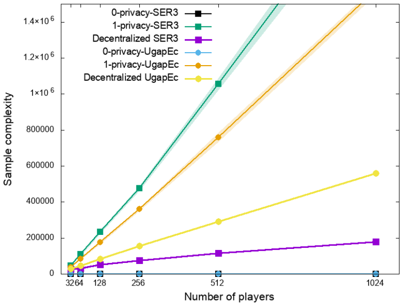
Baselines

- 1-privacy: an $(\epsilon, 1)$-private algorithm that does not share any information between the players.
- 0-privacy: an $(\epsilon, 0)$-private algorithm that shares all the information between players.

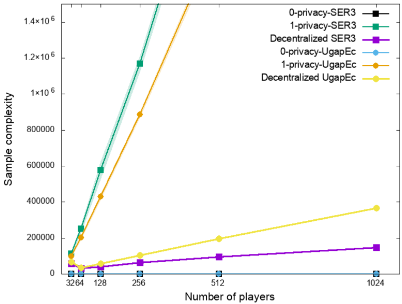
Arm Elimination subroutines

- SER3 (Successive Elimination with Randomized Round Robin) is based on uniform sampling and successive eliminations of suboptimal arms.
- UGapEc uses adaptive sampling and a stopping rule to output the best arm.

Experiments



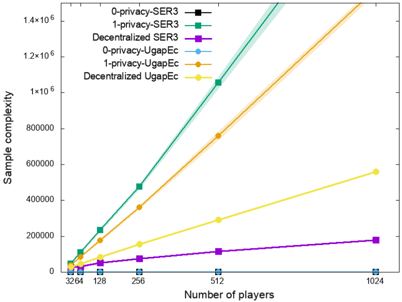
Uniform distribution of players - Sample Complexity



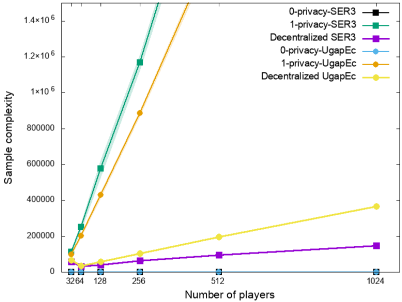
50% of players generates 80% of events - Sample Complexity

The performances of 1-PRIVACY baselines are horrendous in both problems. Worse, when the distribution of players moves away from the uniformity, the performances of 1-PRIVACY baselines decrease.

Experiments



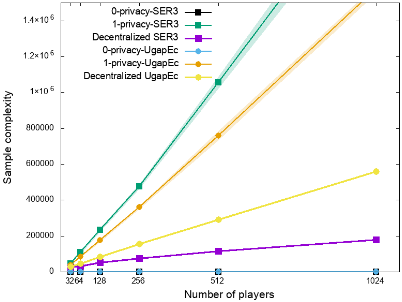
Uniform distribution of players - Sample Complexity



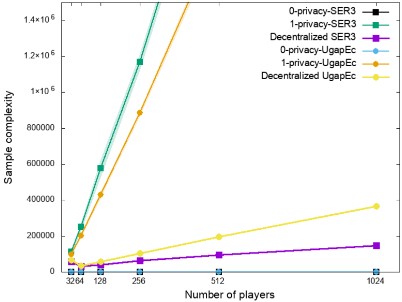
50% of players generates 80% of events - Sample Complexity

1-privacy-UGAPEC outperforms 1-privacy-SER3, while DECENTRALIZED SER3 outperforms DECENTRALIZED UGAPEC: Successive Elimination algorithms are better suited for DECENTRALIZED ELIMINATION than Explore Then Commit algorithms.

Experiments



Uniform distribution of players - Sample Complexity



50% of players generates 80% of events - Sample Complexity

The linear dependency of the sample complexity of DECENTRALIZED ELIMINATION with respect to the number of players is due to the fact that in the considered problems, the probability of the least frequent voting player p^*decreases in $O(1/N)$.

Outline

1. Context and Motivation
2. Decentralized Exploration Problem
3. Decentralized Elimination Algorithm
4. Experiments
5. Conclusion

Centralized versus Decentralized Exploration

Benefits of decentralized exploration:

Privacy, by using an (ϵ, η)-private algorithm,
Dramatic reduction of the communication cost, which is a strict requirement for Internet Of Things,
Increasing responsiveness of mobile phone applications, by vanishing the interactions with a central server,
Scalability, thanks to parallel processing.

Cost of decentralized exploration:

higher sample complexity, due to the privacy and the communication cost constraints.

The decentralized exploration allows a good balance between conflicting interests: the service provider performs sequential A/B testing, while saving resources and protecting privacy of users.

Centralized versus Decentralized Exploration

Benefits of decentralized exploration:

Privacy, by using an (ϵ, η)-private algorithm,
Dramatic reduction of the communication cost, which is a strict requirement for Internet Of Things,
Increasing responsiveness of mobile phone applications, by vanishing the interactions with a central server,
Scalability, thanks to parallel processing.

Cost of decentralized exploration:

higher sample complexity, due to the privacy and the communication cost constraints.

The decentralized exploration allows a good balance between conflicting interests: the service provider performs sequential A/B testing, while saving resources and protecting privacy of users.

Centralized versus Decentralized Exploration

Benefits of decentralized exploration:

Privacy, by using an (ϵ, η)-private algorithm,
Dramatic reduction of the communication cost, which is a strict requirement for Internet Of Things,
Increasing responsiveness of mobile phone applications, by vanishing the interactions with a central server,
Scalability, thanks to parallel processing.

Cost of decentralized exploration:

higher sample complexity, due to the privacy and the communication cost constraints.

The decentralized exploration allows a good balance between conflicting interests: the service provider performs sequential A/B testing, while saving resources and protecting privacy of users.

Conclusion

Main contributions:

- the decentralized exploration problem, where players collaborate to find an ϵ-optimal arm,
- a privacy definition for decentralized exploration, based on the quality of information an adversary could infer from the messages of each player,
- a generic algorithm for decentralized exploration, Decentralized Elimination, which ensures privacy and low communication cost, while controlling the sample complexity,
- experiments which suggest that the successive elimination algorithms are better suited for Decentralized Elimination.

Bonus

- Thanks to the generality of the approach, we have extended it to the case of non-stationary bandits (in the paper).

Conclusion

Main contributions:

- the decentralized exploration problem, where players collaborate to find an ϵ-optimal arm,
- a privacy definition for decentralized exploration, based on the quality of information an adversary could infer from the messages of each player,
- a generic algorithm for decentralized exploration, Decentralized Elimination, which ensures privacy and low communication cost, while controlling the sample complexity,
- experiments which suggest that the successive elimination algorithms are better suited for Decentralized Elimination.

Bonus

- Thanks to the generality of the approach, we have extended it to the case of non-stationary bandits (in the paper).