Online Batch Decision-Making with High-Dimensional Covariates

Chi-Hua Wang
Purdue University

Guang Cheng
Purdue University

Abstract

We propose and investigate a class of new algorithms for sequential decision making that interacts with *a batch of users* simultaneously instead of *a user* at each decision epoch. This type of batch models is motivated by interactive marketing and clinical trial, where a group of people are treated simultaneously and the outcomes of the whole group are collected before the next stage of decision. In such a scenario, our goal is to allocate a batch of treatments to maximize treatment efficacy based on observed high-dimensional user covariates. We deliver a solution, named *Teamwork LASSO Bandit algorithm*, that resolves a batch version of explore-exploit dilemma via switching between teamwork stage and selfish stage during the whole decision process. This is made possible based on statistical properties of LASSO estimate of treatment efficacy that adapts to a sequence of batch observations. In general, a rate of optimal allocation condition is proposed to delineate the exploration and exploitation trade-off on the data collection scheme, which is sufficient for LASSO to identify the optimal treatment for observed user covariates. An upper bound on expected cumulative regret of the proposed algorithm is provided.

1 Introduction

We consider a high-dimensional online batch decision making problem, a setting in which the decision-maker must interacts with *a group of users*, instead of *a user*, at each decision epoch. Such setting arises very naturally in real-world applications but received less attention in the literature. In interactive marketing Bertsimas and Mersereau (2007), marketers choose among a set of marketing messages to be sent to several customers simultaneously; updating marketing strategy once receiving a feedback is computationally impractical, and a more practical approach is to aggregate data in a prescribed length of period before adopting new strategy. In clinical trials Ahuja and Birge (2016), physicians choose among a set of available treatments to be administered to a group of patients simultaneously; updating treatment policy once measuring a response is unrealistic, and therapy in real practice is to collect data in a pre-approved length of period before designing new policy. In real-world practice, applicability of online decision-making methodology turns out to be impeded by such *limited adaptivity*. Similar issue has been addressed by Bai et al. (2019) in reinforcement learning setting, but it remains open in the setting of online decision making under bandit feedback.

One feature of modern economy digitization for decision makers is to deliver *personalized* products, services, and solutions based on individual-level data. Additionally, in many practical settings, individual-level data are *high-dimensional* but typically only a small number of the observed covariates are decisive. An additional layer of complexity in online batch decision-making is that, the whole decision making process is learned from bandit feedback: decision makers only observe the outcome for the product that was delivered, but not for any other available products that could have been delivered. Taking heterogeneity and bandit feedback into account, approaches of approximate dynamic programming or Markov decision process in Ahuja and Birge (2016) addressed the dimensionality issue by transforming the covariate space into a finite number of states and then solving the corresponding Bellman equation. However, how to construct such a transformation is often unspecified in this line of approach and particularly ambiguous in the case of high-dimensional covariate space, impeding itself to embrace the blessing of modern economy digitization.

In this paper, we propose a new class of approaches, named *Teamwork LASSO Bandit algorithm*, for online

Proceedings of the 23[rd]International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

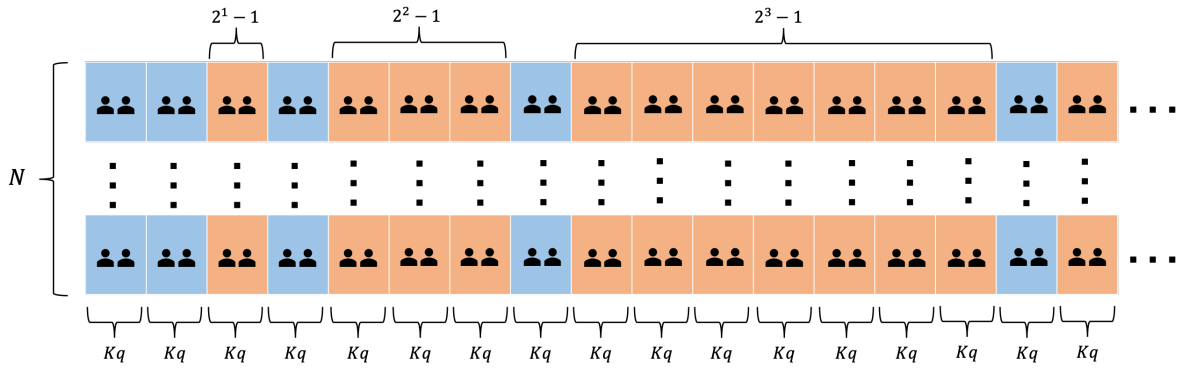

Figure 1: A realization of proposed Teamwork-Selfish policy in online batch decision making. N is the batch size. K is the number of available treatments. q is the number of repetition in a block. The agent runs teamwork mode in blue blocks and runs selfish mode in red blocks. The samples collected from blue blocks is called Teamwork sample. The samples collected from red blocks is called Selfish sample.

batch decision-making to tackle the curse of heterogeneity and high-dimensional covariate space under bandit feedback. To achieve both goals simultaneously, a deliberate balance between exploration and exploitation is required to efficiently learn a personalize decision-making rule that maximizes the cumulative treatment efficacy. In particular, in *Teamwork LASSO Bandit* algorithm, the agent switches between teamwork mode and selfish mode during the whole decision process to alternatively perform pure exploration and pure exploitation. By the proposed Teamwork-Selfish policy, the resulting sample set strikes a balance on optimal allocation in the sense that the rate of optimal allocation is beyond certain level for each treatment (see Definition 1) with high probability. This strategy ensures that the LASSO regression accommodates dependency arising in a sequence of batch observations and enjoys certain statistical properties in order to achieve optimal allocation for each incoming user.

Contributions. Our contribution is three-fold. First, we propose the "Teamwork LASSO Bandit algorithm," that solves the online batch decision-making problem with high-dimensional user covariates via learning LASSO estimates of treatment efficacy. Second, we propose a general "optimal allocation rate condition" on sample set as a theoretical guideline in designing a data collection scheme to identify the optimal treatment given observed user covariates. Such a scheme is illustrated by the proposed Teamwork-Selfish policy. Last, we establish an upper bound of the expected cumulative regret that scales linearly with the batch size. As a technical by-product, we develop a deviation inequality of LASSO regression that adaptes to a sequence of batch observations. To our knowledge, this is the first work addressing batch version of explore-exploitation dilemma in the setting of online batch decision-making.

Related works Batched bandit problem has received less research attention over the last decade. The special case of two-armed bandit setting has been addressed in Perchet et al. (2016); recently, Gao et al. (2019) extends the setting to K-armed bandit. While these works addressed the problem by mean model, we addressed batched bandit problem with high-dimensional linear model. In non-batch setting, Rusmevichientong and Tsitsiklis (2010) addressed the bandit problem by linear model in low dimensional setting; Bastani and Bayati (2020) then extends the setting to high-dimensional linear model. Comparing to these works, our model extends such setting to batched bandit problem.

A toy example for illustration

Consider the simplest scenario where the agent allocates one of two available treatments to each user in a size-2 batch at every decision epoch (That is, $N = 2, K = 2$ in Figure 1). In this case, each block contains $2q$ epochs and each epoch has two user. In the blue block, the agent runs in a teamwork mode: it sends both users to the first treatment at the first q epochs and to the second treatment at the second q epochs; in the red block, the agent runs in a selfish mode: it sends each user to her estimated optimal treatment based on all historical information over all $2q$ epochs. The agent runs in a teamwork mode (blue) on the block whose number is of power of 2, that is, $1, 2, 4, 8, 16, \cdots$ and runs in a selfish mode on the red block. The samples collected from blue blocks are called Teamwork sample; the samples collected from red blocks are called Selfish sample. More specifically, in red block, the agent runs a selfish mode by a two-step procedure: the first step is to run LASSO regression on Teamwork sample to output a set of optimal treatment candidates for the current user; the second step is to run LASSO regression on the union of Teamwork sample and Selfish sample to decide the optimal treatment for the current user.

Notation For any positive integer n, define $[n] = \{1, \cdots, n\}$ and for any $n_1 < n_2, [n_1 : n_2] = \{n_1, \cdots, n_2\}$ and $(n_1 : n_2] = \{n_1 + 1, \cdots, n_2\}$. $|A|$ denotes the number of elements for any collection A. For any vector v, notation $\text{supp}(v) \equiv \{i | v_i \neq 0\}$ denotes the indexes of non-zero coordinate. For a vector $u = (u_1, \cdots, u_d)$, $\|u\|_\infty \equiv \max_{i \in [d]} |u_i|$ denotes the maximum absolute value of its entries; $\|u\|_1 \equiv \sum_{i=1}^d |u_i|$ the L_1-norm.

2 Online Batch Decision Making

Treatment efficacy model We consider an agent, who needs to allocate one of K different available treatments (arms), denoted as $\mathcal{W} = \{w_k | k \in [K]\}$, for a size-$N$ batch of users in each decision epoch $t = 1, 2, \cdots, T$, where T denotes the length of the decision epoch horizon. The users in the current batch are represented by N observable vectors of features (covariates) $x_{i,t} \in \mathcal{X} \subseteq \mathbb{R}^d, i \in [N]$. We assume that feature vectors $x_{i,t}$ may vary across decision epoch and are sampled independently from a fixed, but a priori unknown distribution $\mathcal{P}_X$ with bounded support $\mathcal{X}$.

The efficacy of treatment w for the i*th* user (feedback) at epoch t has value $y(x_{i,t}, w)$, where the function y is unknown. At each decision epoch t, the agent allocates a batch of treatments $\{w(x_{i,t}) : i \in [N]\}$, and then observes a batch of efficacy $\{y(x_{i,t}, w(x_{i,t})) : i \in [N]\}$. The objective is to design an *allocation policy*, which maps users to their own treatments, that maximizes the cumulative sum of observed efficacy.

We assume that the efficacy of treatment w for user is a linear function of her covariates $x_{i,t}$, namely

$$y(x_{i,t}) = \langle \beta_w, x_{i,t} \rangle + \epsilon_{(i,t)}, \tag{1}$$

where $\{\epsilon_{(i,t)} : i \in [N], t \in [T]\}$ are martingale difference noise. At each epoch t, $\epsilon_{(i,t)}$'s are drawn independently from a mean zero σ-sub-Gaussian distribution (that is, $E[\exp(\lambda \epsilon_{(i,s)})] \leq \exp(\sigma^2 \lambda^2 / 2)$for all real λ) and $\{(\epsilon_{i,t}, \mathcal{F}_t)\}_{t \in [T]}$ forms a martingale difference sequence (that is, $E[\epsilon_{(i,t)} | \mathcal{F}_{t-1}] = 0$ for all $i \in [N]$). The noise is often introduced to account for the features that are not included in the model.

Efficacy parameters $\mathbf{B} \equiv \{\beta_{w_k} : w \in \mathcal{W}\}$ are a prior unknown to agent. Therefore, the agent deals with exploration-exploitation trade-off as it needs to choose between learning $\mathbf{B}$ and exploiting what has been learned so far to maximize treatment efficacy.

Our proposed algorithm exploits the structure (sparsity) of the feature space to improve its performance. For this purpose, let $s_{0,w} := \|\beta_w\|_0$ denote the number of nonzero coordinates of β_w. Define $s_0 = \max_{w \in \mathcal{W}} s_{0,w}$ and note that s_0 is a priori unknown to the agent.

Technical assumptions For ease of presentation, we assume that $\|x_{i,t}\|_\infty \leq x_{\max}$, for all $x_{i,t} \in \mathcal{X}$, and $\max_{w \in \mathcal{W}} \|\beta_w\|_1 \leq b$ for a known constant b.

We denote by Ω the set of feasible parameters, that is,

$$\Omega \equiv \{\beta \in \mathbb{R}^d : \|\beta\|_0 \leq s_0, \|\beta\|_1 \leq b\}, \tag{2}$$

and we write $\mathbf{B} \subseteq \Omega$ if $\beta \in \Omega$ for all $\beta \in \mathbf{B}$.

To measure the performance of proposed bandit algorithm, we present three technical assumptions.

Assumption 1. *(Margin Condition) There exists a constant $C_0 > 0$ such that for $w_i \neq w_j$ in $\mathcal{W}$, $P(0 < |X^\top \beta_{w_i} - X^\top \beta_{w_j}| \leq \kappa) \leq C_0 \kappa$ for all $\kappa > 0$.*

Assumption 1 is referred to the Margin Condition in the classification literature Tsybakov et al. (2004) and is introduced in multi-armed linear bandit literature to ensure only a small fraction of features can be drawn near the classification boundary $\{x : x^T(\beta_{w_i} - \beta_{w_j}) = 0\}$ in which efficacy of both treatments are almost equivalent; see Rusmevichientong and Tsitsiklis (2010), Bastani and Bayati (2020).

Assumption 2. *(Treatment Optimality Condition) There exist some constant $h > 0$ and two mutually exclusive sets, denoted as $\mathcal{W}_{opt}$ and $\mathcal{W}_{sub}$ with $\mathcal{W} = \mathcal{W}_{opt} \cup \mathcal{W}_{sub}$ such that*

(a) For each treatment w_i in $\mathcal{W}_{sub}$, it holds for every covariate vector $x \in \mathcal{X}$ that

$$\langle \beta_{w_i}, x \rangle < \max_{w \in \mathcal{W} \setminus \{w_i\}} \langle \beta_w, x \rangle - h. \tag{3}$$

(b) For each treatment w_i in $\mathcal{W}_{opt}$, there exists a constant $p_* > 0$ such that

$$\min_{w_i \in \mathcal{W}_{opt}} P(X \in U_{w_i}) \geq p_*, \tag{4}$$

where
$U_{w_i} \equiv \{x \in \mathcal{X} | \langle \beta_{w_i}, x \rangle > \max_{w \in \mathcal{W} \setminus \{w_i\}} \langle \beta_w, x \rangle + h\}$.

Assumption 2 is referred to the Treatment Optimality Condition in Rusmevichientong and Tsitsiklis (2010), Bastani and Bayati (2020) and is to separate available treatments into an optimal subset $\mathcal{K}_{opt}$ and a sub-optimal subset $\mathcal{K}_{sub}$ such that every optimal treatment $w \in \mathcal{K}_{opt}$ is strictly optimal for *some* users (denoted by the set U_w) and every sub-optimal treatment is strictly sub-optimal for *every* users.

Assumption 3. *(Compatibility Condition) There exists a constant $\phi_0 > 0$ such that for each optimal treatment $w \in \mathcal{W}_{opt}$, the population covariance matrix $\Sigma_w \equiv E[XX^\top | X \in U_w]$ belongs to the compatibility set of its treatment efficacy parameter β_w, that is,*

$$\Sigma_w \in \mathcal{C}(\mathit{supp}(\beta_{w_i}), \phi_0),$$

where $\mathcal{C}(I,\phi)$ is defined as

$$\mathcal{C}(I,\phi) \equiv \{M \in \mathbb{R}^{p\times p}_{\succeq 0} | \forall v \in \mathbb{R}^p \text{ such that } \|v_{I^c}\|_1 \leq 3\|v_I\|_1, \text{ we have} \|v_I\|_1^2 \leq |I| \frac{v^\top M v}{\phi^2}\}.$$

Assumption 3 is referred to as the Compatibility Condition in high-dimensional statistics literature Bühlmann and Van De Geer (2011) and is to ensure that LASSO estimate trained on samples $X \in U_w$ converges to the true parameter β_w with high probability as the number of samples grows to infinity.

Oracle Policy and Performance Metric We evaluate the performance of our algorithm using the usual notion of regret: the expected treatment efficacy loss compared with the oracle allocation policy that has full knowledge of $\mathbf{B}$, but not of the realizations of noise $\{\epsilon_{(i,t)} : i \in [N], t \in [T]\}$). Let us first characterize this benchmark policy. Based on the model (1), the optimal treatment allocation, denoted by w^*, is defined as

$$w^*(x_{i,t}) = \arg\max_{w\in\mathcal{W}} \{\langle \beta_w, x_{i,t}\rangle\}. \quad (5)$$

Throughout the paper, $w^*_{i,t}$ denotes the optimal treatment allocation for the ith user at epoch t.

We now define regret of a policy. Let π be the agent's policy that allocates the treatment $w_{i,t}$ to user $x_{i,t}$, and the choice of $w_{i,t}$ may depend on the information up to decision epoch $t-1$. The worst-case regret is defined as:

$$\text{Regret}_\pi(T) \equiv \max_{\mathbf{B}\in\Omega, \mathbb{P}_X \in Q(\mathcal{X})} E[\sum_{t=1}^{T}\sum_{i=1}^{N} (\langle \beta_{w^*_{i,t}}, x_{i,t}\rangle - \langle \beta_{w_{i,t}}, x_{i,t}\rangle)],$$

where the expectation is with respect to the distribution of martingale difference noise, $\epsilon_{i,t}$, and the distribution of feature vector, $\mathbb{P}_X$. The notation $Q(\mathcal{X})$ denotes a set of probability distributions with bounded support in $\mathcal{X}$. We want to point out that $\text{Regret}_\pi(T)$ corresponds to the usual cumulative expected regret in online learning when $N = 1$ and the empirical expected estimation error in batch learning when $T = 1$.

Rate of Optimal Allocation Condition Suppose the agent has collected a sample set $\mathcal{A}_w \equiv \{(x_{(i,t)}, y_{(i,t)})\}_{i=1,2,\cdots;t=1,2,\cdots}$ for treatment w. The set $\mathcal{A}_w$ consists of both optimal and sub-optimal allocation. The optimal allocation subsample set is defined as

$$\mathcal{A}^\sharp_w \equiv \{(\mathbf{X}_{(i,t)}, Y_{(i,t)}) \in \mathcal{A}_w | \mathbf{X}_{(i,t)} \in U_w\}. \quad (6)$$

Intuitively, bigger size of $\mathcal{A}^\sharp_w$ improves the accuracy of statistical procedure, which is the prize of achieving optimal allocation. On the contrary, bigger size of $\mathcal{A}_w$ undermines the accuracy of statistical procedure, which is the price of misallocating users to any suboptimal treatment. The ratio $|\mathcal{A}^\sharp_w|/|\mathcal{A}_w|$ matters and is termed as the *rate of optimal allocation* of sample set $\mathcal{A}_w$.

To gain a deeper insight of such balance into high-dimensional setting, consider the LASSO regression:

$$\widehat{\beta}_w(\mathcal{A}_w, \lambda) \equiv \arg\min_\beta \left\{ \frac{\|Y - X\beta\|_2^2}{|\mathcal{A}_w|} + \lambda\|\beta\|_1 \right\}, \quad (7)$$

where Y is a $|\mathcal{A}_w|$-dimension response vector and X is a $|\mathcal{A}_w| \times d$ covariate matrix.

Note that, if $\hat{\Sigma}(\mathcal{A}^\sharp_w)$ satisfies the compatability condition with constant ϕ, then $\hat{\Sigma}(\mathcal{A}_w)$ satisfies the compatability condition with constant $\phi\sqrt{|\mathcal{A}^\sharp_w|/|\mathcal{A}_w|}$. The above observation suggests that certain balance between $|\mathcal{A}^\sharp_{w_k}|$ and $|\mathcal{A}_w|$ should be stricken during the decision process. We make this intuition precise for sample sets characterization by introducing the *optimal allocation rate condition*:

Definition 1. *A sample set $\mathcal{A}_{w_k}$ satisfies the* ***rate r optimal allocation condition****, if it satisfies both conditions*

(i) Size of sample set: $|\mathcal{A}_w| \geq \frac{6\log d}{rC_2(\phi_1)^2}$

(ii) Rate of Optimal Allocation : $\frac{|\mathcal{A}^\sharp_w|}{|\mathcal{A}_w|} \geq \frac{r}{2}$.

We briefly call $\mathcal{A}_w$ a **rate r optimal allocation sample set** if $\mathcal{A}_w$ satisfies a optimal allocation condition of certain rate r. Note that the optimal allocation subsample set $\mathcal{A}^\sharp_w$ is not directly observable. The rate r optimal allocation condition is to lower bound the size of $\mathcal{A}^\sharp_w$ to ensure enough accuracy of LASSO estimator.

Our first contribution is a deviation inequality for LASSO regression based on the sample set collected in online batch decision making setting:

Theorem 1. ***(Deviation inequality for batch-adapted LASSO)*** *Given a rate r optimal allocation sample set $\mathcal{A}_{w_k}$ follows the dependence structure of online batch decision making problem. If $\lambda = \chi\phi^2/4s_0$, then for any $\chi > 0$, the oracle inequality holds that*

$$\begin{aligned} & P(\|\widehat{\beta}(\mathcal{A}_{w_k}, \lambda) - \beta_{\omega_k}\|_1 > \chi) \\ \leq\ & 2\exp[-C_1(\frac{\phi_1\sqrt{r}}{2})|\mathcal{A}_{w_k}|\chi^2 + \log d] \\ +\ & \exp[-|\mathcal{A}^\sharp_{w_k}|C_2(\phi_1)^2]. \end{aligned} \quad (8)$$

Theorem 1 is a general version of LASSO deviation inequality for adapted observations (see Proposition 1 in Bastani and Bayati (2020)). Our contribution is to extend the deviation inequality for adapted sequence of batch observations with martingale difference noise.

3 Teamwork LASSO Bandit Algorithm

In this section, we propose the Teamwork LASSO Bandit algorithm that runs in an interactive fashion: the agent switches between *Teamwork* mode (for pure exploration) and *Selfish* mode (for pure exploitation). The collection of decision epochs that the agent runs in teamwork mode and selfish mode are called teamwork stage and selfish stage, respectively. In the former stage, the agent allocates the current batch of users to a prescribed and possibly sub-optimal treatment for studying treatment efficacy. In the latter stage, the agent allocates each user in the current batch to her estimated optimal treatment for maximizing treatment efficacy. Epochs in teamwork and selfish stages are collected into sets $\mathbb{T}$ and $\mathbb{T}^c(\equiv [T] \setminus \mathbb{T})$, respectively.

3.1 Allocation in Teamwork mode

The teamwork stage $\mathbb{T}$ consists of exploration on all available treatments; that is, $\mathbb{T} = \cup_{w_k \in \mathcal{W}} \mathbb{T}_{\cdot,w_k}$, where $\mathbb{T}_{\cdot,w_k}$ denotes the prescribed decision epochs for studying efficacy of treatment w_k. We defer the exact specification of $\mathbb{T}_{\cdot,w_k}$ to Section 4.1.

Our treatment allocation policy now runs in Teamwork mode that, in any decision epoch $t \in \mathbb{T}_{\cdot,w_k}$, the agent allocates the whole batch of users to the treatment w_k:

$$\pi(x_{i,t}) \equiv w_k \text{ if } t \in \mathbb{T}_{\cdot,w_k} \quad \text{(Teamwork Stage)},$$

for all $i \in [N]$. Such an allocation results in a batch of covariate-efficacy pairs, $\mathcal{T}_{t,w_k} \equiv \{(x_{i,t}, y(x_{i,t}, w_k)) : i \in [N]\}$, which is aggregated into the teamwork sample set up to epoch t $\mathcal{T}_{[t],w_k} \equiv \mathcal{T}_{[t-1],w_k} \cup \mathcal{T}_{t,w_k}$.

3.2 Allocation in Selfish mode

The selfish stage $\mathbb{T}^c$ deploys exploitation by allocating the current best-possible treatment to each user in the current batch. This is done by a two-step procedure: estimating candidates of personal optimal treatment and then committing selfish allocation. Such an allocation results in another batch of covariate-efficacy pairs, $\{(x_{i,t}, y(x_{i,t}, w_{i,t})) : i \in [N]\}$. Then, for each treatment w_k, the set $\mathcal{E}_{t,w_k} \equiv \{(x_{i,t}, y(x_{i,t}, w_{i,t})) : i \in [N], w_{i,t} = w_k\}$ is aggregated into the selfish sample set up to epoch t $\mathcal{E}_{[t],w_k} \equiv \mathcal{E}_{[t-1],w_k} \cup \mathcal{E}_{t,w_k}$.

Here we outline the above two-step procedure whose details are deferred to Section 4.2. At a selfish decision epoch $t \in \mathbb{T}^c$, the agent first estimates β_w by using LASSO regression based on teamwork sample set $\mathcal{T}_{[t],w}$ and the resulting estimator $\widehat{\beta}_w(\mathcal{T}_{[t],w})$ is called Teamwork LASSO. The Teamwork LASSO screens out the sub-optimal treatments for each user $x_{i,t}$ in current batch and then outputs a set of optimal treatment candidates $\hat{\mathcal{K}}(x_{i,t})$(See Eq.(11) for detailed description).

Algorithm 1: `TeamworkLASSOBandit`(q, h)

Given parameters q, h, λ_1, $\lambda_{2,0}$
Initialize $\mathcal{T}_{[0],t}, \mathcal{E}_{[0],t}$
for $t \in \{1, \ldots, T\}$ **do**
  for $w \in \mathcal{W}$ **do**
    $\mathcal{T}_{[t],w} = \mathcal{T}_{[t-1],w}$; $\mathcal{E}_{[t],w} = \mathcal{E}_{[t-1],w}$
  end
  if $t \in \mathbb{T}$ **then**
    if $t \in \mathbb{T}_{\cdot,w}$ **then**
      for $i \in \{1, \ldots, N\}$ **do**
        Observe $x_{i,t}$
        Allocate $w_{i,t} = w$
        Observe $y(x_{i,t}, w)$
        $\mathcal{T}_{[t],w} = \mathcal{T}_{[t],w} \cup \{(x_{i,t}, y(x_{i,t}, w))\}$
      end
    end
  else
    for $w \in \mathcal{W}$ **do**
      Compute $\widehat{\beta}_w(\mathcal{T}) = \widehat{\beta}_w(\mathcal{T}_{[t-1],w})$
      Compute $\widehat{\beta}_w(\mathcal{S}) = \widehat{\beta}_w((\mathcal{T} \cup \mathcal{E})_{[t-1],w})$
    end
    for $i \in \{1, \ldots, N\}$ **do**
      Observe $x_{i,t}$
      Select $\widehat{K}(x_{i,t}) = \{w | \langle x_{i,t}, \widehat{\beta}_w(\mathcal{T}) \rangle \geq \max_{\tilde{w} \in \mathcal{K} \setminus \{w\}} \langle x_{i,t}, \widehat{\beta}_{\tilde{w}}(\mathcal{T}) \rangle - \frac{h}{2}\}$
      Allocate $w_{i,t} = \arg\max_{w \in \widehat{K}(x_{i,t})} \langle x_{i,t}, \widehat{\beta}_w(\mathcal{S}) \rangle$
      Observe $y(x_{i,t}, w_{i,t})$
      $\mathcal{E}_{[t],w_{i,t}} = \mathcal{E}_{[t],w_{i,t}} \cup \{(x_{i,t}, y(x_{i,t}, w_{i,t}))\}$
    end
  end
end

The agent then estimates β_w by using LASSO regression based on full sample set $(\mathcal{T} \cup \mathcal{E})_{[t],w}$ and the resulting estimator $\widehat{\beta}_w((\mathcal{T} \cup \mathcal{E})_{[t],w})$ is called All LASSO. The All LASSO then selects the treatment with highest expected efficacy for the user $x_{i,t}$:

$$\pi(x_{i,t}) \equiv \arg \max_{w_k \in \widehat{\mathcal{K}}(x_{i,t})} \langle \widehat{\beta}_{w_k}((\mathcal{T} \cup \mathcal{E})_{[t],w}), x_{i,t} \rangle \quad \text{if} \quad t \notin \cup_{w_k \in \mathcal{W}} \mathbb{T}_{\cdot,w_k} \quad \text{(Selfish Stage)}.$$

Observe that by design of our interactive policy (see Figure 2), the agent maintains for each treatment w two different sample sets (teamwork and greedy). The independence in teamwork sample set is preserved by the pure exploration nature of teamwork stage, facilitating the subsequent analysis of LASSO estimate. However, the full sample set mixes independence from teamwork stage with the dependency arising in the selfish stage, complicating the subsequent analysis of LASSO estimate. Tackling such dependency requires a closer look at every epoch of decision making process.

4 A Teamwork-Selfish Policy

4.1 Planing Teamwork Stage

One factor contributes to the success of our Teamwork-Selfish policy is the design of $\mathbb{T}_{[t],w_k}$ in teamwork mode (See eq.(3.1)). Here we explain our design of $\mathbb{T}_{[t],w_k}$ and reveal the quantity q that characterized the complexity of online batch decision making problem.

Recall that the collection of teamwork decision epochs up to epoch t is the set $\mathbb{T}_{[t],w_k}$. Such set is a truncation to epoch range $[t]$ of the collection of treatment w_k teamwork decision epoch $\mathbb{T}_{\cdot,w_k}$, which is developed by the *teamwork rounds* $\{\mathbb{T}_{n,w_k}\}_{n=1}^{\infty}$.

The *nth teamwork round for treatment* w_k $\mathbb{T}_{n,w_k}$ is defined as a prescribed range of decision epochs

$$\mathbb{T}_{n,w_k} \equiv \{(2^n-1)\times Kq+j | j \in [q(k-1)+1, qk]\}. \quad (9)$$

(Refer to the blue block in Figure.1) Observe that each teamwork round prescribes q decision epochs for treatment pure exploration. The magnitude q characterizes the complexity of a online batch decision problem.

Our theoretical results suggests a lower bound of q by $4\lceil q_0 \rceil$(See Eq.(10) for q_0). Such lower bound ensures both teamwork sample set and full sample set collected from the Teamwork-Selfish policy to satisfy the rate of optimal allocation condition (Definition 1) with high probability. Consequently, the Teamwork LASSO and ALL LASSO enjoy their statistical properties and the regret bound (2) can be guaranteed.

Remark that our lower bound on q is proportional to $1/N$, where N is the batch size . Compare to the full adaptive setting in Bastani and Bayati (2020), we have $q_{\text{batch}} \approx q_{\text{non-batch}}/N$. In terms of update frequency, LASSO Bandit requires $Kq_{\text{non-batch}}(NT - logNT)$, while `Teamwork LASSO Bandit` requires $K(q_{\text{non-batch}}/N)(T - logT)$. In terms of regret, LASSO Bandit is of rate $K[log(NT)]^2$, while `Teamwork LASSO Bandit` is of rate $KN(logT)^2$. Thus, there is a trade off between regret and update frequency.

Our design of teamwork stage $\mathbb{T}_{\cdot,w}$ is a generalization of the forced sampling for the two-arm and non-batch setting in Rusmevichientong and Tsitsiklis (2010) and for the K-arm and non-batch setting in Bastani and Bayati (2020). Our contribution is to redesign such type of policy when batch is the fundamental sampling unit to accommodate the restrictions from limited adaptivity.

4.2 Two Step Procedure in Selfish Stage

Another factor contributing to the success of our Teamwork-Selfish policy is the two step procedure to commit selfish allocation. In a selfish decision epoch t, the agent first estimates every treatment's efficacy based on information up to decision epoch $(t-1)$ by

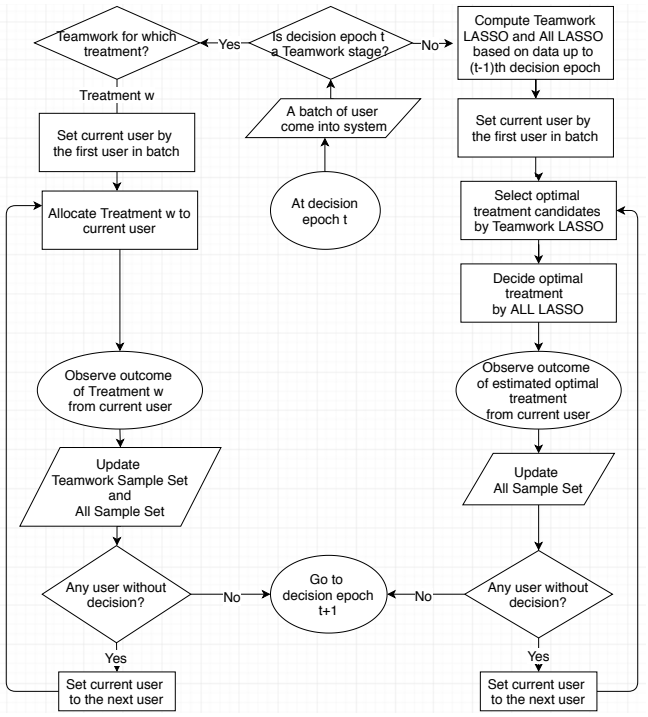

Figure 2: Teamwork-Selfish policy.

Teamwork LASSO $\widehat{\beta}_w(\mathcal{T}_{[t-1],w}, \lambda_1)$ to separate optimal treatments from suboptimal treatments. Then, the agent estimates candidate treatments' efficacy by ALL LASSO $\widehat{\beta}_w((\mathcal{T} \cup \mathcal{E})_{[t-1],w}, \lambda_{2,t-1})$ to identify the true optimal treatment for current user. In particular, the success of step 1 relies on a "good event" E_t, defied as

$$E_t \equiv \cap_{w_k \in \mathcal{W}} \{\|\widehat{\beta}_{w_k}(\mathcal{T}_{[t],k}, \lambda_1) - \beta_{w_k}\|_1 \leq \frac{h}{4x_{\max}}\},$$

which marks the case that every Teamwork LASSO are accurate enough to screen out suboptimal treatments.

Now we give a high-probability statement for E_{t-1} to access the performance of Step 1.

Lemma 1. *For $t \geq (Kq)^2$, $P(E_t) \geq 1 - 5K/t^4$.*

Lemma 1 is an immediate consequence of Corollary 1

Corollary 1. *(Deviation inequality for Teamwork LASSO) For all treatments $w_k \in \mathcal{W}$, if $t \geq (Kq)^2$ for $q \geq 4\lceil q_0 \rceil$, the Teamwork LASSO estimator satisfies the deviation inequality (set $\lambda_1 = \frac{\phi_0^2 p_* h}{64 s_0 x_{\max}}$)*

$$P(\|\hat{\beta}_{w_k}(\mathcal{T}_{[t],w_k}, \lambda_1) - \beta_{w_k}\|_1 > \frac{h}{4x_{\max}}) \leq \frac{5}{t^4},$$

where

$$q_0 \asymp \frac{1}{N} \max\left\{\frac{\log d}{p_*}, \frac{x_{\max}^2 \log d}{h^2 p_*^2}\right\} \quad (10)$$

Corollary 1 is an application of Theorem 1, given a deviation inequality of LASSO regression based on

$\mathcal{T}_{[t],w_k}$. As shown in lemma 7, $\mathcal{T}_{[t],w_k}$ is a rate p_* optimal allocation sample set with probability at least $1 - \exp(2/t^4)$.

Step 1: Screen out Sub-optimal Treatments

Given a current user x, the agent's objective at step 1 is to output a set of user's potential optimal treatments. To construct such set of treatment candidates, we require that a candidate treatment should have its estimated treatment efficacy almost as good as the maximum of estimated treatment efficacy of all available treatments, as the Teamwork LASSO can tell.

Formulating such intuition motivates us to define the set of *personal optimal treatment candidates* $\widehat{\mathcal{K}}(x)$:

$$\begin{aligned}\widehat{\mathcal{K}}(x) \equiv \{w_k \in \mathcal{W} : \langle x, \hat{\beta}_{w_k}(\mathcal{T}_{[t-1],w_k}, \lambda_1)\rangle \\ \geq \max_{w_k \in \mathcal{W}} \langle x, \hat{\beta}_{w_k}(\mathcal{T}_{[t-1],w_k}, \lambda_1)\rangle - \frac{h}{2}\}\end{aligned} \tag{11}$$

The exact members in candidate set $\hat{\mathcal{K}}(x)$ depend on the region that x belongs to. Recall that $w^*(x) = \arg\max_{w \in \mathcal{W}} \langle x, \beta_w \rangle$ denotes the optimal treatment of x. For the case $x \in U_{w_k}$, $w^*(x) = w_k$ by definition of U_{w_k}. As shown in lemma 13, given the event E_{t-1} holds ,the candidate set $\hat{\mathcal{K}}(x)$ contains *only* the optimal treatment of x, that is,

$$\hat{\mathcal{K}}(x) = \{w^*(x)\} \quad \text{if } x \in U_{w_k}.$$

Therefore, the agent definitely has an optimal allocation for every user $x \in U_{w_k}$ under the event E_{t-1}.

For the case $x \in \mathcal{X} \setminus \cup_{w \in \mathcal{W}} U_w$, the user covariate x lies near a decision boundary $\{x : \langle x, \beta_{w^*(x)} - \beta_{w_j} \rangle = 0\}$ for some comparable treatment w_j. As shown in lemma 12, the candidate set $\hat{\mathcal{K}}(x)$ contains *at least* the optimal treatment of x, that is,

$$w^*(x) \in \hat{\mathcal{K}}(x) \quad \text{if } x \in \mathcal{X} \setminus \cup_{w \in \mathcal{W}} U_w,$$

but may contain other comparable treatments w_j that performs almost equally well as the optimal treatment $w^*(x)$, under the event E_{t-1}.

Step 2: Commit Selfish Allocation The agent's objective in step 2 is to commit a selfish allocation for current user x. Such selfish allocation is done by allocating user x to the treatment with highest estimated efficacy, as far as the All LASSO can tell:

$$\pi(x) \equiv \arg \max_{w_k \in \widehat{K}(x)} \langle x, \widehat{\beta}_{w_k}((\mathcal{T} \cup \mathcal{E})_{[t-1],w_k}, \lambda_{2,t})\rangle \tag{12}$$

An application of Theorem 1 gives a deviation inequality of LASSO regression based on $(\mathcal{T} \cup \mathcal{E})_{[t],w_k}$:

Corollary 2. *(Deviation inequality for All LASSO) For treatments $w_k \in \mathcal{W}_{opt}$, if $t \geq C_5 \equiv \min\{t : t \geq 24Nq \log t + 4(Kq)^2\}$, the All LASSO estimator satisfies the deviation inequality (Set $\lambda_{2,t} = \frac{\phi_0^2}{2s_0}\sqrt{\frac{\log t + \log d}{p_* C_1(\phi_0)} \frac{1}{t}}$)*

$$\begin{aligned}&\|\hat{\beta}_{w_k}((\mathcal{T} \cup \mathcal{E})_{[t],w_k}, \lambda_{2,t}) - \beta_{w_k}\|_1 \\ &> \frac{16}{\sqrt{p_*^3 C_1(\phi_0)}}\sqrt{\frac{\log t + \log d}{t}}\end{aligned} \tag{13}$$

with probability at least $2(\frac{1}{t} + \exp(-\frac{p_*^2 C_2(\phi_0)^2}{32} \cdot t)$.

5 Regret Analysis

The following theorem bounds the regret of our Teamwork-Selfish treatment allocation policy.

Theorem 2. *Suppose Assumptions 1, 2 and 3 hold. Then, the regret of the Teamwork-Selfish policy over decision horizon $[T]$ satisfies*

$$\begin{aligned}Regret_\pi(T) &\leq N\{[C_3](\log T)^2 \\ &\quad + [2bx_{\max}K(6q+2) + C_3 \log d] \log T \\ &\quad + [2bx_{\max}(C_5 + K(1+4C_4))]\} \\ &= O(NKs_0^2\sigma^2[\log T + \log d]^2)\end{aligned}$$

Below we provide a roadmap for the proof of Theorem 2. The proof is motivated by the regret analysis in Bastani and Bayati (2020). Our contribution is to generalize the approach for regret analysis in online batch decision making setting.

1. Regret Guarantee in Good and Bad Epochs

To reason about different sources of regret contribution, we decompose the decision process into four subcases (i-iv) so that we can examine each one independently:

(i) During initialization period ($t \leq C_5$)
- After initialization ($t > C_5$)
  (ii) In teamwork stage ($t \in \mathbb{T}$)
  - In selfish stage ($t \notin \mathbb{T}$)
    (iii) Event E_t does not hold.
    (iv) Event E_t does hold.

As shown in lemma 14 in section F.3, the expected regret in case (iv) is guaranteed to be bounded by the function $f(t)$ as

$$\begin{aligned}f(t) &= [4Kbx_{\max} + C_3(\phi_0, p_*) \cdot \log d]/t \\ &\quad + 8Kbx_{\max} \exp[-(p_*^2 C_2(\phi_0)^2/32) \cdot t] \\ &\quad + C_3(\phi_0, p_*)(\log t/t).\end{aligned} \tag{14}$$

For this reason, we define case (iv) as good epochs

$$G_t = I(t > C_5, t \in \mathbb{T}, \text{Event } E_t \text{ holds}). \tag{15}$$

and then we interpret lemma 14 in section F.3 as

$$r_{i,t} I(G_t) \leq f(t) I(G_t). \tag{16}$$

Outside the good epochs are bad epochs G_t^c. In a bad epoch, allocations cannot be guaranteed to be optimal, due to the pure exploration nature in teamwork mode or the insufficient accuracy of LASSO estimate in selfish mode, resulting in the worst-case regret guarantee $2bx_{\max}$ for cases (i-iii). We interpret the above fact as

$$r_{i,t}I(G_t^c) \leq 2bx_{\max}I(G_t^c) \tag{17}$$

.

2. Bounding Expected Instantaneous Regret Combine (16) and (17), the expected instantaneous regret is upper bounded by

$$\begin{aligned} E[r_t] &= E[r_t \cdot I(G_t)] + E[r_t \cdot I(G_t^c)] \\ &\leq E[f(t) \cdot I(G_t)] + E[2bx_{\max} \cdot I(G_t^c)] \\ &= f(t)P(G_t) + 2bx_{\max}P(G_t^c) \end{aligned} \tag{18}$$

3. Bounding Regret$_\pi(T)$ Apply (18) to gain

$$\begin{aligned} &E[\sum_{t=1}^{T}\sum_{i=1}^{N} r_{i,t}] \leq N\sum_{t=1}^{T} E[r_{i,t}] \\ &\leq N\int_0^T f(t)dt + 2Nbx_{\max}\int_0^T P(G_t^c)dt \end{aligned} \tag{19}$$

Note that, by lemma 1, we have $P(G_t^c) \leq 5K/t^4$.

6 Experiment

We illustrate the trade-off between regret and update frequency by comparing the cumulative regret between `LASSO Bandit` algorithm (high update frequency) and our `Teamwork LASSO Bandit` algorithm (low update frequency) in Figure 3 (Appendix, section H). Here we give remarks on the experiment:

1. In terms of the number of updates, say case q=1, LASSO Bandit (N=1) (high update frequency) requires $3[5000/3 - log(5000/3, 2)] \sim 4968$ updates, Teamwork LASSO Bandit (low update frequency) requires $3[5000/3/4 - log(5000/3/4, 2)] \sim 1224$ updates for N=4 case and $3[5000/3/12 - log(5000/3/12, 2)] \sim 396$ updates for N=12 case. Note that both give comparable regrets.

2. If the length of exploration phase q is sufficiently large, high update frequency algorithm has lower cumulative regret than low update frequency algorithm; if q is not sufficiently large, low update frequency algorithm outperforms high update frequency algorithm.

3. In general, the performance of high update frequency algorithm has higher variance than the performance of low update frequency algorithm. In particular, the performance of high update frequency algorithm is more sensitive to the increase in covariate dimension than our low update frequency algorithm.

In conclusion, high update frequency algorithm (`Lasso Bandit`) do have lower cumulative regret than low update frequency algorithm (`Teamwork Lasso Bandit`) if the length of exploration phase q is sufficiently large. However, it is hard to determine how large of q can be thought of as being sufficiently large in practice. On the other hand, low update frequency algorithm is immune from such a concern in the sense that we can simply set q=1 when we have a batch of new samples at every decision epoch.

7 Discussion and Conclusions

We have proposed a framework to address batch-version explore-exploit dilemma in the setting of online batch decision making with high dimensional covariate. In terms of regret analysis, we formulate the *rate of optimal allocation condition* on the collected sample set to characterize the underlying constraint behind the data collection scheme and to serve as a guideline for designing policy in bandit algorithms. Based on the rate of optimal allocation condition, we propose the `Teamwork LASSO Bandit` algorithm for sequential decision making. In theory, the cumulative total regret of the `Teamwork LASSO Bandit` algorithm of constant batch size N over finite time horizon T is shown to be bounded by $O(N(\log T)^2)$. In terms of observed covariate dimension p and sparsity parameter s_0, the cumulative total regret of the `Teamwork LASSO Bandit` algorithm grows as $O(s_0^2(\log p)^2)$.

In the end, we highlight a few particularly relevant questions that are left as future works. The first one is the minimax lower bound of regret over all possible algorithms solving batched bandit problem with covariates. In one pull situation, Rusmevichientong and Tsitsiklis (2010) showed the lower bound is $O(\log T)$. Recently, Wang et al. (2018) showed that the regret of Bastani and Bayati (2020) can be reduced from $O((\log T)^2)$ to $O(\log T)$ by invoking the Minimax Concave Penalized (MCP) penalty. Hence, the MCP-Bandit algorithm matches the oracle policy with high probability. We expect the regret of `Teamwork LASSO Bandit` algorithm can also be reduced from $O(N(\log T)^2)$ to $O(N \log T)$ if MCP penalty instead of the lasso one is adopted, although whether this rate matches the theoretical lower bound remains unknown. The second one is more relevant to practice: can we design an effective teamwork strategy when batch size is non-constant? In Grover et al. (2018), the authors have proposed four different kinds of delayed feedback mechanism that frequently happen in online advertising context, which may lead to non-constant batch size in our setting. When we are performing batch update in the above delayed feedback scenario, is there a guideline for algorithm design? In particular, can the rate of optimal allocation condition be extended to handle delayed feedback situation?

Acknowledgements

Chi-Hua Wang thanks Yang Yu (Purdue University) for discussion and Zhanyu Wang (Purdue University) for help on experiment. Guang Cheng would like to acknowledge support by NSF DMS-1712907, DMS-1811812, DMS-1821183, and Office of Naval Research (ONR N00014- 18-2759). While completing this work, Guang Cheng was a member of Institute for Advanced Study, Princeton and visiting Fellow of SAMSI for the Deep Learning Program in the Fall of 2019; he would like to thank both Institutes for their hospitality.

References

Abbasi-Yadkori, Y., Pal, D., and Szepesvari, C. (2012). Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, pages 1–9.

Ahuja, V. and Birge, J. R. (2016). Response-adaptive designs for clinical trials: Simultaneous learning from multiple patients. *European Journal of Operational Research*, 248(2):619–633.

Alon, N. and Spencer, J. H. (2004). *The probabilistic method*. John Wiley & Sons.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422.

Bai, Y., Xie, T., Jiang, N., and Wang, Y.-X. (2019). Provably efficient q-learning with low switching cost. *arXiv preprint arXiv:1905.12849*.

Bastani, H. and Bayati, M. (2020). Online decision making with high-dimensional covariates. *Operations Research*, 68(1):276–294.

Bertsimas, D. and Mersereau, A. J. (2007). A learning approach for interactive marketing to a customer segment. *Operations Research*, 55(6):1120–1135.

Bolton, P. and Harris, C. (1999). Strategic experimentation. *Econometrica*, 67(2):349–374.

Bühlmann, P. and Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Dimakopoulou, M., Athey, S., and Imbens, G. (2017). Estimation considerations in contextual bandits. *arXiv preprint arXiv:1711.07077*.

Gao, Z., Han, Y., Ren, Z., and Zhou, Z. (2019). Batched multi-armed bandits problem. *arXiv preprint arXiv:1904.01763*.

Goldenshluger, A. and Zeevi, A. (2013). A linear response bandit problem. *Stochastic Systems*, 3(1):230–261.

Grover, A., Markov, T., Attia, P., Jin, N., Perkins, N., Cheong, B., Chen, M., Yang, Z., Harris, S., Chueh, W., et al. (2018). Best arm identification in multi-armed bandits with delayed feedback. *arXiv preprint arXiv:1803.10937*.

Jun, K.-S., Jamieson, K. G., Nowak, R. D., and Zhu, X. (2016). Top arm identification in multi-armed bandits with batch arm pulls. In *AISTATS*, pages 139–148.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670.

Perchet, V., Rigollet, P., Chassang, S., Snowberg, E., et al. (2016). Batched bandit problems. *The Annals of Statistics*, 44(2):660–681.

Perchet, V., Rigollet, P., et al. (2013). The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41(2):693–721.

Rusmevichientong, P. and Tsitsiklis, J. N. (2010). Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411.

Tsybakov, A. B. et al. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166.

Wang, X., Wei, M., and Yao, T. (2018). Minimax concave penalized multi-armed bandit model with high-dimensional covariates. In *International Conference on Machine Learning*, pages 5200–5208.