Bayesian Design Principles for Frequentist Sequential Learning

Yunbei Xu and Assaf Zeevi

Columbia University

Sequential learning with partial feedback

- An agent sequentially makes decisions to maximize cumulative reward.

Sequential learning with partial feedback

- An agent sequentially makes decisions to maximize cumulative reward.
- Challenge: only observes feedback of her chosen decision, but does not observe feedback of other unchosen decisions.

Sequential learning with partial feedback

- An agent sequentially makes decisions to maximize cumulative reward.
- Challenge: only observes feedback of her chosen decision, but does not observe feedback of other unchosen decisions.
- Trade-off between *exploration* and *exploitation*:
  - needs to try different decisions to learn the environment.
  - wants to focus on good decisions and avoid bad decisions to maximize incurred reward.

Sequential learning with partial feedback

- An agent sequentially makes decisions to maximize cumulative reward.
- Challenge: only observes feedback of her chosen decision, but does not observe feedback of other unchosen decisions.
- Trade-off between *exploration* and *exploitation*:
  - needs to try different decisions to learn the environment.
  - wants to focus on good decisions and avoid bad decisions to maximize incurred reward.
- We develop a general theory encompassing bandit problems, reinforcement learning, and beyond.

Frequentist vs Bayesian

Frequentist:

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).

Bayesian:

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.

Bayesian:

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.
- **Cons:** heavily relies on case-by-case design (e.g., reward estimators) and special structures.

Bayesian:

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.
- **Cons:** heavily relies on case-by-case design (e.g., reward estimators) and special structures.

Bayesian:

- **Example:** Thompson Sampling (TS) with a fixed, pre-specified prior.

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.
- **Cons:** heavily relies on case-by-case design (e.g., reward estimators) and special structures.

Bayesian:

- **Example:** Thompson Sampling (TS) with a fixed, pre-specified prior.
- **Pros:** offers optimality and generality if prior is known.

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.
- **Cons:** heavily relies on case-by-case design (e.g., reward estimators) and special structures.

Bayesian:

- **Example:** Thompson Sampling (TS) with a fixed, pre-specified prior.
- **Pros:** offers optimality and generality if prior is known.
- **Cons:** knowledge of prior not accessible in complex settings; maintaining posterior computationally expensive.

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.
- **Cons:** heavily relies on case-by-case design (e.g., reward estimators) and special structures.

Bayesian:

- **Example:** Thompson Sampling (TS) with a fixed, pre-specified prior.
- **Pros:** offers optimality and generality if prior is known.
- **Cons:** knowledge of prior not accessible in complex settings; maintaining posterior computationally expensive.

Frequentist vs Bayesian

Frequentist:

- **Example:** Upper Confidence bound (UCB).
- **Pros:** does not require prior knowledge of environment; computationally efficient.
- **Cons:** heavily relies on case-by-case design (e.g., reward estimators) and special structures.

Bayesian:

- **Example:** Thompson Sampling (TS) with a fixed, pre-specified prior.
- **Pros:** offers optimality and generality if prior is known.
- **Cons:** knowledge of prior not accessible in complex settings; maintaining posterior computationally expensive.

Frequentist approach requires less information, but is more bottom-up;
Bayesian approach is more top-down, but requires stronger assumptions.

Main research question

Can we develop principled Bayesian-type algorithms, that are prior-free, computationally efficient, and work well in both stochastic and adversarial/non-stationary environments?

Contributions: approach

A general theory that creates algorithmic beliefs to simulate worst-case environment and uses posteriors to make decisions

$\Rightarrow$ Synergizing Frequentist and Bayesian approaches

Contributions: approach

A general theory that creates algorithmic beliefs to simulate worst-case environment and uses posteriors to make decisions

$\Rightarrow$ Synergizing Frequentist and Bayesian approaches

- Creates "algorithmic beliefs" on the fly, rather than using pre-specified, fixed prior. Automatically adapts to adversarial environments.
  - The first approach that allows Bayesian-type algorithms to operate without prior assumptions and be applicable in adversarial settings.

Contributions: approach

A general theory that creates algorithmic beliefs to simulate worst-case environment and uses posteriors to make decisions
$\Rightarrow$ Synergizing Frequentist and Bayesian approaches

- Creates "algorithmic beliefs" on the fly, rather than using pre-specified, fixed prior. Automatically adapts to adversarial environments.
    - The first approach that allows Bayesian-type algorithms to operate without prior assumptions and be applicable in adversarial settings.
- Uses Bayesian posteriors for randomized estimation and decision making.
    - More principled and precise than existing frequentist algorithms.

Contributions: approach

A general theory that creates algorithmic beliefs to simulate worst-case environment and uses posteriors to make decisions
$\Rightarrow$ Synergizing Frequentist and Bayesian approaches

- Creates "algorithmic beliefs" on the fly, rather than using pre-specified, fixed prior. Automatically adapts to adversarial environments.
  - The first approach that allows Bayesian-type algorithms to operate without prior assumptions and be applicable in adversarial settings.
- Uses Bayesian posteriors for randomized estimation and decision making.
  - More principled and precise than existing frequentist algorithms.
- Introduces Algorithmic Information Ratio (AIR) as an optimization objective to create "algorithmic beliefs", as well as a complexity measure to bound the frequentist regret of any algorithm.
  - Develop a "principle of maximal AIR" to derive novel learning algorithms and unify existing ones.

Contributions: applications

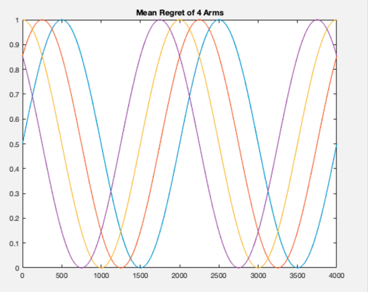
- In Multi-armed bandits (MAB), our proposed Adaptive Posterior Sampling (APS) algorithm achieves "best-of-all-worlds" empirical performance in stochastic, adversarial, and non-stationary environments!

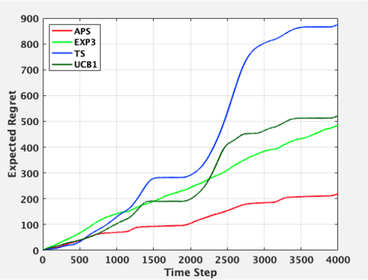
Contributions: applications

- In Multi-armed bandits (MAB), our proposed Adaptive Posterior Sampling (APS) algorithm achieves "best-of-all-worlds" empirical performance in stochastic, adversarial, and non-stationary environments!
- We also provide theoretical guarantees and insights to linear bandits, bandit convex optimization, and reinforcement learning.

Numerical evidence: changing environment

- Non-stationary MAB: generate a 4-armed bandit problem with the mean-reward structure showed in the left figure:

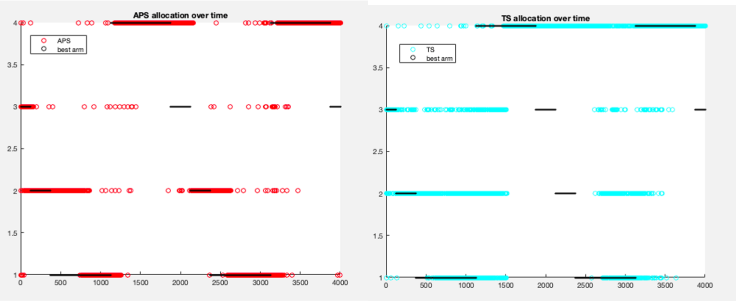




- APS achieves best performance, while TS fails in this non-stationary environment.

Numerical evidence: changing environment

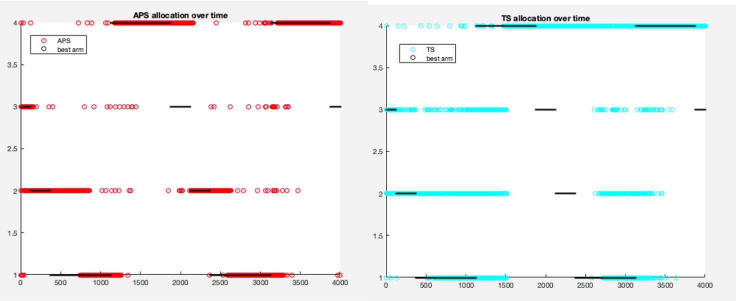
- Non-stationary MAB: generate the "sine curve" environment, track the selected arms and the best arms throughout the process.



- APS is highly responsive to changes in the best arm, whereas TS is relatively sluggish in this regard!

Numerical evidence: changing environment

- Non-stationary MAB: generate the "sine curve" environment, track the selected arms and the best arms throughout the process.



- APS is highly responsive to changes in the best arm, whereas TS is relatively sluggish in this regard!
- Creating new algorithmic beliefs has the potential to be a game changer.

Thanks!