Bandit Algorithms for Factorial Experiments

Yutong Yan[1], Audrey Durand[2], Joelle Pineau[1]
[1]School of Computer Science, McGill University [2]Computer Science and ECSE Department, Université Laval

Abstract

- **Multi-armed bandit (MAB)** problem is a classic Reinforcement Learning problem where a player faces multiple arms, each associated with a probability distribution over possible rewards [1].
- **Upper Confidence bound applied to trees (UCT)** algorithm has been popular in solving board games. UCT is naturally a better fit to solve Multivariate bandit problems than other approaches by treating the decision-making process as a bandit algorithm for tree search.
- **Why to use Bandit Algorithms for factorial experiments?** Multi-armed bandits minimize the opportunity cost of running an experiment. Previous work has shown Linear Thompson Sampling (LinTS) performs well than traditional heuristics [2].

Factorial Experiment

- Goal: identify the optimal sequence of choices
- Factorial design is composed of factors and choices per factor
  - Each node stores the **history** of previous choices, and each edge represents a **decision**.
  - The order of factors is predefined by the problem
    - i.e. Graphical design for mobile applications
    - i.e. Adaptive health intervention optimization

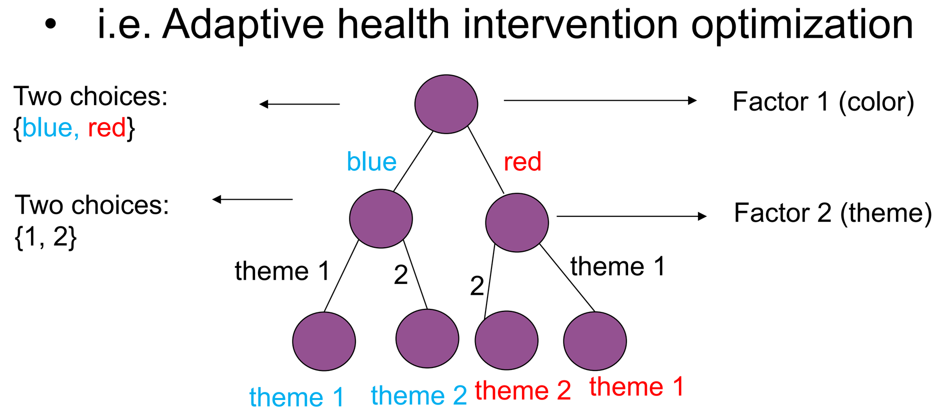

Figure 1: factorial design for graphical design with *2 factors x 2 choices per factor*

- Typically, there are **many factors** such as:
  - Gender, genotype, diet, and experimental protocols
    - Influence the outcome of the experiment
    - Determine the generality of a response.
- Traditional way: separate experiments (**A/B testing**) in each choice of one factor (very **wasteful**)
- To include several factors can **avoid** excessive number of experimental subjects.
- Detect **interactions** amongst intervention components.

Standard Bandit Formation

- **MAB** is a problem where a decision maker has many slot machines to choose from.

- Choose an action a_t from action set $\mathcal{A}$, and receives a reward r_t.
- Goal: maximize $\sum_{t=1}^{T} r_t$.
  - By choosing the **optimal action** $a_* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \mu_a$,

  where $\mu_a = \mathbb{E}[\mathcal{D}_a]$.
- To maximize the cumulative reward is equivalent to minimize the cumulative regret

$$\sum_{t=1}^{T} [\mu_{a_*} - \mu_{a_t}]$$

- **Upper Confidence Bound (UCB)**
  - Pick the arm $a_t = \underset{a \in \mathcal{A}}{\operatorname{argmax}} B_{a,t}$.
  - The most popular choice is UCB1:

$$B_{a,t} = \hat{\mu}_{a,t} + \sqrt{\frac{2\ln t}{N_{a,t}}}$$

- **Standard bandit formation** in Factorial experiments:
  - Given M factors and N choices per factor.
    - N^M actions if treating each *experimental unit* as an action.

Linear Bandit Formation

- **Linear bandit (LB)**
  - Choose an action x_t, and receives a reward r_t.
  - Optimal action

$$x_* = \underset{x \in \boldsymbol{X}_t}{\operatorname{argmax}} < \theta_*, x > = \underset{x \in \boldsymbol{X}_t}{\operatorname{argmax}} \mu_x$$

  - Action selection policy

$$x_t = \underset{x \in \boldsymbol{X}_t}{\operatorname{argmax}} < \hat{\theta}, x >$$

  - Reward is a linear combination of x_t and θ_*

$$r_t = < \theta_*, x_t >$$

- **LB formation** in Factorial experiments:
  - Given M factors and N choices per factor.
    - N^M actions (binary vector).

Tree Search Bandit Formation

- **Bandit Algorithm for Tree Search (BATS)**
  - Start from root node v_0, and running trajectory for depth d =1, 2, …, D.
  - At depth d, select a child node $v' \in \mathcal{C}(v_d)$.
  - Set $v_d \leftarrow v'$.
  - At depth D, the player receives a reward r_t.
- **UCB applied to trees (UCT)** [3]
  - Combines UCB1 and BATS.
  - One of the most popular algorithms in board games.
- **UCT-Laplace**
  - Uses tighter concentration bound from [4].
  - Explicit control of failure probability δ.
  - With confidence 1 - δ, selection rule is

$$B_{v',t} = \hat{\mu}_{v',t} + \sqrt{\frac{(1 + \frac{1}{N_{v',t}})\ln(K\sqrt{N_{v',t} + 1}/\delta)}{2N_{v',t}}},$$

  where K is the number of choices.
- **BATS formation** in Factorial experiments:
  - Given M factors and N choices per factor
    - Tree with depth M and N nodes at each tree level

Results

Using synthetic experiments, we show
- UCT-Laplace significantly improves the performance of standard UCT.
- Robustness and outperformance of BATS formation than MAB and LB formation.

- **UCT vs UCT-Laplace**

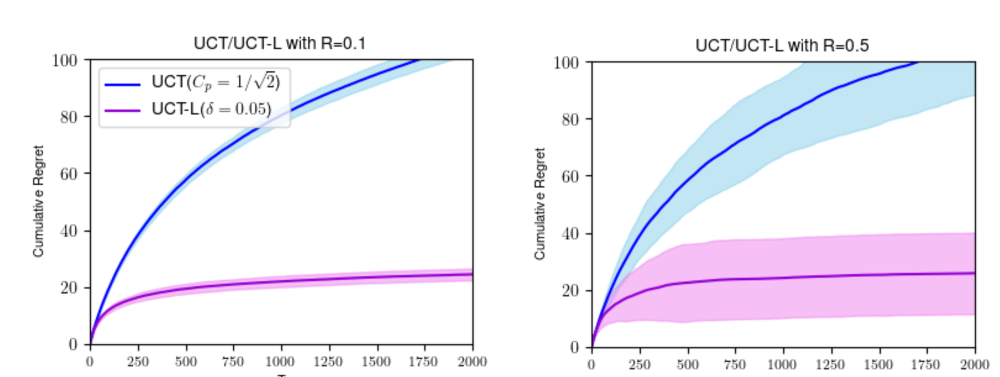

Figure 1: factorial design with *3 factors x 2 choices per factor* with *linear* outcome function

- **Varying number of choices per factor**

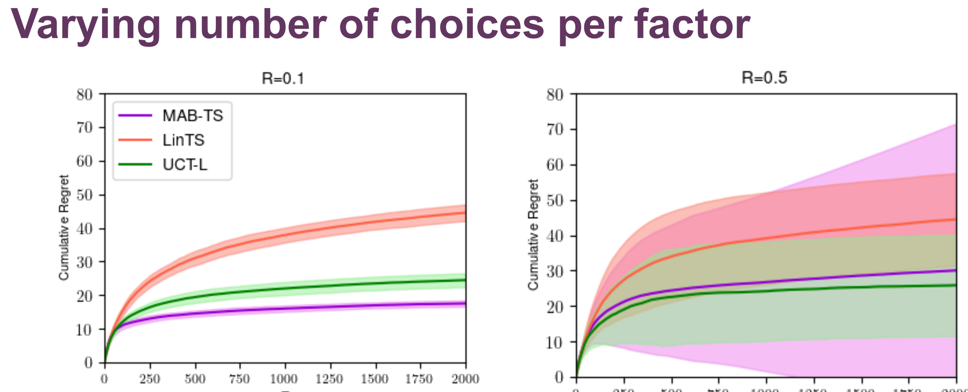

Figure 2: factorial design with *3 factors x 2 choices per factor* with *linear* outcome function

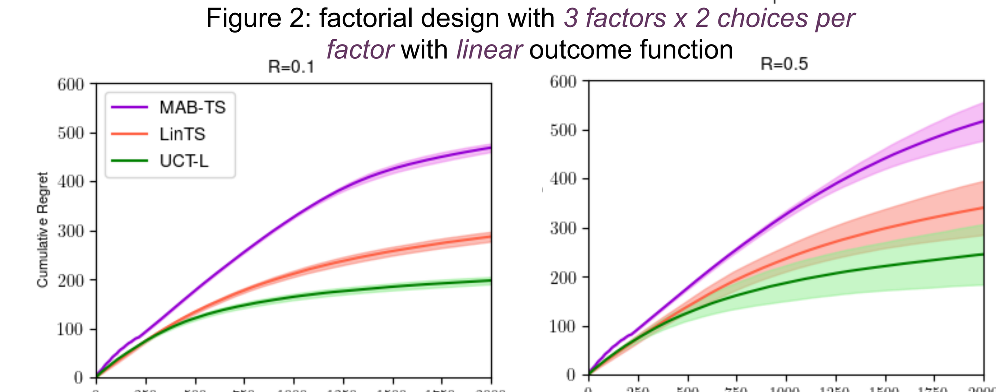

Figure 3: factorial design with *3 factors x 6 choices per factor* with *linear* outcome function

- Simple setting
  - Few actions under MAB formation.
  - Standard bandit algorithms are preferred.
- Complex setting
  - BATS formation can capture the underlying structure.
  - Algorithms under BATS formation seem more robust to noise variance

- **Varying number of factors**

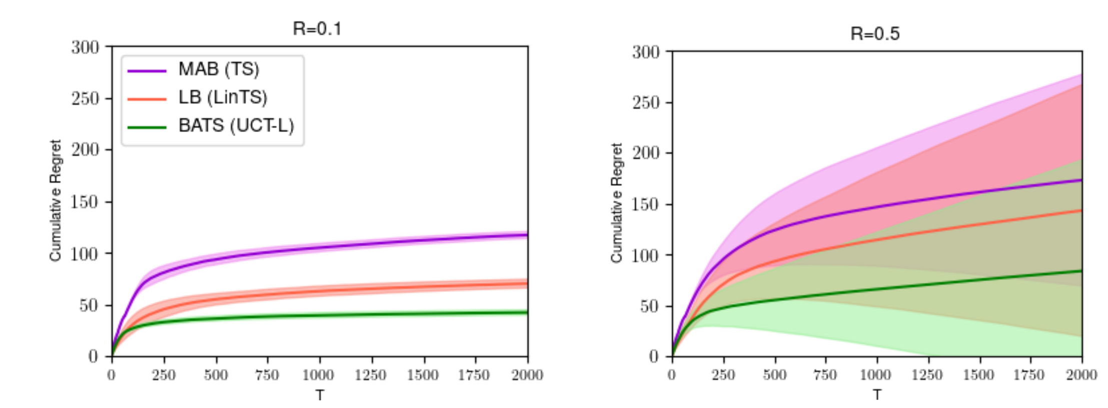

Figure 4: factorial design with *6 factors x 2 choices per factor* with *linear* outcome function

- **Non-linear Outcome Function**

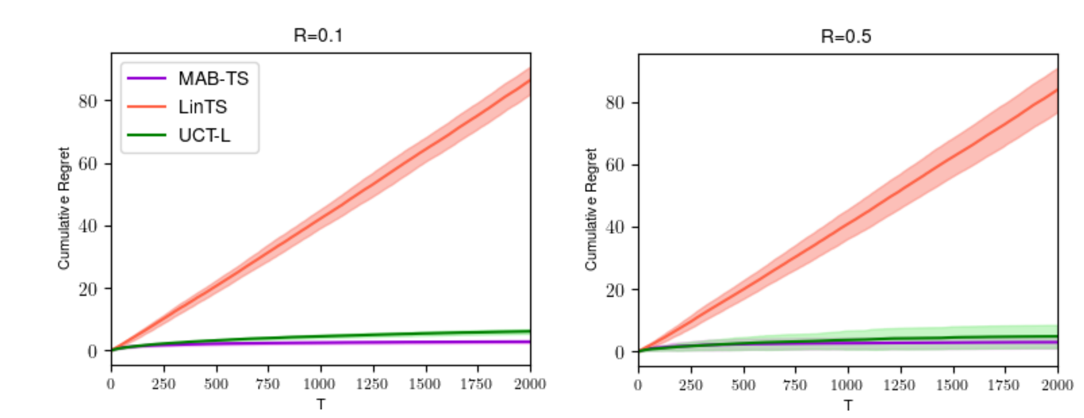

Figure 5: factorial design with *3 factors x 2 choices per factor* with *max* outcome function

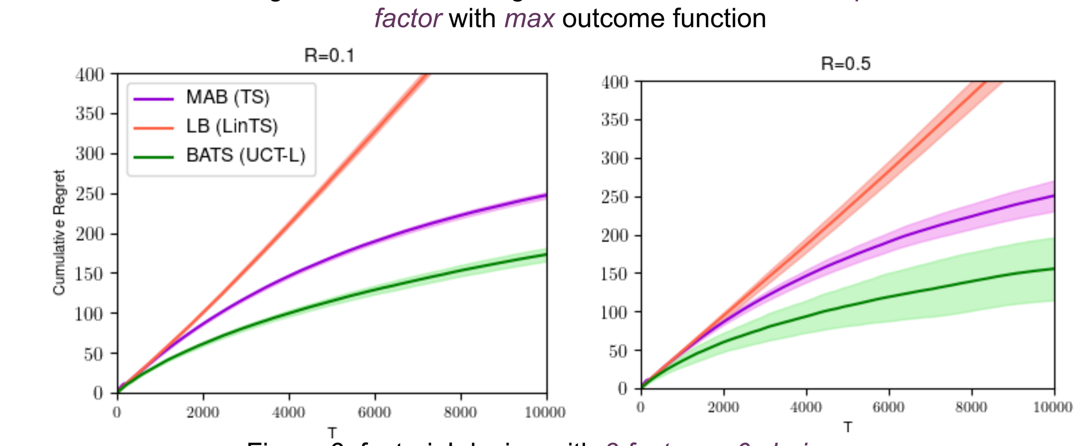

Figure 6: factorial design with *3 factors x 6 choices per factor* with *max* outcome function

- Simple setting
  - LB formation breaks due to strong linearity assumption.
  - Standard bandit algorithms are still preferred.
- Complex setting
  - BATS formation can capture the underlying structure.
  - BATS employs information-sharing.
  - BATS formation outperforms LB and MAB formation.

Conclusion

- We have employed bandits algorithms for tree search in factorial experiments. Our contributions include a proposed UCT-based algorithm, which significantly improve the performance of the standard UCT, and BATS formation in factorial experiments, algorithms under which are robust and best-performing under complex settings.

References

[1] Robbins, H. (1952). Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 58(5), 527-535.
[2] Scott, S. L. (2010). A modern Bayesian look at the multi-armed bandit. Applied Stochastic Models in Business and Industry, 26(6), 639-658.
[3] Kocsis, L., & Szepesvári, C. (2006, September). Bandit based monte-carlo planning. In European conference on machine learning (pp. 282-293). Springer, Berlin, Heidelberg.
[4] Abbasi-Yadkori, Y., Pál, D., & Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In Advances in Neural Information Processing Systems (pp. 2312-2320).