Journal of the
American
Statistical
Association

March 2024 ▸ Volume 119 ▸ Number 545

Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uasa20

Nearly Dimension-Independent Sparse Linear Bandit over Small Action Spaces via Best Subset Selection

Yi Chen, Yining Wang, Ethan X. Fang, Zhaoran Wang & Runze Li

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2024, VOL. 119, NO. 545, 246–258: Theory and Methods
https://doi.org/10.1080/01621459.2022.2108816

Taylor & Francis
Taylor & Francis Group

Check for updates

Nearly Dimension-Independent Sparse Linear Bandit over Small Action Space Best Subset Selection

Yi Chen*[a], Yining Wang*[b], Ethan X. Fang[c], Zhaoran Wang[d] and Runze Li[e]

[a]Department of Industrial Engineering and Decision Analytics, Hong Kong University of Science and Technology, Hong Kong, China; [b]Naveen Jindal school of Management, University of Texas at Dallas, Richardson, TX; [c]Department of Biostatistics & Bioinformatics, Duke University, Durham, NC; [d]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL; [e]Department of Statistics, Pennsylvania State University, University Park, PA

ABSTRACT

We consider the stochastic contextual bandit problem under the high dimensional linear model. We focus on the case where the action space is finite and random, with each action associated with a randomly generated contextual covariate. This setting finds essential applications such as personalized recommendations, online advertisements, and personalized medicine. However, it is very challenging to balance the exploration and exploitation tradeoff. We modify the LinUCB algorithm in doubly growing epochs and estimate the parameter using the best subset selection method, which is easy to implement in practice. This approach achieves $O(s\sqrt{T})$ regret with high probability, which is nearly independent of the "ambient" regression model dimension d. We further attain a sharper $O(\sqrt{sT})$ regret by using the SupLinUCB framework and match the minimax lower bound of the low-dimensional linear stochastic bandit problem. Finally, we conduct extensive numerical experiments to empirically demonstrate our algorithms' applicability and robustness. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received August 2020
Accepted July 2022

KEYWORDS
Best subset selection; High-dimensional models; Regret analysis; Stochastic bandit

1. Introduction

Contextual bandit problems receive significant attention over the past years in different communities, such as statistics, operations research, and computer science (Bubeck and Cesa-Bianchi 2012; Lattimore and Szepesvári 2020). This class of problems studies how to make optimal sequential decisions with new information in different settings, where we aim to maximize our accumulative reward, and we iteratively improve our policy given newly observed results. It finds many important modern applications such as personalized recommendation (Li et al. 2010, 2011), online advertising (Krause and Ong 2011), cost-sensitive classification (Agarwal et al. 2014), and personalized medicine (Goldenshluger and Zeevi 2013; Bastani and Bayati 2015; Tewari and Murphy 2017; Keyvanshokooh et al. 2019). In most contextual bandit problems, at each time period, we first obtain some new information. Then, we take action based on a certain policy and observe a new reward. Our goal is to maximize the total reward, where we iteratively improve our policy. This article is concerned with the linear stochastic bandit model, one of the most fundamental models in contextual bandit problems. We model the expected reward at each time period as a linear function of some random information depending on our action. This model receives considerable attention (Auer 2002; Abe, Biermann, and Long 2003; Dani, Hayes, and Kakade 2008; Rusmevichientong and Tsitsiklis 2010; Chu et al. 2011), and as we will discuss it in more detail, it naturally finds applications in optimal sequential treatment regimes.

In a linear stochastic bandit model, at each time period $t \in \{1, 2, \ldots, T\}$, we are given some action space $\mathcal{A}_t$. Here each feasible action $i \in \mathcal{A}_t$ is associated with a d-dimensional contextual covariate $X_{t,i} \in \mathbb{R}^d$ that is known before any action is taken. Next, we take an action $i_t \in \mathcal{A}_t$ and then observe a reward $Y_t \in \mathbb{R}$. Assume that the reward follows a linear model

$$Y_t = \langle X_{t,i_t}, \theta^* \rangle + \varepsilon_t, \tag{1}$$

where $\theta^* \in \mathbb{R}^d$ is an unknown d-dimensional parameter, and $\{\varepsilon_t\}_{t=1}^T$ are noises. Without loss of generality, we assume that we aim to maximize the total reward. We measure the performance of the sequence of selected actions $\{i_t\}_{t=1}^T$ by the accumulated regret

$$R_T\left(\{i_t\}_{t=1}^T; \theta^*\right) := \sum_{t=1}^T \left(\max_{i \in \mathcal{A}_t} \langle X_{t,i}, \theta^* \rangle - \langle X_{t,i_t}, \theta^* \rangle \right). \tag{2}$$

Essentially, the regret measures the difference between the "best" we can achieve if we know the true parameter θ^* and the "noiseless" reward we get. It is clear that the regret is always nonnegative, and our goal is to minimize it.

Meanwhile, due to the advance in technology, there are many modern applications where we encounter the high-dimensionality issue, that is, the dimension d of the covariate

CONTACT Runze Li ril4@psu.edu Department of Statistics, Pennsylvania State University, University Park, PA 16802-2111.

* These authors contributed equally to this work.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JASA.

is large. Note that here the total number of iterations T is somewhat equivalent to the sample size, which is the number of pieces of information we are able to "learn" the true parameter θ^*. Such a high-dimensionality issue presents in practice. For example, given the genomic information of some patients, we aim to find a policy that assigns each patient to the best treatment for him/her. It is usually the case that the number of patients is much smaller than the covariate dimension. In this article, we consider the linear stochastic contextual bandit problem under such a high-dimensional setting, where the parameter θ^* is of high dimension that $d \; T$. We also assume that θ^* is sparse that only at most s_d components of θ^* are nonzero. This assumption is commonly imposed in high-dimensional statistics and signal processing literature (Donoho 2006; Bühlmann and Van De Geer 2011).

In addition, for the action spaces $\{A_t\}_{t=1}^{T}$, it is known in literature (Dani, Hayes, and Kakade 2008; Shamir 2015; Szepesvári 2016) that if there are infinitely many feasible actions at each time period, the minimax lower bound is of order $O(\sqrt{sdT})$, which does not solve the curse of dimensionality. To simplify notations, throughout the article, we use $O(\cdot)$ to denote the big-O notation that ignores all logarithmic factors. In this work, we assume that the action spaces $\{A_t\}_{t=1}^{T}$ are finite, small, and random. In particular, we assume that for all t, $\|A_t\| = k \; d$, and each action in A_t is associated with a randomly generated contextual covariate. In most practical applications, this finite action space setting is naturally satisfied. For example, in the treatment example, there are usually only a small number of feasible treatments available. We refer the readers to Section 3.1 for a complete description and discussion of the assumptions.

Literature review. In the next, we first briefly review existing works on linear stochastic bandit problems under both low-dimensional and high-dimensional settings. Under the classical low-dimensional setting, Auer (2002) pioneers the use of upper-confidence-bound (UCB) type algorithms for the linear stochastic bandit, which is one of the most powerful and fundamental algorithms for this class of problems, and is also considered in Chu et al. (2011) and Li et al. (2010). Dani, Hayes, and Kakade (2008) and Rusmevichientong and Tsitsiklis (2010) study linear stochastic bandit problems with large or infinite action spaces, and derive corresponding lower bounds. Under the high-dimensional setting, where we assume that θ^* is sparse, when the action spaces A_t's are hyper-cube spaces $[-1, 1]^d$, Lattimore, Crammer, and Szepesvári (2015) develop the SETC algorithm that attains nearly dimension-independent regret bounds. We point out that this algorithm exploits the unique structure of hyper-cubes and is unlikely to be applicable for general action spaces including the ones of our interest where the action spaces are finite. Abbasi-Yadkori, Pál, and Szepesvári (2011) and Abbasi-Yadkori, Pal, and Szepesvari (2012) consider a UCB-type algorithm for general action sets and obtain a regret upper bound of $O(\sqrt{sdT})$, which depends polynomially on the ambient dimension d. Carpentier and Munos (2012) consider a different reward model and obtain an $O(\;^{*}\;{}_2 s\sqrt{T})$ regret upper bound, where $\cdot$ $_2$ denotes the norm of the vector. Goldenshluger and Zeevi (2013) and Bastani and Bayati (2015) study a variant of the linear stochastic bandit problem in which only one contextual covariate X_t is observed at each time period t, while each action i corresponds to a different unknown model θ_i^*. We point out that this model is a special case of our model (1), as discussed in Foster et al. (2018). In addition to above works, there are also some interesting progresses in the linear bandit problem recently. For example, Chen, Lu, and Song (2021) study the inference problem of the linear contextual bandit. Shao et al. (2018) and Medina and Yang (2016) consider models when the payoff is heavy-tailed.

The major challenge of the bandit problem is balancing the tradeoff between exploration and exploitation. One commonly used principle is the optimism-in-the-face-of-uncertainty (Lattimore and Szepesvári 2020), which is also the motivation of UCB-type algorithms (Auer 2002; Chu et al. 2011). Beyond that, several other ideas exist. For example, the -greedy method (Yang and Zhu 2002; Chambaz et al. 2017; Sutton and Barto 2018) takes the random action with probability for the purpose of exploration and takes the greedy action otherwise. The action elimination method (Goldenshluger and Zeevi 2013; Qian and Yang 2016) rules out the suboptimal actions sequentially until the optimal one is found. Stemming from the Bayesian formula, the Thompson sampling method (Agrawal and Goyal 2013; Russo and Van Roy 2016) updates the posterior distribution of potential rewards sequentially and samples action based on the posterior distribution.

Another closely related problem is the online sparse prediction problem (Gerchinovitz 2013; Foster, Kale, and Karloff 2016), in which sequential predictions Y_t's of $Y_t = X_t, \theta^* + \varepsilon_t$ are of the interest, and the regret is measured in mean-squared error $_t |Y_t - Y_t|^2$. It can be further generalized to online empirical-risk minimization (Langford, Li, and Zhang 2009) or even the more general derivative-free/bandit convex optimization (Nemirovsky and Yudin 1983; Flaxman, Kalai, and McMahan 2005; Agarwal, Dekel, and Xiao 2010; Shamir 2013; Besbes, Gur, and Zeevi 2015; Bubeck, Lee, and Eldan 2017; Wang et al. 2017). Most existing works along this direction have continuous (infinite) action spaces $\{A_t\}$. They allow small-perturbation type methods like estimating gradient descent.

From the application perspective of finding the optimal treatment regime, existing literatures focus on achieving the optimality through batch settings. General approaches include model-based methods such as Q-learning (Watkins and Dayan 1992; Murphy 2003; Moodie, Richardson, and Stephens 2007; Chakraborty, Murphy, and Strecher 2010; Goldberg and Kosorok 2012; Song et al. 2015) and A-learning (Robins, Hernan, and Brumback 2000; Murphy 2005) and model-free policy search methods (Robins, Orellana, and Rotnitzky 2008; Orellana, Rotnitzky, and Robins 2010a, 2010b; Zhang et al. 2012; Zhao et al. 2012, 2015). These methods are all developed based on batch settings where we use the whole dataset to estimate the optimal treatment regime. They are applicable after the clinical trial is completed, or when the observational dataset is fully available. However, the batch setting approaches are not applicable when it is emerging to identify the optimal treatment regime. For a contemporary example, during the recent outbreak of coronavirus disease (COVID-19), it is extremely important to quickly identify the optimal or nearly optimal treatment regime to assign each patient the best treatment among a few choices. However, since the disease is novel, there is no or very little historical data. Thus, the batch setting approaches mentioned above are not applicable. On the other hand, our model naturally provides a "learn while optimizing" alternative approach to sequentially improve the policy/treatment regime.

Major Contributions. We summarize our major contributions as follows. In this article, we propose new algorithms, which iteratively learn the parameter θ^* while optimizing the regret. Our algorithms use the "doubling trick" and modern optimization techniques, which carefully balance the randomization for exploration to fully learn the parameter and maximizing the reward to achieve the near-optimal regret. In particular, our algorithms fall under the general UCB-type algorithms (Lai and Robbins 1985; Auer, Cesa-Bianchi, and Fischer 2002). Briefly speaking, we take the action at each period by optimizing some upper confidence bands using the previous estimator. At the end of each period, we renew the estimator using new information. We then enter the next period using the new estimator and renew the estimator at the end of the next period. We repeat this until the Tth time period.

The high-dimensional regime (i.e., $d \gg T$) poses significant challenges in our setting, which cannot be solved by existing works. First, unlike in the low-dimensional regime where ordinary least squares (OLS) always admits closed-form solutions and error bounds, in the high-dimensional regime, most existing methods like the Lasso (Tibshirani 1996) or the Dantzig selector (Candes and Tao 2007) require the sample covariance matrix to satisfy certain "restricted eigenvalue" conditions (Bickel, Ritov, and Tsybakov 2009), which do not hold under our setting for sequentially selected covariates. Additionally, our action spaces $\{A_t\}$ are finite. This rules out several existing algorithms, including the SETC method (Lattimore, Crammer, and Szepesvári 2015) that exploits the specific structure of hyper-cube actions sets and finite-difference type algorithms in stochastic sparse convex optimization (Wang et al. 2017; Balasubramanian and Ghadimi 2018). We adopt the best subset selection estimator (Miller 2002) to derive valid confidence bands only using ill-conditioned sample covariance matrices. Note that while the optimization for best subset selection is NP-hard in theory (Natarajan 1995), by the tremendous progress of modern optimization, solving such problems is practically efficient, as discussed in Pilanci, Wainwright, and El Ghaoui (2015) and Bertsimas, King, and Mazumder (2016). In addition, the renewed estimator may correlate with the previous one. This decreases the efficiency. We let the epoch sizes grow exponentially, which is known as the "doubling trick" (Auer et al. 1995). This "removes" the correlation between recovered support sets by best subset regression. Our theoretical analysis is also motivated by some known analytical frameworks such as the elliptical potential lemma (Abbasi-Yadkori, Pál, and Szepesvári 2011) and the SupLinUCB framework (Auer 2002) in order to obtain sharp regret bounds.

We summarize our main theoretical contribution in the following corollary, which is essentially a simplified version of Theorem 2.

Corollary 1. Under assumptions in Theorem 2, Algorithm 2 achieves a regret

$$R_T(\{i_t\}; \theta^*) = O(\sqrt{sT}).$$

Note that this result holds even if $T \ll d$. Meanwhile, a simpler and more implementable algorithm (Algorithm 1) achieves a weaker regret guarantee of order $O(s\sqrt{T})$ as shown in Theorem 1.

Notations. Throughout this article, for an integer n, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$ to denote the ℓ_1, ℓ_2 and ℓ_∞ norms of vector, respectively. Given a matrix $\mathbf{A}$, we use $\|\cdot\|_{\mathbf{A}}$ to denote the ℓ_2 norm weighted by $\mathbf{A}$. Specifically, we have $\|X\|_{\mathbf{A}} := \sqrt{X^\top \mathbf{A} X}$. We also use $\langle \cdot, \cdot \rangle$ to denote the inner product of two vectors. Given a set $S \subseteq [d]$, S^c is its complement and $|S|$ denotes the cardinality. Given a d-dimensional vector X, we use $[X]_i$ to denote its ith coordinate. We also use supp(X) to represent the support of X, which is the collection of indices corresponding to nonzero coordinates. Furthermore, we use $[X]_S = ([X]_i)_{i\in S}$ to denote the restriction of X on S, which is a $|S|$-dimensional vector. Similarly, for a $d \times d$ matrix $\mathbf{A} = ([\mathbf{A}]_{ij})_{i,j\in[d]} \in \mathbb{R}^{d\times d}$, we denote by $[\mathbf{A}]_S = ([\mathbf{A}]_{jk})_{j\in S,k\in S}$ the restriction of A on $S \times S$, which is a $|S| \times |S|$ matrix. When $S = S$, we further abbreviate $[\mathbf{A}]_S = [\mathbf{A}]_{SS}$. For real numbers a and b, let $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$. In addition, given two sequences of nonnegative real numbers $\{a_n\}_{n\geq 1}$ and $\{b_n\}_{n\geq 1}$, $a_n \lesssim b_n$ and $a_n \gtrsim b_n$ mean that there exists an absolute constant $0 < C < \infty$ such that $a_n \leq Cb_n$ and $a_n \geq Cb_n$ for all n, respectively. We also abbreviate $a_n \asymp b_n$, if $a_n \lesssim b_n$ and $a_n \gtrsim b_n$ hold simultaneously. We say that a random event E holds with probability at least $1 - \delta$, if there exists some absolute constant C such that the probability of E is larger than $1 - C\delta$. Finally, we remark that arm, action, and treatment all refer to actions in different applications. We also denote by i_t the action taken in period t and $X_t = X_{t,i_t}$ the associated covariate.

2. Methodologies

In this section, we present the proposed methods to solve the linear stochastic bandit problem where we aim to minimize the regret defined in (2). In Section 2.1, we first introduce an algorithm called "Sparse-LinUCB" (SLUCB), as summarized in Algorithm 1, which can be efficiently implemented and demonstrate the core idea of our algorithmic design. The SLUCB algorithm is a variant of the celebrated LinUCB algorithm (Chu et al. 2011) for classical linear contextual bandit problems. The SLUCB algorithm is intuitive and easy to implement. However, we cannot derive the optimal upper bound for the regret due to technical reasons. To close this gap, we further propose a more sophisticated algorithm called "Sparse-SupLinUCB" (SSUCB) (Algorithm 2) in Section 3.2. In comparison with the SLUCB algorithm, the SSUCB algorithm constructs the upper confidence bands through sequentially selected historical data and achieves the optimal regret (up to logarithmic factors).

2.1. Sparse-LinUCB Algorithm

As we mentioned above, our algorithm is inspired by the classic LinUCB algorithm, which balances the tradeoff between exploration and exploitation following the principle of "optimism in the face of uncertainty." In particular, the LinUCB algorithm repeatedly constructs upper confidence bands for the potential rewards of the actions. The upper confidence bands are optimistic estimators. We then pick the action associated with the largest upper confidence band. This leads to the optimal regret under the low-dimensional setting. However, under the high-dimensional setting, directly applying the LinUCB algorithm incurs some suboptimal regret since we only get loose confidence bands under the high-dimensional regime. Thus, it is desirable to construct tight confidence bands under the high-dimensional and sparse setting to achieve the optimal regret.

Inspired by the remarkable success of the best subset selection (BSS) in high-dimensional regression problems, we propose incorporating this powerful tool into the LinUCB algorithm. Meanwhile, since the BSS procedure is computationally expensive, it is impractical and unnecessary to execute the BSS method during every time period. In contrast, we first partition the whole decision periods into several consecutive epochs and only execute the BSS method at the end of each epoch. Then within each epoch, restricting on the selected dimensions, we calculate the upper confidence bands of each potential reward and pick the arm with the largest upper confidence band.

Before we present the details of our algorithm, we briefly discuss the support of parameter. Given a d-dimensional vector θ, we denote by supp(θ) the support set of θ, which is the collection of dimensions of θ with nonzero coordinates that

$$\text{supp}(\theta) = \ j \in [d] : [\theta]_j = 0 \ .$$

This definition agrees with that of most literature. However, for the BSS procedure, it is desirable to generalize this definition. We propose the concept of "generalized support" as follows.

Definition 1 (Generalized Support). Given a d-dimensional vector θ, we call a subset $S \subseteq [d]$ the generalized support of θ and denote it by supp$^+(\theta)$, if

$$[\theta]_j = 0, \ \forall j \in S.$$

The generalized support supp$^+(\theta)$ is a relaxation of the normal support, since any support is a generalized support (but not vice versa). Moreover, the generalized support is not unique. Any subset including the support is a valid generalized support.

We distinguish the difference between support and generalized support in order to define the best subset selection without causing confusion. For example, we consider a linear model $\theta^* \in \mathrm{R}^d$, $X_t \in \mathrm{R}^d$, and $Y_t = X_t, \theta^* + \varepsilon_t$. Calculating the ordinary least square estimator restricted on the generalized support $S \in [d]$, which is denoted by

$$\theta = \operatorname{argmin}_{\text{supp}^+(\theta)=S} \ _t \ Y_t - X_t, \theta \ ^2 ,$$

means that we consider a low-dimensional model only using the information in S and set the coordinates of estimator except in S as zeros. Formally, let $[\theta]_S \in \mathrm{R}^{|S|}$ be

$$[\theta]_S = \operatorname{argmin}_{\varphi \in \mathrm{R}^{|S|}} \ _t \ Y_t - [X_t]_S, \varphi \ ^2 .$$

Then we have

$$\theta_j = \ [\theta]_{S\,j}, \ j \in S; \ \theta_j = 0, \ j \in S.$$

Since we do not guarantee $[\theta]_j = 0, \ \forall j \in S$, we call S the generalized support instead of support.

We are ready to present the details of the SLUCB algorithm now. Our algorithm works as follows. We first apply the "doubling trick," which partitions the whole T decision periods into several consecutive epochs such that the lengths of the epochs increase doubly. We only implement the BSS procedure at the end of each epoch to recover the support of the parameter θ^*. Within each epoch, we fix the support of size s recovered from the previous epoch, and treat the problem as an s-dimensional regression problem. Specifically, at each time period, we use the ridge estimator with penalty weight λ to estimate θ^* in the selected dimensions and construct corresponding confidence bands to help us make decisions in the next time period.

In summary, we partition the time horizon $[T]$ into consecutive epochs $\{E_\tau\}_{\tau=1}^{\mathcal{T}}$ such that

$$[T] = \ _{\tau=1}^{\mathcal{T}} E_\tau, \ |E_\tau| = 2^\tau .$$

Without loss of generality, we assume that the last epoch is of length $2^{\mathcal{T}}$ exactly. By definition, the number of epochs $\mathcal{T}$ log(T). Hence, in the SLUCB algorithm, we run the BSS procedure at most $O($log$(T))$ times. In our later simulation studies, we find that this is practical for moderately large dimensions.

Next, we introduce the details of constructing upper confidence bands in the SLUCB algorithm. We assume that at period $t \in E_\tau$, we pick action $i_t \in A_t$ and observe the associated covariate X_{t,i_t} and reward Y_t. We also abbreviate X_{t,i_t} as X_t, if there is no confusion. We denote by $\theta_{\tau-1,\lambda}$ the BSS estimator for the true parameter θ^* at the end of previous epoch $E_{\tau-1}$ and let

$$S_{\tau-1} = \text{supp}^+(\theta_{\tau-1,\lambda})$$

be its generalized support, that is, the generalized support recovered by epoch $E_{\tau-1}$. For period $t \in E_\tau$, we estimate θ^* by a ridge estimator. Let $\theta_{\tau,\lambda}^{t-1}$ be the most recently updated ridge estimator of θ^* by $t \in E_\tau$, which is estimated by restricting its support on $S_{\tau-1}$ using data $\{X_t, Y_t\}_{t \in E_\tau^{t-1}}$, where $E_\tau^{t-1} = \{t \in E_\tau : t \leq t-1\}$. In particular, all components of $\theta_{\tau,\lambda}^{t-1}$ outside $S_{\tau-1}$ are set as zeros and

$$\theta_{\tau,\lambda}^{t-1} = \operatorname{argmin}_{\text{supp}^+(\theta)=S_{\tau-1}} \ _{t \in E_\tau^{t-1}} \ Y_t - X_t, \theta \ ^2 + \lambda \ \ ^2 \theta.$$

Given $\theta_{\tau,\lambda}^{t-1}$, we calculate the upper confidence band of potential reward $X_{t,i}, \theta^*$ for each possible action $i \in A_t$. In particular, we introduce two tuning parameters α and β that correspond to the confidence level and an upper estimate of the potential reward, respectively. The recommended choices of α and β will be discussed in Section 3. Then for each $i \in A_t$, we calculate the upper confidence band associated with action i as

$$\min \ \beta, \ X_{t,i}, \theta_{\tau,\lambda}^{t-1} \ + \alpha \cdot \ \sigma \ \overline{\log(kTd/\delta) \ |E_{\tau-1}|}$$
$$+ \ \overline{[X_{t,i}]_{S_{\tau-1}} \ \ _{\tau-1,\lambda}^{t-1} \ ^{-1} [X_{t,i}]_{S_{\tau-1}}} \ ,$$

where

$$\ _{\tau-1,\lambda}^{t-1} = \lambda \mathbf{I}_{|S_{\tau-1}|} + \ _{t \in E_\tau^{t-1}} \ [X_t]_{S_{\tau-1}} [X_t]_{S_{\tau-1}},$$

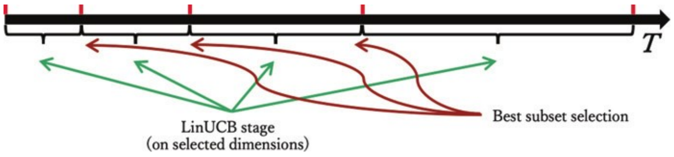

Figure 1. An illustration of SLUCB Algorithm.

and $\delta \in (0, 1)$ is a tuning parameter of confidence level. After that, we pick the arm i_t corresponding to the largest upper confidence band to play and observe the corresponding reward $Y_t = X_{t,i_t}, \theta^* + \varepsilon_t$. We repeat this process until the end of epoch E_τ.

Then we run the BSS procedure using all data collected in E_τ to recover the support of θ^*. We also enlarge the size of generalized support by s. To be specific, let S_τ be the generalized support recovered in this step. We require that S_τ satisfies constraints

$$S_\tau \supseteq S_{\tau-1}, \ |S_\tau| \le \tau s.$$

and obtain the BSS estimator $\theta_{\tau,\lambda}$ as

$$\theta_{\tau,\lambda} = \operatorname{argmin}_{S_{\tau-1} \subseteq \operatorname{supp}^+(\theta), |\operatorname{supp}^+(\theta)| \le \tau s, \quad \theta \quad \le r} \quad \underset{t \in E_\tau}{} \quad Y_t - X_t, \theta^{\ 2} + \lambda \quad {}^2\theta. \qquad (3)$$

Note that in comparison with the standard BSS estimator, we further restrict the $_2$ norm to be bounded. The boundedness also simplifies our later theoretical analysis. We also add the inclusion restriction $S_\tau \supseteq S_{\tau-1}$ for technical convenience, which does not lead to any fundamental difference. As a result, we need to consider the sparsity τs instead of s. It boosts the probability of recovering the true support. See Figure 1 for an illustration of the SLUCB algorithm. A pseudo-code description of the SLUCB algorithm is also presented in Algorithm 1. In addition, we briefly discuss how to compute the BSS estimator in Section S.3 in the supplementary materials.

2.2. Sparse-SupLinUCB Algorithm

Although the SLUCB algorithm is intuitive and easy to implement, we are unable to prove the optimal upper bound for its regret due to some technical reasons. Specifically, as discussed in the next section, we can only establish an $\overline{O}(\sqrt{s}$ upper bound for the regret of the SLUCB algorithm, while the optimal regret should be $O(\sqrt{sT})$. Here we omit all the constants and logarithmic factors and only consider the dependency on horizon length and dimension parameters. The obstacle leading to suboptimality is the dependency of covariates on random noises. Recall that in each period $t \in E_\tau$, the SLUCB algorithm constructs the ridge estimator $\theta^{t-1}_{\tau,\lambda}$ using all historical data, where the designs $\{X_t\}_{t \in E^{t-1}_\tau}$ are correlated with noises $\{\varepsilon_t\}_{t \in E^{t-1}_\tau}$ due to the UCB-type policy. Such a complicated correlation impedes us from establishing tight confidence bands for predicted rewards, which results in a suboptimal regret.

To close the aforementioned gap and achieve the optimality, we modify the seminal SupLinUCB algorithm (Auer 2002; Chu et al. 2011), which is originally proposed to attain the optimal regret for classic stochastic linear bandit problems, as a subroutine in our framework. Then we propose the Sparse-SupLinUCB (SSUCB) algorithm. Specifically, we replace the ridge estimator and UCB-type policy in the SLUCB algorithm with a modified SupLinUCB algorithm. The basic idea of the SupLinUCB algorithm is to separate the dependent designs into several groups such that within each group, the designs and noises are independent of each other. Then the ridge estimators of the true parameters are calculated based on group individually. Thanks to the desired independency, now we can derive tighter confidence bands by applying sharper concentration inequality, which gives rise to the optimal regret in the final.

In the next, we present the details of SupLinUCB algorithm and show how to embed it in our framework. For each period $t \in E_\tau$, the SupLinUCB algorithm partitions the historical periods E^{t-1}_τ into ζ disjoint groups

$$E^{t-1}_\tau = \{ \ {}^{t-1,1}_{\tau}, \ldots, {}^{t-1,\zeta}_{\tau} \},$$

where $\zeta = \log(\beta T)$ and same as before, *beta* is an upper estimate of the potential reward. We initialize these groups as empty sets, and we update them sequentially as follows. For each period t, we screen the groups $\{^{t-1,\zeta}\}$ one by one (in an ascending order of index ζ) to determine which action to take or eliminate some obvious suboptimal actions.

```
Input: sequentially arriving covariates $\{X_{t,i}\}_{t\in[T], i\in A_t}$, confidence level
       $\alpha$, estimated upper bound of reward $\beta$, sparsity level $s$, ridge
       regression penalty $\lambda$.
Output: action sequence $\{i_t\}_{t\in[T]}$.
partition $[T]$ into consecutive epochs $E_1, E_2, \ldots, E_\tau$ such that $|E_\tau| = 2^\tau$;
initialization: $\theta_{0,\lambda} = 0$, $S = \emptyset$;
for $\tau = 1, 2, \ldots, \tau$ do
    for time periods $t \in E_\tau$ do
    end
    calculate matrix
        ${}^{t-1}_{\tau-1,\lambda} = \lambda \mathbf{I}_{|S|} + \underset{t \in E^{t-1}_\tau}{} [X_t]_S [X_t]_S$;
    calculate the upper confidence band of reward for each arm
        $r(X_{t,i}) = \min \ \beta, \ X_{t,i}, \theta^{t-1}_{\tau,\lambda} + \alpha \cdot \ \sigma \ \overline{\log(kTd/\delta) \ |E_{\tau-1}|}$
            $+ \ [X_{t,i}]_S \ [\ {}^{t-1}_{\tau-1,\lambda}]^{-1}$ ;
    select arm with the largest upper confidence band
        $i_t = \operatorname{argmin}_{i \in A_t} \ r(X_{t,i})$ ;
    observe reward
        $Y_t = X_{t,i_t}, \theta^* + \varepsilon_t$;
    update the ridge estimator:
        $\theta^t_{\tau,\lambda} = \operatorname{argmin}_{\operatorname{supp}^+(\theta)=S} \ \underset{t \in E^t_\tau}{} \ Y_t - X_t, \theta^{\ 2} + \lambda \quad {}^2 ; \theta$
    } update the best subset selection estimator
        $\theta_{\tau,\lambda} = \operatorname{argmin}_{S \subseteq \operatorname{supp}^+(\theta), |\operatorname{supp}^+(\theta)| \le \tau s, \quad \theta \quad \le r}$
            $\underset{t \in E_\tau}{} \ Y_t - X_t, \theta^{\ 2} + \lambda \quad {}^2 ; \theta$
    update $S = \operatorname{supp}^+(\theta_{\tau,\lambda})$;
end
```

Algorithm 1: Sparse-LinUCB Algorithm.

```
Input: epoch index $\tau$, sequential arriving covariates $\{X_{t,i}\}_{t\in E_\tau, i\in A_t}$,
       confidence level $\gamma$, estimated upper bound of reward $\beta$, support
       recovered in previous epoch $S_{\tau-1}$, sparsity level $s$, ridge
       regression penalty $\lambda$.
Output: action sequence $\{i_t\}_{t\in E_\tau}$.
set $\zeta = \log(\beta T)$, $S = S_{\tau-1}$, and initialize sets $\{{}_{\tau}^{t,1}, \ldots, {}_{\tau}^{t,\zeta}\}$ as
  empty;
for time periods $t$ in $E_\tau$ do
    initialize $\zeta = 1$, $N_\tau^{t-1,\zeta} = A_t$;
    repeat
        compute restricted ridge estimator
            $\theta_{\tau,\lambda}^{t-1,\zeta} = \mathrm{argmin}_{\mathrm{supp}^+(\theta)=S} \sum_{t\in {}_{\tau}^{t-1,\zeta}} Y_t - X_t, \theta^{2} + \lambda^{2}$; $\theta$
        compute matrix
            ${}_{\tau-1,\lambda}^{t-1,\zeta} = \lambda \mathbf{I}_{|S|} + \sum_{t\in {}_{\tau}^{t-1,\zeta}} [X_t]_S [X_t]_S$;
        compute confidence band for each $i \in N_\tau^{t-1,\zeta}$,
            $\omega_{\tau,\lambda}^{t-1,\zeta}(i) = \gamma \cdot \overline{s/|E_{\tau-1}|} + [X_{t,i}]_S\,[{}_{\tau-1,\lambda}^{t-1,\zeta}]^{-1}$;
        if $\omega_{\tau,\lambda}^{t-1,\zeta}(i) \le 1/\sqrt{T}$, $\forall i \in N_\tau^{t-1,\zeta}$ then
            select
                $i_t = \mathrm{argmin}_{i\in N_\tau^{t-1,\zeta}} \beta, X_{t,i}, \theta_\tau^{t-1,\zeta} + \omega_{\tau,\lambda}^{t-1,\zeta}(i)$;
            as the arm to play and update ${}_{\tau}^{t,\zeta} \leftarrow {}_{\tau}^{t-1,\zeta}$ for all $\zeta \in [\zeta]$;
        end
        else if $\omega_{\tau,\lambda}^{t-1,\zeta}(i) \le 2^{-\zeta}\beta$, $\forall i \in N_\tau^{t-1,\zeta}$ then
            eliminate suboptimal arms as
                $N_\tau^{t-1,\zeta+1} = i \in N_\tau^{t-1,\zeta} : X_{t,i}, \theta_{\tau,\lambda}^{t-1,\zeta}$
                    $\ge \max_{j\in N_\tau^{t-1,\zeta}} X_{t,j}, \theta_{\tau,\lambda}^{t-1,\zeta} - 2^{1-\zeta}\beta$;
            move to the next group and update $\zeta \leftarrow \zeta + 1$;
        end
        else
            select $i_t \in N_\tau^{t-1,\zeta}$ such that $\omega_{\tau,\lambda}^{t-1,\zeta}(i) > 2^{-\zeta}\beta$ as the arm to
            play;
            update ${}_{\tau}^{t,\zeta} \leftarrow {}_{\tau}^{t-1,\zeta} \cup \{t\}$ and ${}_{\tau}^{t,\zeta} \leftarrow {}_{\tau}^{t-1,\zeta}$ for all
            $\zeta = \zeta$;
        end
    until an arm $i_t \in A_t$ is selected;
end
```

Algorithm 2: Sparse-SupLinUCB Subroutine.

Suppose that we are at the ζ th group now. Let $N_\tau^{t-1,\zeta}$ be the set of candidate actions that are still kept by the ζ th step, which is initialized as the whole action space A_t when $\zeta = 1$. We first calculate the ridge estimator $\theta_{\tau,\lambda}^{t-1,\zeta}$ restricted on the generalized support $S_{\tau-1}$, using data from group ${}_{\tau}^{t-1,\zeta}$. Then for each action $i \in N_\tau^{t-1,\zeta}$, we calculate $\omega_{\tau,\lambda}^{t-1,\zeta}(i)$, the width of confidence band of the potential reward. Specifically, we have

$$\theta_{\tau,\lambda}^{t-1,\zeta} = \mathrm{argmin}_{\mathrm{supp}^+(\theta)=S_{\tau-1}} \sum_{t\in {}_{\tau}^{t-1,\zeta}} Y_t - X_{t,i_t}, \theta^{2} + \lambda^{2}_{\theta}$$

$$\omega_{\tau,\lambda}^{t-1,\zeta}(i) = \gamma \cdot \sqrt{s/|E_{\tau-1}|} + \overline{[X_{t,i}]_{S_{\tau-1}} [{}_{\tau-1,\lambda}^{t-1,\zeta}]^{-1} [X_{t,i}]_{S_{\tau-1}}},$$

where γ is a tuning parameter of confidence level. A recommended choice of γ will be discussed in Section 3 as well. Our next step depends on the values of $\omega_{\tau,\lambda}^{t-1,\zeta}(i)$. If

$$\omega_{\tau,\lambda}^{t-1,\zeta}(i) \le 1/\sqrt{T}, \ \forall i \in N_\tau^{t-1,\zeta},$$

which means that the widths of confidence bands are uniformly small, we pick the action associated with the largest upper confidence band

$$\min \ \beta, X_{t,i}, \theta_{\tau,\lambda}^{t-1,\zeta} + \omega_{\tau,\lambda}^{t-1,\zeta}(i) \ .$$

In this case, we discard the newly observed data point (Y_t, X_t) and do not update any group, that is, setting ${}_{\tau}^{t,\zeta} = {}_{\tau}^{t-1,\zeta}$, for all $\zeta \in [\zeta]$.

Otherwise, if there exists some $i \in N_\tau^{t-1,\zeta}$ such that $\omega_{\tau,\lambda}^{t-1,\zeta}(i) \ge 2^{-\zeta}\beta$, which means that the width of confidence band is not sufficiently small, then we pick such an action i to play for exploration. In this case, we add the period t into the ζ th group while keeping all other groups unchanged, that is,

$${}_{\tau}^{t,\zeta} = {}_{\tau}^{t-1,\zeta} \cup \{t\}, \quad {}_{\tau}^{t,\eta} = {}_{\tau}^{t-1,\eta}, \text{ if } \eta = \zeta .$$

Finally, if neither one of the above scenarios happens, which implies that for all $i \in N_\tau^{t-1,\zeta}$ $\omega_{\tau,\lambda}^{t-1,\zeta}(i) \le 2^{-\zeta}\beta$, then we do not take any action for now. Instead, we eliminate some obvious suboptimal actions and move to the next group ${}_{\tau}^{t-1,\zeta+1}$. Particularly, we update the set of candidate arms as

$$N_\tau^{t-1,\zeta+1} = i \in N_\tau^{t-1,\zeta} : X_{t,i}, \theta_{\tau,\lambda}^{t-1,\zeta}$$
$$\ge \max_{j\in N_\tau^{t-1,\zeta}} X_{t,j}, \theta_{\tau,\lambda}^{t-1,\zeta} - 2^{1-\zeta}\beta \ .$$

We repeat the above procedure until an arm is selected. Since the number of groups is $\zeta = \log(\beta T)$ and $2^{-\zeta}\beta = 1/T \le 1/\sqrt{T}$, the SupLinUCB algorithm stops eventually.

By replacing the direct ridge regression and UCB-type policy with the SupLinUCB algorithm above, we obtain the SSUCB algorithm. The pseudo-code is presented in Algorithm 2.

3. Theoretical Results

In this section, we present the theoretical results of the SLUCB and SSUCB algorithms. We use the regret to evaluate the performance of our algorithms, which is a standard performance measure in literature. We denote by $\{i_t\}_{t\in[T]}$ the actions sequence generated by an algorithm. Then given the true parameter θ^* and covariates $\{X_{t,i}\}_{t\in[T], i\in A_t}$, recall that the regret of the sequence $\{i_t\}_{t\in[T]}$ is defined in (2), where $i_t^* = \mathrm{argmax}_{i\in A_t} X_{t,i}, \theta^*$ denotes the optimal action under the true parameter. The regret measures the discrepancy in accumulated reward between real actions and oracles where the true parameter is known to a decision-maker. In what follows, in Section 3.1, we first introduce some technical assumptions to facilitate our discussions. Then we study the regrets of the SLUCB and SSUCB algorithms in Sections 3.2 and 3.3, respectively.

3.1. Assumptions

We present the assumptions in our theoretical analysis and discuss their relevance and implications. To simplify, we consider finite action spaces $\{A_t\}_{t=1}^T$ and assume that there exists some constant k such that

$$|A_t| = k, \ \forall t \in [T].$$

We also assume that for each period t, the covariates $\{X_{t,i}\}_{i \in A_t}$ are sampled independently from an unknown distribution P_0. We further impose the following assumptions on distribution P_0.

Assumption 1. Let random vector $X \in \mathbb{R}^d$ follow the distribution P_0. Then X satisfies:

(A1) *(Sub-Gaussianity)*: Random vector $X \in \mathbb{R}^d$ is centered and sub-Gaussian with variance proxy σ^2, that is, $\mathbb{E}[X] = 0$ and

$$\mathbb{E}\big[\exp\{\sigma a^\top X\}\big] \le \exp\big(\sigma^2\|a\|_2^2/2\big), \ \forall a \in \mathbb{R}^d;$$

(A2) *(Non-degeneracy)*: There exists a constant $\rho \in (0, \sigma^2]$ such that

$$\mathbb{E}\big[[X]_j^2\big] \ge \rho, \ \forall j \in [d];$$

(A3) *(Independent coordinates)*: The d coordinates of X are independent of each other.

We briefly discuss Assumption 1. First of all, (A1) is a standard assumption in literature, with sub-Gaussianity covering a broad family of distributions like Gaussian, Rademacher, and bounded distributions. Assumption (A2) is a non-degeneracy assumption which, together with (A3), implies that the smallest eigenvalue of the population covariance matrix $\mathbb{E}[XX^\top]$ is lower bounded by some constant $\rho > 0$. Similar assumptions are also adopted in high-dimensional statistics literature in order to prove the "restricted eigenvalue" conditions of sample covariance matrices (Raskutti, Wainwright, and Yu 2010), which are essential in the analysis of penalized least square methods (Wainwright 2009; Bickel, Ritov, and Tsybakov 2009). However, we emphasize that in our setting, the covariates indexed by the selected actions $\{i_t\}$ do not guarantee the restricted eigenvalue condition in general, and therefore, we need novel and non-standard analysis of the high-dimensional M-estimators. For Assumption (A3), at a higher level, independence among coordinates enables relatively independent explorations in different dimensions, which is similar to the key idea of the SETC method (Lattimore, Crammer, and Szepesvári 2015). Technically, (A3) is used to establish the key independence of sample covariance matrices restricted within and outside the recovered support. Due to such independence, the rewards in the unexplored directions at each period are independent as well, which can be estimated efficiently. In addition, we discuss more details of the technical reason why we need (A3) and some relaxations in the supplementary materials.

We next impose the following assumptions on the unknown d-dimensional true parameter θ^*.

Assumption 2.

(B1) *(Sparsity)*: The true parameter θ^* is sparse. In other words, there exists an $s \ll d$ such that $|\text{supp}(\theta^*)| = s$.

(B2) *(Boundedness)*: There exists a constant $r > 0$ such that $\|\theta^*\|_2 \le r$.

Note that in Assumption 2, (B1) is the key sparsity assumption, which assumes that only $s \ll d$ components of the true parameter θ^* are nonzero. Assumption (B2) is a boundedness condition on the ℓ_2-norm of θ^*. This assumption is often imposed, either explicitly or implicitly, in contextual bandit problems for deriving an upper bound for rewards (Dani, Hayes, and Kakade 2008; Chu et al. 2011).

Finally, we impose the sub-Gaussian assumption on noises sequence $\{\epsilon_t\}_{t=1}^T$, which is a standard assumption adopted in most statistics and bandit literature.

Assumption 3.

(C1) *(Sub-Gaussian noise)*: The random noises $\{\epsilon_t\}_{t=1}^T$ are independent, centered, and sub-Gaussian with variance proxy ν^2.

3.2. Regret Analysis of Sparse-LinUCB

In this section, we analyze the performance of the SLUCB algorithm. As discussed earlier, we measure the performance via the regret defined in (2). We show that with a tailored choice of tuning parameters α and β, the accumulated regret of the SLUCB algorithm is upper bounded by $O(\sqrt{s}T)$ (up to logarithmic factors) with high probability. Formally, we have the following theorem.

Theorem 1. For any $\delta \in (0, 1)$, let

$$\alpha = \big((\sigma r + \nu) \vee 1\big) \cdot \Big(\sqrt{s \log\big(kTd/(\delta\lambda)\big)} + \sqrt{\lambda} r\Big),$$
$$\beta = r\sigma \log(kTd/\delta).$$

Under Assumptions 1–3, the regret of the actions sequence $\{i_t\}_{t=1}^T$ generated by the Sparse-LinUCB algorithm is upper bounded by

$$R_T\big(\{i_t\}, \theta^*\big) \lesssim \big((\sigma r + \nu) \vee 1\big) \cdot \sigma\rho^{-1/2} \log(T) \log^2(kTd/\delta)$$
$$\cdot \Big(\sqrt{sT} + \log(kTd/\delta)\Big) \sqrt{\log(T) \log\big(1 + \sigma \sqrt{d} \log(kTd/\delta)/\lambda\big)}$$
$$\cdot s\sqrt{T}$$

with probability at least $1 - \delta$.

Note that in Theorem 1, if we omit all the constants and logarithmic factors, the dominating part in the accumulated regret is of order $O(s\sqrt{T})$. Moreover, the regret upper bound contains two terms. The first term, $O(\sqrt{sT})$, is incurred by selection bias of best subset regression. The dominating term $O(s\sqrt{T})$ is the regret incurred by the UCB-type selection policy. In the SSUCB algorithm, we improve this part through the SupLinUCB algorithm and finally achieves the $O(\sqrt{s}T)$ regret. We point out that since the our regret upper bounds depend on the action space size k through a polylogarithmic function, we can ignore the regret's dependence on k when $k \ll \exp(T)$. The analysis of Theorem 1 builds upon a nontrivial combination of the UCB-type algorithm and the best subset selection method. The proof of Theorem 1 is provided in Section S.1 of the supplementary materials.

3.3. Regret Analysis of Sparse-SupLinUCB

In comparison with the SLUCB algorithm, the SSUCB algorithm splits the historical data into several groups dynamically. In each period, we sequentially update the ridge estimator and corresponding confidence bands using data from a single group instead of the whole data. The motivation of only using a single group of data is to achieve the independence between the design matrix and random noises within each group, which leads to tighter confidence bands by applying a sharper concentration inequality. The tighter upper confidence bands of the predicted rewards lead to an improved regret. In particular, we have the following theorem.

Theorem 2. For any $\delta \in (0, 1)$, let

$$\beta = r\sigma \log(kTd/\delta),$$
$$\gamma = r \cdot (\sigma \vee 1)(\rho \wedge 1)^{-1/2} \left(\sqrt{\lambda \vee 1} + \nu + \sigma\right) \log^2 \left(kTd/((\lambda \wedge 1)\delta)\right).$$

Then under Assumptions 1–3, the regret of actions sequence $\{i_t\}_{t=1}^T$ generated by the Sparse-SupLinUCB algorithm is upper bounded by

$$R_T\left(\{i_t\}, \theta^*\right) \quad r \cdot (\sigma \vee 1)(\rho \wedge 1)^{-1/2} \left(\sqrt{\lambda \vee 1} + \nu + \sigma\right) \cdot \log^3 \left(kTd/((\lambda \wedge 1)\delta)\right) \cdot \sqrt{sT},$$

with probability at least $1 - \delta$.

Note that in Theorem 2, if we omit all constants and logarithmic factors, the dominating part in the regret upper bound is of order $O(\sqrt{sT})$. This improves the rate in Theorem 1 by an order of $O(\sqrt{s})$ and achieves the optimal rate (up to logarithmic factors). Theorem 2 builds on a tailored analysis of the SupLinUCB algorithm. The proof of Theorem 2 is given in Section S.2 of the supplementary materials.

4. Numerical Experiments

In this section, we use extensive numerical experiments to investigate our algorithm's empirical performances. Here we focus on the SLUCB algorithm since it is easy to implement. Theoretically, we are only to prove a suboptimal regret upper bound for the SLUCB algorithm due to technical reasons. However, extensive numerical experiments imply that it already performs very well in practice. We further implement the SSUCB algorithm as well and compare its empirical performance with the SLUCB algorithm.

4.1. Simulation Studies

We first show the $O(\sqrt{T})$ growth rate of regret empirically. Then we fix time horizon length T and dimension d and study the dependency of accumulated regret on sparsity s. To demonstrate the power of best subset selection, we also compare our algorithm's performance with the oracle, where the decision-maker knows the true support of underlying parameters. Since the bottleneck of computing time in our algorithm is the best subset selection, which requires solving a mixed-integer programming problem, it is appealing to replace this step with other variable selection methods, such as Lasso and iterative hard thresholding (IHT) (Blumensath and Davies 2009). We test the performance of those variants. Throughout the simulation, all the covariates $X_{t,i}$'s are drawn from a d-dimensional multivariate Gaussian distribution with identity covariance matrix independently.

4.1.1. Experiment 1: Growth of Regret

In this experiment, we study the growth rate of regret. We run two sets of experiments, where in the first case d = 100, T = 310, s = 5, 10, 15, and in the second case, d = 300, T = 620, s = 5, 10, 15. For each setup, we replicate 20 times and then calculate corresponding mean and 90%-confidence interval. We present the results in Figure 2. For each fixed d and s, the growth rate of regret is about $O(\sqrt{T})$, which validates our theory. Note that in Figure 2, when T is comparable to d, we observe the $\sqrt{T}$-shaped growth. When T is larger, the growth of regret further slows down.

We also consider a scenario where the horizon length T is much smaller than the dimension d. Specifically, we set d = 500, s= 5, T = 200 and d = 500, s = 15, T = 300. Furthermore, to demonstrate the necessity of best subset selection, we implement the vanilla LinUCB algorithm as a benchmark and compare the corresponding regret with our algorithm's. The results are presented in Figure 3. In this case, our algorithm also achieves superior performance that is much better than the vanilla LinUCB algorithm.

4.1.2. Experiment 2: Dependency on Sparsity

In this experiment, we fix the dimension d and horizon length T, and let sparsity s change. We calculate the accumulated regret at the end of horizon. We also run two sets of experiments, where in the first case d = 100, T = 310, s = 4, 6, 8, . . . , 20 and in the second case d = 620, T = 1970, s = 4, 6, 8, . . . , 20. We present the results in Figure 4. Although Theorem 1 only provides an $O(s\sqrt{T})$ regret guarantee for the SLUCB algorithm. The linear dependency of accumulated regret on $\sqrt{s}$ suggests that it actually attains the optimal $O(\sqrt{sT})$ rate in practice.

4.1.3. Experiment 3: Performance of SSUCB Algorithm

In this experiment, we implement the SSUCB algorithm, which achieves the nearly optimal regret in theory, and compare its performance with the SLUCB algorithm. We consider an instance where s= 5, d = 100, T = 300 and an instance where s = 15, d = 300, T = 500. The results are presented in Figure 5. As we can see, the regret curves of both algorithms admit approximate $O(\sqrt{T})$-growth rates. Moreover, in the beginning, the SLUCB algorithm performs slightly better than the SSUCB algorithm. This is not surprising since the SSUCB algorithm uses a more sophisticated mechanism to guarantee the independence of the design matrix with the random noises. As a result, certain efficiency is sacrificed when the sample size is small. The constant in regret upper bound may also not be tight. However, as decision periods length increases, the SSUCB algorithm eventually achieves a smaller final regret than the SLUCB algorithm, which demonstrates its optimality in theory. Note that in both scenarios, the discrepancy between the two algorithms is quite small.

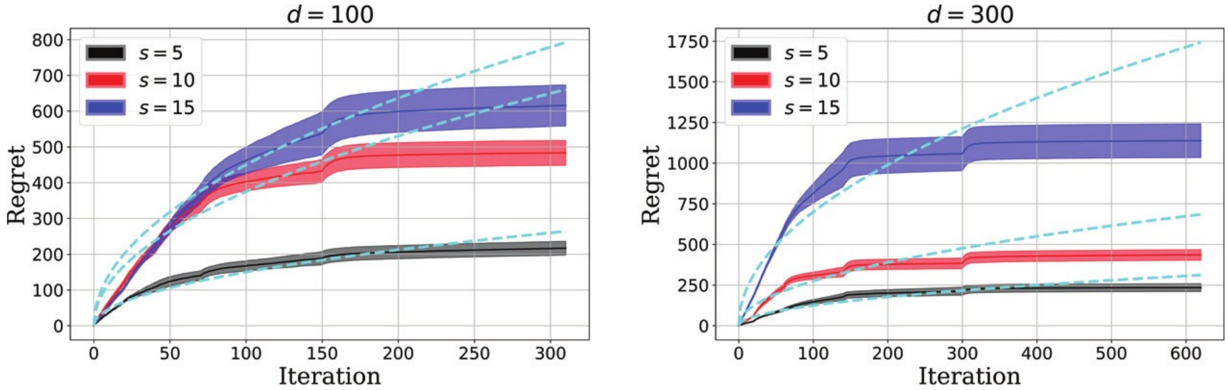

Figure 2. Plot of regret versus time periods. In (a), we set the dimension $d = 100$, the horizon length $T = 310$, and the sparsity $s = 5, 10, 15$. In (b), we set the dimension $d = 300$, the horizon length $T = 620$, and the sparsity $s = 5, 10, 15$. For each setting, we replicate 20 times. Solid lines are the means of regret. Shadow areas denote corresponding empirical confidence intervals.

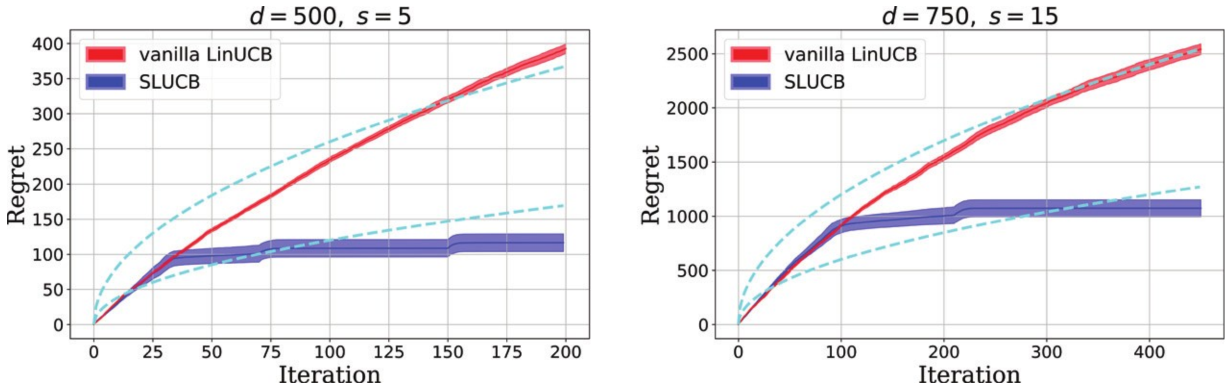

Figure 3. Plot of regret versus time periods for vanilla LinUCB algorithm and SLUCB algorithm when T d. In (a), we set the dimension $d = 500$, the horizon length $T = 200$, and the sparsity $s = 5$. In (b), we set the dimension $d = 750$, the horizon length $T = 450$, and the sparsity $s = 15$. For each setting, we replicate 20 times. Solid lines are the means of regret. Shadow areas denote corresponding empirical confidence intervals.

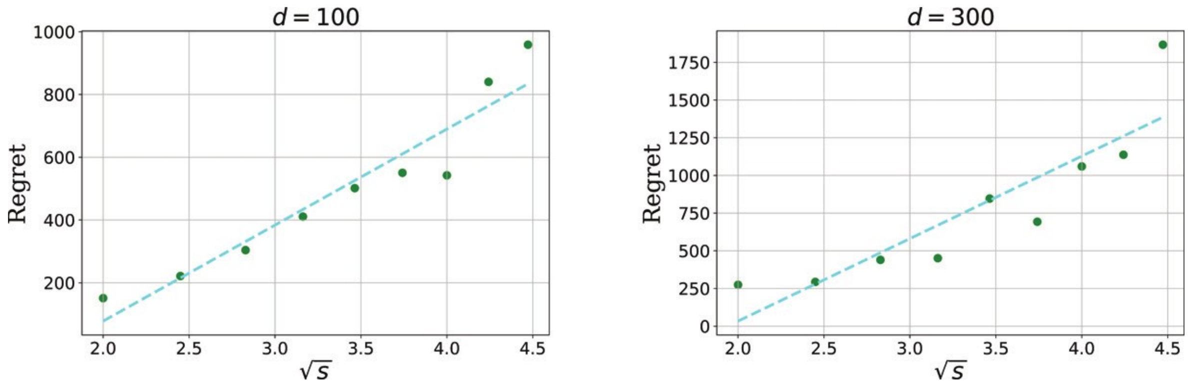

Figure 4. Plot of accumulated regret versus $\sqrt{s}$. In (a), we set the dimension $d = 100$, the horizon length $T = 1300$, and the sparsity $s = 4, 6, 8, \cdots, 20$. In (b), we set the dimension $d = 300$, the horizon length $T = 1970$, and the sparsity $s = 4, 6, 8, \cdots, 20$.

4.1.4. Experiment 4: Comparison with Variants of Main Algorithm and Oracle

In this experiment, we compare the performance and computing time of the SLUCB algorithm with several variants that substitute the best subset selection procedure with Lasso and IHT. We also compare with the oracle regret where the decision-maker knows the true support of parameter. In more detail, for the first variant, we use Lasso to recover the support at the end of each epoch. We tune the $_1$-penalty parameter λ such that the size of the support of the estimator is approximately s, and then use it in the next epoch. For the second variant, we apply IHT to estimate the parameter and set the sparsity level as s.

We run two settings of experiments, corresponding to $d = 100$, $s = 15$, $T = 300$, and $d = 300$, $s = 15$, $T = 300$. We also replicate 20 times in each setting. For the computing time, in the first setting, the average computing times are 32 sec for Lasso, 34 sec for IHT, and 4.3 min for best subset selection. For the second case, the average computing times are 35 sec for Lasso, 32 sec for IHT, and 10.9 min for best subset selection. We display the associated regret curves in Figure 6. We observe that the performance of IHT is significantly weaker than the

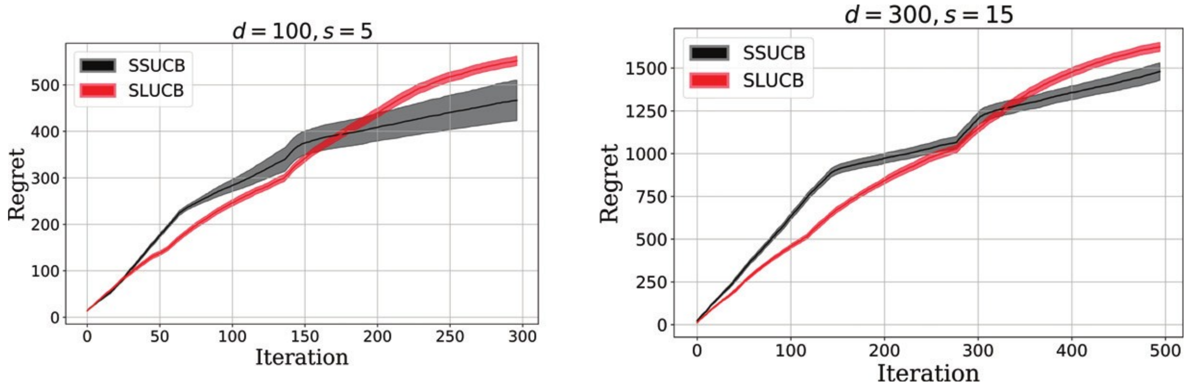

Figure 5. Plot of regret versus time periods for SLUCB algorithm and SSUCB algorithm. In (a), we set the dimension $d = 100$, $s = 5$ and the horizon length $T = 300$. In (b), we set the dimension $d = 300$, $s = 15$, the horizon length $T = 500$. For each setting, we replicate 20 times. Solid lines are the means of regret. Shadow areas denote corresponding empirical confidence intervals.

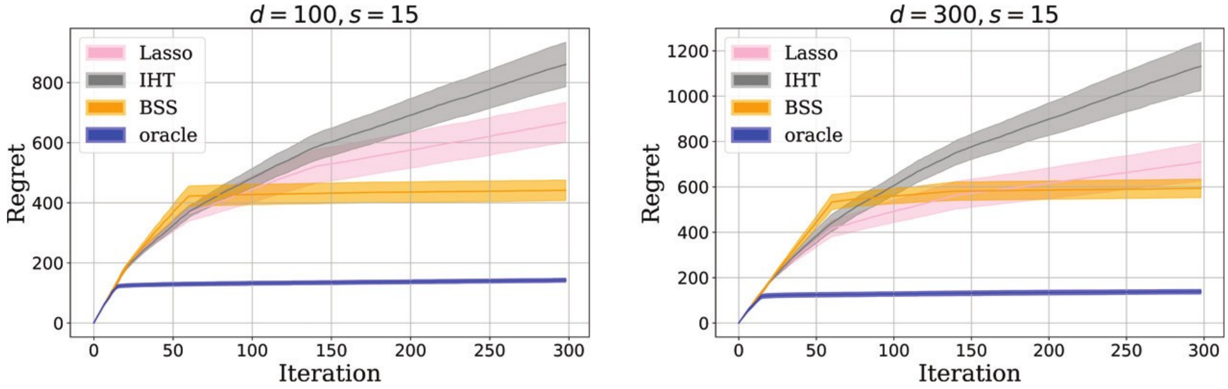

Figure 6. Plot of regret curves of different algorithms. In (a), we set $d = 100$, $s = 15$, and $T = 300$. In (b), we set $d = 300$, $s = 15$, and $T = 300$. We test four variants: Lasso, IHT, BSS, and oracle. We also replicate 20 times in each setting. Solid lines are means of regret. Shadow areas denote corresponding confidence intervals.

other methods. Meanwhile, the computing time of Lasso is much shorter than best subset selection, but it achieves the similar performance, which suggests that Lasso might be a good alternative in practice when the computing resource is limited. Finally, although the computing time of the best subset selection is the longest, it achieves the best performance.

4.2. Real Data Application: Warfarin Dosing Problem

In this section, we use a real data application to demonstrate the usefulness of our model and methodology. Nowadays, the practitioners can use specific individual-level information, combined with advanced data analytics tools, to sequentially determine the optimal clinical treatment for each patient. One example is optimal warfarin dosing. As the most widely used oral anticoagulant agent worldwide, more than 30 million prescriptions were written for warfarin in the United States in 2004. An appropriate dosage is critical but difficult for practitioners to establish, since it can vary by up to 10% among individuals, depending on various factors. The consequences of an incorrect dosage can be catastrophic, which may lead to severe adverse effects such as stroke or internal bleeding. In recent years, abundant medical research has been devoted to determining the optimal using information like demographic, clinical, and genomic factors. However, most of these researches are offline, given that all the data are ready-to-use. In this application, we consider such a problem in an online manner where the data points are collected sequentially through clinical trials. A similar formulation is studied in Bastani and Bayati (2015). We refer the interested readers to Bastani and Bayati (2015) for more details about the setup as well. However, we remark that the algorithm in Bastani and Bayati (2015) relies on a prescribed forced-sampling mechanism. It means that a specific treatment is forced to apply at some fixed periods regardless of the information by then, even if we have enough evidence that such a treatment is inappropriate. Such an enforced sampling scheme may raise ethical concerns in medical applications. In contrast, our algorithm always applies the UCB-type selection, which balances the exploration and ethics in a more delicate way.

In terms of the dataset, we use a publicly available dataset collected by staff at the Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB). It records the true patient-specific optimal warfarin doses, as well as the corresponding patient-level covariates like demographic variables, clinical factors, and genetic information, for 5528 patients who were treated with warfarin from 21 research groups spanning nine countries and four continents. The dimension of the covariates is $d = 93$. Details and a list of names of all the covariates can be found in the supplementary materials of International Warfarin Pharmacogenetics Consortium (2009).

Note that in a natural formulation of the optimal treatment selection problem, we observe a patient-specific covariate $X_t \in \mathrm{R}^d$, and each treatment $i \in [N]$ is associated with a parameter $\theta_i \in \mathrm{R}^d$. The goal is to find the optimal

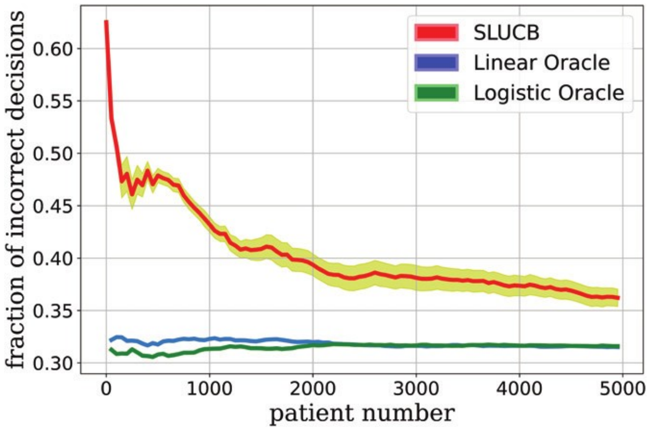

Figure 7. Performance of SLUCB algorithm in optimal warfarin dosing problem.

treatment i_t^* that maximizes the expected treatment effect X_t, θ. However, it is straightforward to translate such a formulation to the model considered of this article. We only need to consider the augmented covariates and parameter, $X_{t,i} = (0, \ldots, 0, X_t, 0, \ldots, 0) \in \mathbb{R}^{N\times d}$ (the ith component is X_t and all others are 0) and $\theta^* = (\theta_1, \theta_2, \ldots, \theta_N) \in \mathbb{R}^{N\times d}$. Then we have $X_{t,i}, \theta^* = X_t, \theta$ for any $i \in [N]$. For performance measure, since the outcome of our dataset is categorical, we modify the accumulated regret as the fraction of incorrect decisions, that is, the misclassification rate so far, which is more relevant to the practice.

We use the SLUCB algorithm to solve the optimal warfarin dosing problem. In this application, since the response is categorical and some covariates are binary, some distributional assumptions imposed in our theoretical analysis may not hold. However, our algorithm still achieves reasonable empirical performance. For comparison, we consider the two offline oracles, where the parameter of each treatment is first estimated using the whole dataset through linear or logistic regression, and then the treatment is selected as the optimal one. Moreover, we simulate 20 random permutations of all the patients. We plot the mean misclassification rate curve and corresponding 95% confidence bands (for the offline oracles, we only present the mean curves for clarity). The results are summarized in Figure 7. As we see, the misclassification rate drops rapidly at the beginning, even if there is only a small batch of data available. When the number of patients increases, the performance of our algorithms continues to improve and approaches the offline oracles very well eventually.

5. Conclusion and Discussion

In this article, we first propose a method for the high-dimensional stochastic linear bandit problem by combining the best subset selection method and the LinUCB algorithm. It achieves the $O(s\sqrt{T})$ regret upper bound and is nearly independent of the ambient dimension d (up to logarithmic factors). In order to attain the optimal regret $O(\sqrt{sT})$, we further improve our method by modifying the SupLinUCB algorithm. Extensive numerical experiments validate the performance and robustness of our algorithms. Moreover, although we cannot prove the $O(\sqrt{sT})$ regret upper bound for the SLUCB algorithm due to some technical reasons, simulation studies show that the regret of the SLUCB algorithm is actually $O(\sqrt{sT})$ rather than our provable upper bound $O(s\sqrt{T})$. A similar phenomenon is also observed in the seminal works (Auer 2002; Chu et al. 2011), where low-dimensional stochastic linear bandit problems are investigated.

There are several future directions worth exploring. First, it remains an open problem whether the SLUCB algorithm achieves the optimal $O(\sqrt{sT})$ upper bound. Note that even in the low-dimensional setting where d T, it is unclear how to show the optimal $O(\sqrt{dT})$ upper bound for LinUCB algorithm. Second, it is interesting to study the high-dimensional sparse linear bandit problem under weaker distributional assumptions, especially when the independent coordinates assumption does not hold. Moreover, in this work, we assume that the random noises in feedbacks are sub-Gaussian. It is also worth considering the heavy-tailed cases, which bring new challenges to balance the exploration-exploitation tradeoff. In this work, we study a linear model with sparsity constraint here. It is appealing to extend to the generalized linear model. It is also important in theory and application to consider other types of constraints, such as convex or shape constraints. How to combine statistical tools with bandit algorithms in these settings remains nontrivial and interesting. More importantly, it is interesting to further investigate the tradeoff between computational cost and optimality of regret in future research. Although the SLUCB algorithm achieves superior performance than the LinUCB algorithm, it needs much longer computational time due to the best subset selection, which is NP-hard. However, many practical applications of bandits, especially ads recommendation and A/B testing, emphasize the quick response time. Hence, it is worthwhile considering a combinatorial search method to tackle NP-hard problems in bandit problems (Streeter 2007; Kotthoff 2016).

Finally, it is worth mentioning that although this work focuses on the bandit model only, there are recently popular reinforcement learning applications in the precision medicine field (Coronato et al. 2020). In a contextual bandit model, the system's status is fixed while in a general reinforcement learning setting, the status may change along the time, which is more challenging (Sutton and Barto 2018). It is also important and interesting to extend our algorithms to the reinforcement learning setting when the data is high-dimensional.

Supplementary Materials

Supplementary Materials contain technical lemmas, some detailed proofs, and some additional numerical results.

Acknowledgments

The authors thank the Editor, Associate Editor, and two anonymous reviewers for their insightful suggestions, which lead to a significant improvement of this article.

Disclosure Statement

The authors report there are no competing interests to declare.

Funding

Fang was partially supported by a Whitehead scholarship, NSF DMS-2230795, DMS-2230797. Zhaoran Wang was partially supported by NSF (Awards 2048075, 2008827, 2015568, 1934931), Simons Institute (Theory of Reinforcement Learning), Amazon, J.P. Morgan, and Two Sigma. Li's research was supported by NSF DMS 1820702, DMS 1953196, and DMS 2015539.

ORCID

Runze Li http://orcid.org/0000-0002-0154-2202

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011), "Improved Algorithms for Linear Stochastic Bandits," i *Advances in Neural Information Processing Systems*, 2312–2320. [247, 248]

——— (2012), "Online-to-Confidence-Set Conversions and Application to Sparse Stochastic Bandits," in *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*. [247]

Abe, N., Biermann, A. W., and Long, P. M. (2003), "Reinforcement Learning with Immediate Rewards and Linear Hypotheses," *Algorithmica*, 37, 263–293. [246]

Agarwal, A., Dekel, O., and Xiao, L. (2010), "Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback," in *Proceedings of Annual Conference on Learning Theory (COLT)*, Citeseer. [247]

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. (2014), "Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits," in *Proceedings of International Conference on Machine Learning (ICML)*. [246]

Agrawal, S., and Goyal, N. (2013), "Thompson Sampling for Contextual Bandits with Linear Payoffs," in *International Conference on Machine Learning*, pp. 127–135. PMLR. [247]

Auer, P. (2002), "Using Confidence Bounds for Exploitation-Exploration Trade-offs," *Journal of Machine Learning Research*, 3, 397–422. [246, 247, 248, 250, 256]

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (1995), "Gambling in a Rigged Casino: The Adversarial Multi-Armed Bandit Problem," in *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*. [248]

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002), "Finite-Time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, 47, 235–256. [248]

Balasubramanian, K., and Ghadimi, S. (2018), "Zeroth-Order (non)-convex Stochastic Optimization via Conditional Gradient and Gradient Updates," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. [248]

Bastani, H., and Bayati, M. (2015), "Online Decision-Making with High-Dimensional Covariates," Available at *https://ssrn.com/abstract=2661896*. [246, 247, 255]

Bertsimas, D., King, A., and Mazumder, R. (2016), "Best Subset Selection via a Modern Optimization Lens," *The Annals of Statistics*, 44, 813–852. [248]

Besbes, O., Gur, Y., and Zeevi, A. (2015), "Non-Stationary Stochastic Optimization," *Operations Research*, 63, 1227–1244. [247]

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2009), "Simultaneous Analysis of Lasso and Dantzig Selector," *The Annals of Statistics*, 37, 1705–1732. [248, 252]

Blumensath, T., and Davies, M. E. (2009), "Iterative Hard Thresholding for Cmpressed Sensing," *Applied and Computational Harmonic Analysis*, 27, 265–274. [253]

Bubeck, S., and Cesa-Bianchi, N. (2012), "Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems," arXiv preprint arXiv:1204.5721. [246]

Bubeck, S., Lee, Y. T., and Eldan, R. (2017), "Kernel-based Methods for Bandit Convex Optimization," in *Proceedings of the Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. [247]

Bühlmann, P., and Van De Geer, S. (2011), *Statistics for High-Dimensional Data: Methods, Theory and Applications*, Berlin: Springer. [247]

Candes, E., and Tao, T. (2007), "The Dantzig Selector: Statistical Estimation When p is much Larger than n," *The Annals of Statistics*, 35, 2313–2351. [248]

Carpentier, A., and Munos, R. (2012), "Bandit Theory Meets Compressed Sensing for High Dimensional Stochastic Linear Bandit," in *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*. [247]

Chakraborty, B., Murphy, S., and Strecher, V. (2010), "Inference for Non-regular Parameters in Optimal Dynamic Treatment Regimes," *Statistical Methods in Medical Research*, 19, 317–343. [247]

Chambaz, A., Zheng, W., and van der Laan, M. J. (2017), "Targeted Sequential Design for Targeted Learning Inference of the Optimal Treatment Rule and its Mean Reward," *Annals of Statistics*, 45, 2537–2564. [247]

Chen, H., Lu, W., and Song, R. (2021), "Statistical Inference for Online Decision Making: In a Contextual Bandit Setting," *Journal of the American Statistical Association*, 116, 240–258. [247]

Chu, W., Li, L., Reyzin, L., and Schapire, R. (2011), "Contextual Bandits with Linear Payoff Functions," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. [246, 247, 248, 250, 252, 256]

Coronato, A., Naeem, M., De Pietro, G., and Paragliola, G. (2020), "Reinforcement Learning for Intelligent Healthcare Applications: A Survey," *Artificial Intelligence in Medicine*, 109, 101964. [256]

Dani, V., Hayes, T. P., and Kakade, S. M. (2008), "Stochastic Linear Optimization under Bandit Feedback," in *Proceedings of Annual Conference on Learning Theory (COLT)*. [246, 247, 252]

Donoho, D. L. (2006), "Compressed Sensing," *IEEE Transactions on Information Theory*, 52, 1289–1306. [247]

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. (2005), "Online Convex Optimization in the Bandit Setting: Gradient Descent without a Gradient," in *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. [247]

Foster, D., Kale, S., and Karloff, H. (2016), "Online Sparse Linear Regression," in *Proceedings of annual Conference on Learning Theory (COLT)*. [247]

Foster, D. J., Agarwal, A., Dudík, M., Luo, H., and Schapire, R. E. (2018), "Practical Contextual Bandits with Regression Oracles," in *Proceedings of the International Conference on Machine Learning (ICML)*. [247]

Gerchinovitz, S. (2013), "Sparsity Regret Bounds for Individual Sequences in Online Linear Regression," *Journal of Machine Learning Research*, 14, 729–769. [247]

Goldberg, Y., and Kosorok, M. R. (2012), "Q-learning with Censored Data," *Annals of Statistics*, 40, 529–560. [247]

Goldenshluger, A., and Zeevi, A. (2013), "A Linear Response Bandit Problem," *Stochastic Systems*, 3, 230–261. [246, 247]

International Warfarin Pharmacogenetics Consortium (2009), "Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data," *New England Journal of Medicine*, 360, 753–764. [255]

Keyvanshokooh, E., Zhalechian, M., Shi, C., Van Oyen, M. P., and Kazemian, P. (2019), "Contextual Learning with Online Convex Optimization: Theory and Application to Chronic Diseases," Available at SSRN. [246]

Kotthoff, L. (2016), "Algorithm Selection for Combinatorial Search Problems: A Survey," in *Data Mining and Constraint Programming*, eds. C. Bessiere, L. De Raedt, L. Kotthoff, S. Nijssen, B. O'Sullivan, and D. Pedreschi, pp. 149–190, Cham: Springer. [256]

Krause, A., and Ong, C. S. (2011), "Contextual Gaussian Process Bandit Optimization," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. [246]

Lai, T. L., and Robbins, H. (1985), "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Mathematics*, 6, 4–22. [248]

Langford, J., Li, L., and Zhang, T. (2009), "Sparse Online Learning via Truncated Gradient," *Journal of Machine Learning Research*, 10, 777–801. [247]

Lattimore, T., and Szepesvári, C. (2020), *Bandit Algorithms*, Cambridge, MA: Cambridge University Press. [246, 247]

Lattimore, T., Crammer, K., and Szepesvári, C. (2015), "Linear Multi-Resource Allocation with Semi-Bandit Feedback," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. [247, 248, 252]

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010), "A Contextual-Bandit Approach to Personalized News Article Recommendation," in *Proceedings of the International Conference on World Wide Web (WWW)*, ACM. [246,247]

Li, L., Chu, W., Langford, J., and Wang, X. (2011), "Unbiased Offline Evaluation of Contextual-Bandit-based News Article Recommendation Algorithms," in *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, ACM. [246]

Medina, A. M., and Yang, S. (2016), "No-Regret Algorithms for Heavy-Tailed Linear Bandits," in *International Conference on Machine Learning*, pp. 1642–1650. PMLR. [247]

Miller, A. (2002), *Subset Selection in Regression*, Boca Raton, FL: Chapman and Hall/CRC. [248]

Moodie, E. E., Richardson, T. S., and Stephens, D. A. (2007), "Demystifying Optimal Dynamic Treatment Regimes," *Biometrics*, 63, 447–455. [247]

Murphy, S. A. (2003), "Optimal Dynamic Treatment Regimes," *Journal of the Royal Statistical Society*, Series B, 65, 331–355. [247]

——— (2005), "An Experimental Design for the Development of Adaptive Treatment Strategies," *Statistics in Medicine*, 24, 1455–1481. [247]

Natarajan, B. K. (1995), "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, 24, 227–234. [248]

Nemirovsky, A. S., and Yudin, D. B. (1983), *Problem Complexity and Method Efficiency in Optimization*, Philadelphia: SIAM. [247]

Orellana, L., Rotnitzky, A., and Robins, J. M. (2010a), "Dynamic Regime Marginal Structural Mean Models for Estimation of Optimal Dynamic Treatment Regimes, Part I: Main Content," *The International Journal of Biostatistics*, 6, 1–49. DOI: 10.2202/1557-4679.1200. [247]

——— (2010b), "Dynamic Regime Marginal Structural Mean Models for Estimation of Optimal Dynamic Treatment Regimes, Part II: Proofs of Results," *The International Journal of Biostatistics*, 6, 1–19. DOI: 10.2202/1557-4679.1242. [247]

Pilanci, M., Wainwright, M. J., and El Ghaoui, L. (2015), "Sparse Learning via Boolean Relaxations," *Mathematical Programming*, Series B, 151, 63–87. [248]

Qian, W., and Yang, Y. (2016), "Randomized Allocation with Arm Elimination in a Bandit Problem with Covariates," *Electronic Journal of Statistics*, 10, 242–270. [247]

Raskutti, G., Wainwright, M. J., and Yu, B. (2010), "Restricted Eigenvalue Properties for Correlated Gaussian Designs," *Journal of Machine Learning Research*, 11, 2241–2259. [252]

Robins, J., Orellana, L., and Rotnitzky, A. (2008), "Estimation and Extrapolation of Optimal Treatment and Testing Strategies," *Statistics in Medicine*, 27, 4678–4721. [247]

Robins, J. M., Hernan, M. A., and Brumback, B. (2000), "Marginal Structural Models and Causal Inference in Epidemiology," *Epidemiology*, 11, 550–560 [247]

Rusmevichientong, P., and Tsitsiklis, J. N. (2010), "Linearly Parameterized Bandits," *Mathematics of Operations Research*, 35, 395–411. [246,247]

Russo, D., and Van Roy, B. (2016), "An Information-Theoretic Analysis of Thompson Sampling," *The Journal of Machine Learning Research*, 17, 2442–2471 [247]

Shamir, O. (2013), "On the Complexity of Bandit and Derivative-Free Stochastic Convex Optimization," in *Proceedings of annual Conference on Learning Theory (COLT)*. [247]

——— (2015), "On the Complexity of Bandit Linear Optimization," in *Proceedings of Annual Conference on Learning Theory (COLT)*. [247]

Shao, H., Yu, X., King, I., and Lyu, M. R. (2018), "Almost Optimal Algorithms for Linear Stochastic Bandits with Heavy-Tailed Payoffs," arXiv preprint arXiv:1810.10895 [247]

Song, R., Wang, W., Zeng, D., and Kosorok, M. R. (2015), "Penalized q-learning for Dynamic Treatment Regimens," *Statistica Sinica*, 25, 901–920. [247]

Streeter, M. (2007), "Using Online Algorithms to Solve NP-Hard Problems More Efficiently in Practice," PhD thesis, Carnegie Mellon University. [256]

Sutton, R. S., and Barto, A. G. (2018), *Reinforcement Learning: An Introduction*, Cambridge, MA: MIT Press. [247,256]

Szepesvari, C. (2016), "Lower Bounds for Stochastic Linear Bandits," *http://banditalgs.com/2016/10/20/lower-bounds-for-stochastic-linear-bandits/*. Accessed October 15 2018. [247]

Tewari, A., and Murphy, S. A. (2017), "From Ads to Interventions: Contextual Bandits in Mobile Health," in *Mobile Health*, eds. J. M. Rehg, S. A. Murphy, and S. Kumar, pp. 495–517, Cham: Springer. [246]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [248]

Wainwright, M. J. (2009), "Sharp Thresholds for High-Dimensional and Noisy Sparsity Recovery using ℓ_1-Constrained Quadratic Programming (Lasso)," *IEEE Transactions on Information Theory*, 55, 2183–2202. [252]

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. (2017), "Stochastic Zeroth-Order Optimization in High Dimensions," in *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*. [247,248]

Watkins, C. J., and Dayan, P. (1992), "Q-Learning," *Machine Learning*, 8, 279–292. [247]

Yang, Y., and Zhu, D. (2002), "Randomized Allocation with Nonparametric Estimation for a Multi-Armed Bandit Problem with Covariates," *The Annals of Statistics*, 30, 100–121. [247]

Zhang, B., Tsiatis, A. A., Laber, E. B., and Davidian, M. (2012), "A Robust Method for Estimating Optimal Treatment Regimes," *Biometrics*, 68, 1010–1018 [247]

Zhao, Y., Zeng, D., Rush, A. J., and Kosorok, M. R. (2012), "Estimating Individualized Treatment Rules using Outcome Weighted Learning," *Journal of the American Statistical Association*, 107, 1106–1118. [247]

Zhao, Y.-Q., Zeng, D., Laber, E. B., Song, R., Yuan, M., and Kosorok, M. R. (2015), "Doubly Robust Learning for Estimating Individualized Treatment with Censored Data," *Biometrika*, 102, 151–168. [247]