Efficient and Robust High-Dimensional Linear Contextual Bandits

Cheng Chen[1] , **Luo Luo**[2] , **Weinan Zhang**[1] , **Yong Yu**[1] and **Yijiang Lian**[3]
[1]Shanghai Jiao Tong University
[2]Hong Kong University of Science and Technology
[3]Baidu
jack_chen1990@sjtu.edu.cn, luoluo@ust.hk, {wnzhang, yyu}@apex.sjtu.edu.cn, lianyijiang@baidu.com

Abstract

The linear contextual bandits is a sequential decision-making problem where an agent decides among sequential actions given their corresponding contexts. Since large-scale data sets become more and more common, we study the linear contextual bandits in high-dimensional situations. Recent works focus on employing matrix sketching methods to accelerating contextual bandits. However, the matrix approximation error will bring additional terms to the regret bound. In this paper we first propose a novel matrix sketching method which is called Spectral Compensation Frequent Directions (SCFD). Then we propose an efficient approach for contextual bandits by adopting SCFD to approximate the covariance matrices. By maintaining and manipulating sketched matrices, our method only needs $O(md)$ space and $O(md)$ update time in each round, where d is the dimensionality of the data and m is the sketching size. Theoretical analysis reveals that our method has better regret bounds than previous methods in high-dimensional cases. Experimental results demonstrate the effectiveness of our algorithm and verify our theoretical guarantees.

1 Introduction

The contextual bandit is a popular model for sequential decision-making problems [Auer, 2002; Dani *et al.*, 2008; Chu *et al.*, 2011; Abbasi-Yadkori *et al.*, 2011]. In contextual bandit problems, a decision maker repeatedly (a) receives a set of contexts (or actions), (b) chooses an action, and (c) observes a reward for the chosen action. The goal is to maximize the expected cumulative reward over some time horizon. Recently, contextual bandits have been applied to many machine learning tasks, including personalized recommendation [Li *et al.*, 2010], web advertising [Tang *et al.*, 2013], social network analysis [Zhao *et al.*, 2014] and mobile health platforms [Tewari and Murphy, 2017].

Contextual bandits have been widely studied in recent years [Langford and Zhang, 2008; Tang *et al.*, 2013]. The traditional methods for linear contextual bandits including upper-confidence bound algorithms [Chu *et al.*, 2011; Abbasi-Yadkori *et al.*, 2011] and Thompson sampling algorithms [Agrawal and Goyal, 2013; Russo and Van Roy, 2014]. Some other works extends linear bandits to generalized linear models [Filippi *et al.*, 2010; Li *et al.*, 2017; Dumitrascu *et al.*, 2018]. A famous algorithm for linear contextual bandits is `LinUCB` [Chu *et al.*, 2011], which regret bound is $O(\sqrt{dT\log^3(KT\log(T)/\delta)})$. Here d is the dimension of the contextual vectors and T is the time horizon. This result was improved to $O(d\sqrt{T}\log(T) + \sqrt{dT\log(T)\log(1/\delta)})$ [Abbasi-Yadkori *et al.*, 2011], which is irrelevant to the number of arms K.

In the big data era, the dimension of the data grows rapidly. In high-dimensional situation, i.e., $d \approx T$, most traditional bandit algorithms are time-consuming. Also, the upper regret bounds become larger with the growth of the dimension. Some recent works focus on high dimensional linear contextual bandits. `SLUCB` [Carpentier and Munos, 2012] achieves a regret bound $O(S\sqrt{T})$ by assuming that the contextual data contains only S non-zero components. `BallEXP` [Deshpande and Montanari, 2012] leverages the technique of ball exploration in the high-dimensional space, but their regret bound is very loose. These two methods are still less efficient in practice because they require $O(d^2)$ time in each round. Calandriello *et al.* [2019] proposed an efficient Gaussian process based algorithm for kernel bandits. But their regret bound is $\tilde{O}(d\sqrt{T})$ for linear kernels.

Recently, some researchers apply matrix sketching techniques to contextual bandit problems [Yu *et al.*, 2017; Kuzborskij *et al.*, 2019]. In [Yu *et al.*, 2017], random projection was adopted to map the high-dimensional contextual information to a random m-dimensional subspace and proposed Contextual Bandits with RAndom Projection (`CBRAP`) algorithm. Though `CBRAP` successfully reduces the update time from $O(d^2)$ to $O(md + m^3)$, the random projection introduces an additive error εT to their regret bound. To enable the failure probability δ away from 1, the parameter ε has to be $\Omega(m^{-1/2})$. Kuzborskij *et al.* [2019] propose `SOFUL` algorithm by adopting Frequent Directions (FD) to sketch covariance matrices of linear contextual bandits. Since FD [Liberty, 2013; Ghashami *et al.*, 2016] has good theoretical guarantees in streaming setting, it is more suitable for contextual bandits than random projection. Due to the advantage of FD, `SOFUL` only require $O(md)$ update time and achieve upper regret bound as $\tilde{O}((1+\Delta_T)^{3/2}(m+d\log(1+\Delta_T))\sqrt{T})$, where Δ_T is upper bounded by the spectral tail of the covariance matrix.

However, FD still has some drawbacks when applied to contextual bandits. First, the sequence of covariance matrices satisfies the positive definite monotonicity, i.e. $\mathbf{V}_t \succeq \mathbf{V}_{t-1}$, where $\mathbf{V}_t$ is the covariance matrix in the t-th round. But FD produces a rank-deficient approximation which destroys the positive definite monotonicity of covariance matrices. This deficiency leads to additional terms in the regret bounds of contextual bandits. Specifically, the regret bound of [Kuzborskij *et al.*, 2019] is associated with the spectral tail to the power of $3/2$. However, the sketch size is usually much smaller than the rank of covariance matrix in practice. Thus their bounds may be heavily affected by the spectral tail.

In this paper, we present a variant of Frequent Directions, which we call Spectral Compensation Frequent Directions (SCFD). To overcome the disadvantages of FD, SCFD compensates a diagonal matrix which contains spectral information to the result of FD. We further adopt SCFD to accelerate LinUCB, and propose an efficient algorithm for high-dimensional linear contextual bandits, which is called Contextual Bandits via Spectral Compensation Frequent Directions (CBSCFD). Unlike the regularization term used in [Kuzborskij *et al.*, 2019], which is invariant during the sequential decisions, the compensated diagonal matrix of SCFD changes adaptively in each round. Compare with previous methods for contextual bandits, our method provide better regret bounds in high-dimensional cases. In addition, our algorithm is much less sensitive to the regularization hyper-parameter, which means that our method is robust.

The main contributions of this paper are summarized as follows:

- We propose SCFD algorithm for matrix sketching. SCFD can approximate a sequence of incremental covariance matrices while keeping the positive definite monotonicity.
- We propose CBSCFD for high-dimensional linear contextual bandits. Our methods only requires $O(md)$ time and $O(md)$ space to update the model in each round. Also, our methods obtain a regret bound of $\tilde{O}((\sqrt{m+d\log(1+\Delta_T)}+\sqrt{\Delta_T})\sqrt{mT})$, which is better than state-of-the-art methods. Here Δ_T is upper bounded by the spectral tail (sum of the last $d-m+1$ singular values) of the covariance matrix.
- We validate our approach on both synthetic and real-world data sets. The result of experiments shows that our algorithm outperforms other state-of-the-art algorithms in the high-dimensional setting.

2 Preliminaries

In this section, we first show notations used in this paper. Then we describe the problem setting of linear contextual bandits. Finally we introduce recent works for Frequent Directions.

2.1 Notation and Definition

Let $\mathcal{X}$ denote the context space and $\mathcal{A} = \{1, \ldots, K\}$ denote the action space. Suppose the total rounds of playing bandits is T. We use $\tilde{O}$ notation to hide $\log(T)$ in the complexity analysis. We use $\mathbf{I}_m$ to denote $m\times m$ identity matrix, and $\mathbf{0}$ to denote a zero vector or matrix of appropriate size. For a vector $\mathbf{x} \in \mathbb{R}^d$, let $\|\mathbf{x}\|_2$ be the ℓ_2-norm of $\mathbf{x}$. For a positive semi-definite (PSD) matrix $\mathbf{A}$, the weighted 2-norm of vector $\mathbf{x}$ is defined by $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^\top \mathbf{A}\mathbf{x}}$. We denote the inner product as $\langle\cdot,\cdot\rangle$ and the weighted inner product as $\mathbf{x}\mathbf{A}^\top\mathbf{y} = \langle\mathbf{x},\mathbf{y}\rangle_{\mathbf{A}}$. Let $\det(\mathbf{A})$ be the determinant of $\mathbf{A}$. For two PSD matrices $\mathbf{A}$ and $\mathbf{B}$, we use $\mathbf{A} \succeq \mathbf{B}$ to represent the fact that $\mathbf{A}-\mathbf{B}$ is PSD.

Let ρ be the rank of matrix $\mathbf{A}$. The reduced SVD of $\mathbf{A}$ is defined as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \sum_{i=1}^{\rho} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$, where σ_i are positive singular values in the descending order. That is, $\sigma_i(\mathbf{A})$ is the i-th largest singular value of $\mathbf{A}$. Let $\kappa(\mathbf{A}) = \sigma_{\max}(\mathbf{A})/\sigma_{\min}(\mathbf{A})$ be the condition number of $\mathbf{A}$. Let $(\mathbf{A})_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ denote the best rank-k approximation of $\mathbf{A}$. Additionally, we let $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$ be the Frobenius norm and $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})$ be the spectral norm.

We define the positive definite monotonicity as follows:

Definition 1. *For a sequence of positive semi-definite matrices $\{\mathbf{A}_i\}_{i=1}^T$, we say $\{\mathbf{A}_i\}_{i=1}^T$ satisfies positive definite monotonicity if and only if*

$$\mathbf{A}_i \succeq \mathbf{A}_{i-1} \quad \textit{for} \quad i \in \{2,3,\ldots,T\}.$$

2.2 Problem Setting

We introduce the problem of linear contextual bandits [Chu *et al.*, 2011; Abbasi-Yadkori *et al.*, 2011]. At each round t, the learner receives contexts $\mathbf{x}_{t,a}$ for all $a \in \mathcal{A}$. Then, the learner chooses an action a_t and observes a reward $y_t \in [0,1]$. The reward has the structure $y_t = \mathbf{x}_{t,a_t}^\top \theta_* + \eta_t$ where $\theta_* \in \mathbb{R}^d$ is a unknown true parameter and η_t is a conditionally R-sub-Gaussian noise. That is

$$\mathbb{E}\left[e^{\lambda\eta_t}|\mathbf{x}_{1,a_1},\ldots,\mathbf{x}_{t,a_t};\eta_1,\ldots,\eta_{t-1}\right] \le \exp\left(\frac{\lambda^2R^2}{2}\right).$$

This condition implies that

$$\mathbb{E}\left[\eta_t|\mathbf{x}_{1,a_1},\ldots,\mathbf{x}_{t,a_t};\eta_1,\ldots,\eta_{t-1}\right] = 0.$$

The cumulative regret in this setting is defined by

$$R_T = \sum_{t=1}^T \mathbf{x}_{t,a^*}^\top\theta_* - \sum_{t=1}^T \mathbf{x}_{t,a_t}^\top\theta_*, \tag{1}$$

where $a^* = \arg\max_{a\in\mathcal{A}} \mathbf{x}_{t,a}^\top\theta_*$ is the optimal action at round t. The goal of the learner is to minimize the regret R_T.

2.3 Frequent Directions

Frequent Directions [Liberty, 2013; Ghashami *et al.*, 2016] is a deterministic matrix sketching method for streaming model. It has been applied to speed up many online learning algorithms including linear bandits [Kuzborskij *et al.*, 2019; Luo *et al.*, 2016; Luo *et al.*, 2019]. For any given matrix

$\mathbf{X}_T \in \mathbb{R}^{T\times d}$ which rows come sequentially, FD aims to generate a matrix $\mathbf{Z}_T \in \mathbb{R}^{m\times d}$ such that

$$\mathbf{X}_T^\top \mathbf{X}_T \approx \mathbf{Z}_T^\top \mathbf{Z}_T.$$

Here $m \ll \min\{d, T\}$ is the sketching size. Suppose the given matrix $\mathbf{X}_T = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]^\top$. At round t, FD inserts $\mathbf{x}_t^\top$ into the last row of $\mathbf{Z}_{t-1}$ and performs following manipulations:

$$\begin{aligned}
&[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \mathrm{SVD}(\mathbf{Z}_{t-1}),\\
&\delta_t = \mathbf{\Sigma}_{mm}^2,\\
&\mathbf{Z}_t = \sqrt{\mathbf{\Sigma}^2 - \delta \mathbf{I}_m}\mathbf{V}^\top.
\end{aligned}$$

By using the doubling space technique, FD only need $O(md)$ time to update the model per round (see Section 3.2 of [Ghashami *et al.*, 2016]). Moreover, the approximation matrix is bounded by:

$$\|\mathbf{X}_T^\top \mathbf{X}_T - \mathbf{Z}_T^\top \mathbf{Z}_T\|_2 \le \frac{\|\mathbf{X}_T - (\mathbf{X}_T)_k\|_F^2}{m-k},$$

where $0 < k < m$.

Some variants of FD have been proposed in recent years. Parameterized Frequent Directions [Desai *et al.*, 2016] specifies the proportion of singular values shrunk in each round; Compensative Frequent Directions [Desai *et al.*, 2016] increases the singular values of the sketching matrix. Both methods may increase the performance empirically, but keep the same error bound as traditional FD algorithm. Robust Frequent Directions[Luo *et al.*, 2019] introduces an adaptive regularizer and improves the approximation error bound by a factor $1/2$. Though these variants of FD have different levels of improvement on the FD, they all destroy the positive definite monotonicity of covariance matrices and may increase the regret bound when applied to linear contextual bandits.

3 Main Results

In this section, we first present our SCFD algorithm. Then, we propose CBSCFD algorithm for linear bandits based on SCFD, and analyze the complexity of our approach. Finally we provide theoretical analysis on the regret bounds of our algorithm.

3.1 Spectral Compensation Frequent Direction

Reviewing the procedure of Frequent Directions, we notice that FD subtracts a small term δ_t from singular values in each round. This manipulation will bring approximation errors and break the positive definite monotonicity. Thus, our idea is to compensate the lost spectral information to the approximation. Specifically, we keep a counter α_t to add up the total mass of subtracted values during the Frequent Directions procedure. Then, we compensate the reduced spectrum to the sketched matrix by adding a diagonal matrix $\alpha_t \mathbf{I}_d$. Namely, we use a full rank matrix to approximate the original matrix as follows:

$$\mathbf{X}_t^\top \mathbf{X}_t \succeq \mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d,$$

where $\mathbf{X}_t = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t]^\top$. We present the procedure of SCFD in Algorithm 1. In Algorithm 1, we show how the doubling space technique [Ghashami *et al.*, 2016] is applied to SCFD. Notice that after performing line 5-9, $\mathbf{Z}_t$ will have only $m-1$ rows. Thus, the "if" statement is triggered every $m+1$ iterations, which indicates that the average time cost of SCFD is $O(md)$. Let $\Delta_t = \sum_{i=1}^t \delta_t$, then $\alpha_t = \alpha_0 + \Delta_t$. We have the following theorem:

Algorithm 1 SCFD

```
Input: $\mathbf{X}_T = [\mathbf{x}_1, \ldots, \mathbf{x}_T]^\top$, $m$, $\alpha_0 \ge 0$.
 1: $\mathbf{Z}_0 = \mathbf{0}_{m\times d}$
 2: for $t = 1, 2, \ldots, T$ do
 3:     $\mathbf{Z}_t \leftarrow [\mathbf{Z}_{t-1}^\top, \mathbf{x}_t]^\top$, $\alpha_t \leftarrow \alpha_{t-1}$
 4:     if $\mathbf{Z}_t$ has $2m$ rows then
 5:         Compute SVD: $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] \leftarrow \mathrm{SVD}(\mathbf{Z}_t)$.
 6:         $\delta_t \leftarrow \sigma_m^2(\mathbf{Z}_t)$, $\alpha_t \leftarrow \alpha_{t-1} + \delta_t$.
 7:         $\widehat{\mathbf{\Sigma}} \leftarrow \sqrt{\max\{\mathbf{\Sigma}^2 - \delta_t \mathbf{I}, 0\}}$.
 8:         $\mathbf{Z}_t \leftarrow \widehat{\mathbf{\Sigma}}\mathbf{V}^\top$.
 9:         Remove zero value rows in $\mathbf{Z}_t$
10:     end if
11: end for
Output: $\mathbf{Z}_T, \alpha_T$.
```

Theorem 1. *Assume that SCFD runs with* α_0=0. *Let* $\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d$ *be the sketch of* $\mathbf{X}_t^\top \mathbf{X}_t$ *at round t. Then we have*

$$\|\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d - \mathbf{X}_t^\top \mathbf{X}_t\|_2 \le \frac{\|\mathbf{X}_t - (\mathbf{X}_t)_k\|_F^2}{m-k}$$

for all $0 < k < m$.

Proof. Since $\alpha_t \mathbf{I}_d$ is only added on the output matrix, the sketched matrix $\mathbf{Z}_t$ still satisfies the property of original FD. The Property 1 and Property 2 of Ghashami *et al.* [2016] shows that $\mathbf{Z}_t^\top \mathbf{Z}_t + \Delta_t \mathbf{I}_d \succeq \mathbf{X}_t^\top \mathbf{X}_t \succeq \mathbf{Z}_t^\top \mathbf{Z}_t$. Combining the fact that $\alpha_t = \alpha_0 + \Delta_t = \Delta_t$, we have

$$\|\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d - \mathbf{X}_t^\top \mathbf{X}_t\|_2 \le \Delta_t.$$

Using the fact that $\Delta_t \le \|\mathbf{X}_t - (\mathbf{X}_t)_k\|_F^2/(m-k)$ for all $0 < k < m$ [Ghashami *et al.*, 2016, Theorem 3.1], we get the conclusion of Theorem 1. □

Theorem 1 shows that the error of SCFD is bounded by the spectral tail of the data matrix. Actually, SCFD has the same error bound as the original Frequent Directions. However, SCFD has the property that the sequence of approximation matrices $\{\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d\}_{t=0}^T$ satisfies the positive definite monotonicity, which is very important in the regret analysis. We formally present the property as follows:

Property 1. *Suppose* $\alpha_0 \ge 0$. *When SCFD is running, at each round t, we have*

$$\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d \succeq \mathbf{Z}_{t-1}^\top \mathbf{Z}_{t-1} + \alpha_{t-1} \mathbf{I}_d.$$

Proof. Let $\mathbf{Z}' = [\mathbf{Z}_{t-1}^\top, \mathbf{x}_t]^\top$, then we have

$$\mathbf{Z}_t^\top \mathbf{Z}_t + \delta_t \mathbf{I}_d \succeq \mathbf{Z}'^\top \mathbf{Z}' \succeq \mathbf{Z}_{t-1}^\top \mathbf{Z}_{t-1}. \tag{2}$$

Then, for any unit vector $\mathbf{w}$, we can get

$$\begin{aligned}
&\mathbf{w}^\top(\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d - \mathbf{Z}_{t-1}^\top \mathbf{Z}_{t-1} - \alpha_{t-1}\mathbf{I}_d)\mathbf{w}\\
=&\mathbf{w}^\top(\mathbf{Z}_t^\top \mathbf{Z}_t + \delta_t \mathbf{I}_d - \mathbf{Z}_{t-1}^\top \mathbf{Z}_{t-1})\mathbf{w} \ge 0
\end{aligned} \tag{3}$$

The last step holds because of Eq.(2). □

Another property is that the approximation matrices of SCFD is more well-conditioned than FD and even the original matrix, which indicates that SCFD will make the whole algorithm more stable. Let $\mathbf{V}_{SCFD} = \mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I}_d$, $\mathbf{V}_{FD} = \mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_0 \mathbf{I}_d$ and $\mathbf{V} = \mathbf{X}_t^\top \mathbf{X}_t + \alpha_0 \mathbf{I}_d$. We have the following property:

Property 2. *Suppose $\alpha_0 \geq 0$. When SCFD is running, at each round t, we have*

$$\kappa(\mathbf{V}_{SCFD}) \leq \kappa(\mathbf{V}_{FD}) \text{ and } \kappa(\mathbf{V}_{SCFD}) \leq \kappa(\mathbf{V}).$$

Proof. Since $\alpha_t \geq \alpha_0$, we have

$$\begin{aligned}\kappa(\mathbf{V}_{SCFD}) &= (\sigma_{\max}(\mathbf{Z}_t^\top \mathbf{Z}_t) + \alpha_t)/\alpha_t \\ &\leq (\sigma_{\max}(\mathbf{Z}_t^\top \mathbf{Z}_t) + \alpha_0)/\alpha_0 = \kappa(\mathbf{V}_{FD}). \\ \kappa(\mathbf{V}_{SCFD}) &= (\sigma_{\max}(\mathbf{Z}_t^\top \mathbf{Z}_t) + \alpha_t)/\alpha_t \\ &\leq (\sigma_{\max}(\mathbf{X}_t^\top \mathbf{X}_t) + \alpha_t)/\alpha_t \\ &\leq (\sigma_{\max}(\mathbf{X}_t^\top \mathbf{X}_t) + \alpha_0)/\alpha_0 = \kappa(\mathbf{V}).\end{aligned}$$

□

Remark 1. *Theorem 1 and Property 2 show that our method makes the approximation matrices both theoretically guaranteed and well-conditioned. To this sense, the α_t selected by Algorithm 1 is optimal. Setting α_t to a larger value would lead to breaking Theorem 1. Setting α_t to a smaller value would cause worse condition number of the approximation matrices.*

3.2 Contextual Bandits via Spectral Compensation Frequent Direction

Our method is a sketched version of LinUCB, which find the solution in a confidence ellipsoid [Abbasi-Yadkori *et al.*, 2011]. Suppose $\mathbf{X}_t = [\mathbf{x}_{1,a_1}, \mathbf{x}_{2,a_2}, \ldots, \mathbf{x}_{t,a_t}]^\top$ be the sequence of selected arms of a contextual bandit. The regularized covariance matrix is defined as

$$\mathbf{V}_t = \mathbf{X}_t^\top \mathbf{X}_t + \lambda \mathbf{I}.$$

In each round, LinUCB require $O(d^2)$ to update the inverse of regularized covariance matrix, which is rather costly when d is large. Thus, we adopt SCFD to approximate $\mathbf{V}_t$ in LinUCB by setting $\alpha_0 = \lambda$. We present CBSCFD in Algorithm 2. Then, we show how CBSCFD efficiently computes the inverse of approximated covariance matrix $\widehat{\mathbf{V}}_t^{-1}$. Let $\mathbf{H}_t = (\mathbf{Z}_t \mathbf{Z}_t^\top + \alpha_t \mathbf{I})^{-1}$. By Woodbury Formula, we have $\widehat{\mathbf{V}}_t^{-1} = (\mathbf{Z}_t^\top \mathbf{Z}_t + \alpha_t \mathbf{I})^{-1} = (\mathbf{I} - \mathbf{Z}_t^\top \mathbf{H}_t \mathbf{Z}_t)/\alpha_t$. When the "if" statement is triggered, We have $\mathbf{Z}_t = \widehat{\mathbf{\Sigma}} \mathbf{V}^\top$, thus $\mathbf{H}_t$ can easily computed as $\mathbf{H}_t = (\widehat{\mathbf{\Sigma}}^2 + \alpha_t \mathbf{I})^{-1}$. When the "if" statement is not triggered, we can compute $\mathbf{H}_t$ as follows:

$$\begin{aligned}\mathbf{H}_t &= \left(\begin{bmatrix} \mathbf{Z}_{t-1} \\ \mathbf{x}_t \end{bmatrix} [\mathbf{Z}_{t-1} \quad \mathbf{x}_t]^\top + \alpha_t \mathbf{I} \right)^{-1} \\ &= \begin{bmatrix} \mathbf{H}_{t-1} + \mathbf{p}\mathbf{p}^\top/k & -\mathbf{p}/k \\ -\mathbf{p}^\top/k & 1/k \end{bmatrix}\end{aligned} \tag{4}$$

where $\mathbf{p} = \mathbf{H}_{t-1} \mathbf{Z}_{t-1} \mathbf{x}_t$ and $k = \mathbf{x}_t^\top \mathbf{x}_t - \mathbf{x}_t^\top \mathbf{Z}_{t-1}^\top \mathbf{p} + \alpha_t$.

Algorithm 2 CBSCFD

```
Input: Sketch size $m$, parameter $\lambda > 0$, $\beta > 0$.
$\widehat{\theta}_0 = \mathbf{0}$, $\alpha_0 = \lambda$, $\widehat{\mathbf{V}}_0 = \alpha_0 \mathbf{I}_d$, $\mathbf{Z}_0 = \mathbf{0}_{m \times d}$.
for t = 1, 2, ..., T do
    Observe contexts $\mathbf{x}_{t,a}$.
    Select $a_t = \arg\max_{a \in \mathcal{A}} \left\{ \widehat{\theta}_{t-1}^\top \mathbf{x}_{t,a} + \beta \|\mathbf{x}_{t,a}\|_{\widehat{\mathbf{V}}_{t-1}^{-1}} \right\}$.
    Receive reward $y_t$.
    $\mathbf{Z}_t \leftarrow [\mathbf{Z}_{t-1}^\top, \mathbf{x}_{t,a_t}]^\top,\ \alpha_t \leftarrow \alpha_{t-1}$.
    if $\mathbf{Z}_t$ has $2m$ rows then
        Compute SVD: $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] \leftarrow \mathrm{SVD}(\mathbf{Z}_t)$.
        $\delta_t \leftarrow \sigma_m^2(\mathbf{Z}_t),\ \alpha_t \leftarrow \alpha_{t-1} + \delta_t$.
        $\widehat{\mathbf{\Sigma}} \leftarrow \sqrt{\max\{\mathbf{\Sigma}^2 - \delta_t \mathbf{I}, 0\}},\ \mathbf{Z}_t \leftarrow \widehat{\mathbf{\Sigma}} \mathbf{V}^\top$.
        $\mathbf{H}_t \leftarrow (\widehat{\mathbf{\Sigma}}^2 + \alpha_t \mathbf{I})^{-1}$.
    else
        Compute $\mathbf{H}_t$ as (4).
    end if
    $\widehat{\mathbf{V}}_t^{-1} \leftarrow \frac{1}{\alpha_t}(\mathbf{I} - \mathbf{Z}_t^\top \mathbf{H}_t \mathbf{Z}_t)$
    $\widehat{\theta}_t \leftarrow \widehat{\mathbf{V}}_t^{-1} \sum_{i=1}^{t} \mathbf{x}_{i,a_i} y_i$.
end for
```

Since the size of $\mathbf{H}_t$ is at most $2m$, CBSCFD require at most $O(md)$ time to compute $\mathbf{H}_t$. In addition, our method do not need to compute the elements of $\widehat{\mathbf{V}}_t^{-1}$ because all operations involving $\widehat{\mathbf{V}}_t^{-1}$ are matrix-vector multiplications. Therefore, we only need to consider the matrix-vector multiplications involving $\mathbf{Z}_t$, which require only $O(md)$ time. As discussed in Section 3.1, SCFD only needs to compute SVD every $m+1$ round. Thus, the average time cost per round of CBSCFD is $O(md)$. Since we only need to store $\mathbf{Z}_t$ and $\mathbf{H}_t$, the space complexity of CBSCFD algorithm is $O(md)$.

3.3 Regret Analysis

Define $\mathbf{Y}_t = [y_1, y_2, \ldots, y_t]^\top$, then we have $\widehat{\theta}_t = \widehat{\mathbf{V}}_t^{-1} \mathbf{X}_t \mathbf{Y}_t$. The upper regret bound of Algorithm 2 is summarized in the following theorem.

Theorem 2. *Assume that $\|\theta_*\|_2 \leq S$, $\|\mathbf{x}_{t,a}\|_2 \leq L$ and η_t is a R-sub-Gaussian noise for $t \in \{1, 2, \ldots, T\}$. If Algorithm 2 runs with $\beta = \beta_T(\delta)$, then with probability $1 - \delta$, the regret of Algorithm 2 is*

$$Regret(T) \leq \beta_T(\delta) \sqrt{8mT \log\left(1 + \frac{TL^2}{m\lambda}\right)}$$

where

$$\begin{aligned}\beta_T(\delta) = & R\sqrt{2\log\frac{1}{\delta} + m\log\left(1 + \frac{TL^2}{m\lambda}\right) + d\log\left(1 + \frac{\Delta_T}{\lambda}\right)} \\ & + S\sqrt{\lambda + \Delta_T}.\end{aligned}$$

Proof sketch. The first step is to find the confidence ellipsoid where θ_* lies. Actually, we can obtain that θ_* lies in the set

$$\mathcal{C}_t = \{\theta : \|\widehat{\theta}_t - \theta\|_{\widehat{\mathbf{V}}_t} \leq \beta_t(\delta)\} \tag{5}$$

Algorithm	Time cost per round	Space	Upper regret bound
LinUCB	$O(d^2)$	$O(d^2)$	$\tilde{O}(d\sqrt{T})$
CBRAP	$O(md + m^3)$	$O(md)$	$\tilde{O}(\sqrt{mT} + m^{-1/2}T)$
SOFUL	$O(md)$	$O(md)$	$\tilde{O}((1+\Delta_T)^{3/2}(m + d\log(1+\Delta_T))\sqrt{T})$
Our method	$O(md)$	$O(md)$	$\tilde{O}((\sqrt{m + d\log(1+\Delta_T)} + \sqrt{\Delta_T})\sqrt{mT})$

Table 1: Comparison of our theoretical results with state-of-the-art approaches. Here we consider the parameter λ as a constant.

with probability $1-\delta$. Notice that the line 4 of Algorithm 2 is equivalent to solving the following problem:

$$(\mathbf{x}_{t,a_t}, \widetilde{\theta}_t) = \arg\max \mathbf{x}^\top \theta \quad \text{s.t. } (\mathbf{x}, \theta) \in \mathcal{D}_t \times \mathcal{C}_{t-1},$$

where $\mathcal{D}_t$ is the decision set that $\mathbf{x}_{t,a}$ belongs to. Thus, the action a_t chosen by Algorithm 2 satisfies $\mathbf{x}_{t,a_t}^\top \widetilde{\theta}_t \geq \mathbf{x}_t^{*\top}\theta_*$.

We next consider the regret r_t at round t:

$$\begin{aligned} r_t =& \mathbf{x}^{*\top}_{t,a_t}\theta_* - \mathbf{x}_{t,a_t}^\top \theta_* \\ \leq& \mathbf{x}_{t,a_t}^\top \widetilde{\theta}_t - \mathbf{x}_{t,a_t}^\top \theta_* \\ =& \mathbf{x}_{t,a_t}^\top (\widehat{\theta}_{t-1} - \theta_*) + \mathbf{x}_{t,a_t}^\top (\widetilde{\theta}_t - \widehat{\theta}_{t-1}) \\ \leq& \|\mathbf{x}_{t,a_t}\|_{\widehat{\mathbf{V}}_t^{-1}} \left(\|\widehat{\theta}_{t-1} - \theta_*\|_{\widehat{\mathbf{V}}_t} + \|\widetilde{\theta}_t - \widehat{\theta}_{t-1}\|_{\widehat{\mathbf{V}}_t} \right) \\ \leq& 2\beta_t(\delta)\|\mathbf{x}_{t,a_t}\|_{\widehat{\mathbf{V}}_t^{-1}} \end{aligned}$$

Since $r_t \leq 2$, we have

$$r_t \leq 2\min(\beta_t(\delta)\|\mathbf{x}_t\|_{\widehat{\mathbf{V}}_t^{-1}}, 1) \leq 2\beta_t(\delta)\min(\|\mathbf{x}_t\|_{\widehat{\mathbf{V}}_t^{-1}}, 1).$$

Then, we have

$$\sum_{t=1}^T r_t \leq \sqrt{T\sum_{t=1}^T r_t^2} \leq 2\beta_T(\delta)\sqrt{2mT\log\left(1 + \frac{TL^2}{m\lambda}\right)}$$

where the last step follow from the following lemma:

Lemma 1.

$$\sum_{t=1}^T \min(1, \|\mathbf{x}_t\|^2_{\widehat{\mathbf{V}}_{t-1}^{-1}}) \leq 2m\log\left(1 + \frac{TL^2}{m\lambda}\right).$$

3.4 Discussion

Briefly, our bound can be written as

$$\tilde{O}((\sqrt{m + d\log(1 + \Delta_T/\lambda)} + \sqrt{\lambda + \Delta_T})\sqrt{mT}).$$

According to Theorem 1, we have

$$\Delta_T \leq \|\mathbf{X}_T - (\mathbf{X}_T)_k\|_F^2/(m-k)$$

for all $0 < k < m$. Thus, Δ_T is very small when the covariance matrix is approximately low-rank. Especially, when the rank of covariance matrix is less than m, we have $\Delta_T = 0$. In this situation, our regret bound becomes $\tilde{O}(m\sqrt{T})$. When the covariance matrix is not low-rank, our method significantly reduces the influence of Δ_T when compared with SOFUL [Kuzborskij *et al.*, 2019]. Note that the regret bound of SOFUL is

$$\tilde{O}((1 + \Delta_T/\lambda)^{3/2}(m + d\log(1 + \Delta_T/\lambda))\sqrt{T}).$$

Our method reduces the order of Δ_T from $3/2$ to $1/2$. In addition, our regret bound decouples the dimension d and Δ_T, which further reduces the influence of Δ_T. Finally, our bound is less sensitive to the parameter λ (which is usually a small number in practice) because the term Δ_T/λ is in the logarithmic function. Overall, our method is more effective and robust than SOFUL. We summarize the comparison between our theoretical results and state-of-the-art methods in Table 1. From the table, we can find that our method has the best regret bound in high dimensional case.

4 Experiments

In this section, we empirically verify the efficiency and effectiveness of our CBSCFD algorithm. We conduct experiments on both synthetic data and real-world data sets. The baseline approaches include LinUCB [Abbasi-Yadkori *et al.*, 2011], CBRAP [Yu *et al.*, 2017] and SOFUL [Kuzborskij *et al.*, 2019]. For CBRAP algorithm, we use the Gaussian random matrix as the projection matrix because it has better performance. We conduct all experiments on a Linux server which contains 8 processors and has total memory of 32GB. The code is implemented in Matlab R2017b.

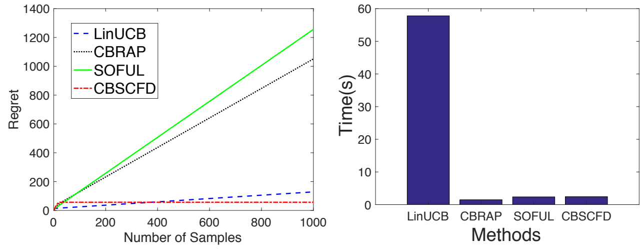
4.1 Synthetic Data

We generated a synthetic data set with 100 arms and 2000 features per context. Specifically, all contexts $\mathbf{x}_{t,a} \in \mathbb{R}^{2000}$ are drawn independently from multivariate Gaussian distributions $\mathbf{x}_{t,a} \sim MVN(\mathbf{1}, \mathbf{I}_{2000})$. The true parameter θ_* is computed as $\theta_* = \theta'/\|\theta'\|_2$ where θ' is drawn from a multivariate Gaussian distribution $\theta' \sim MVN(\mathbf{0}, \mathbf{I}_{2000})$. We set $T = 1000$ and run the experiments for 20 times. The parameter β of all methods is searched in $\{10^{-4}, 10^{-3}, \ldots, 1\}$ and λ is searched in $\{2\times10^{-4}, 2\times10^{-3}, \ldots, 2\times10^{4}\}$. We choose the best values for each approach and report the average results in Figure 1.

In Figure 1, we can find that the running time of LinUCB is much larger than other three methods which use matrix sketching. The CBRAP is a little faster than SOFUL and CBSCFD. The reason is that SOFUL and CBSCFD require to perform SVD decomposition, which is much slower than matrix multiplication though they have the same time complexity. Compare the cumulative regrets in Figure 1, we can find that our CBSCFD outperforms other approaches.

4.2 Online Classification

Then, we perform online classification to evaluate the performance of our methods. We follow the experiments setup of [Kuzborskij *et al.*, 2019]. Specifically, we fit the online classification problem into the contextual bandit setting as follows.



Average Cumulative Regrets

Running Time

Figure 1: Comparison of the average cumulative regrets and running time on the synthetic data set.

Data set	#Samples	#Features	#Classes	Sketch size
MNIST	60000	784	10	10
CIFAR10	50000	3072	10	20

Table 2: Summary of Data sets for Contextual Bandits.

Given a data set with data in K clusters, we first choose one cluster as the target cluster. In each round the environment randomly draws one sample from each class and composes a set of contexts of K samples. The learner chooses one sample from the set and observe the reward corresponding to whether the selected sample belongs to the target cluster. The reward is 1 if the selected sample comes from the target cluster, and is 0 otherwise.

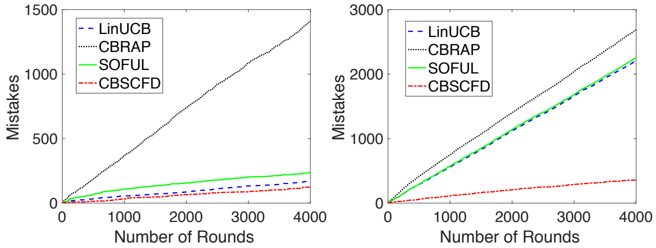
We perform our experiments on two real-world data sets: MNIST [LeCun *et al.*, 1998] and CIFAR10 [Krizhevsky and Hinton, 2009]. We summarize the statistic information of both data sets in Table 2. For MNIST data set, we perform experiments with sketch sizes $m = 10$. For CIFAR10, we set the sketch sizes $m = 20$. As previous experiment, the parameter β is searched in $\{10^{-4}, 10^{-3}, \ldots, 1\}$ and λ is searched in $\{2 \times 10^{-4}, 2 \times 10^{-3}, \ldots, 2 \times 10^{4}\}$. We run the experiments for 20 times and report the average online mistakes in Figure 2. We find that our CBSCFD algorithm outperforms all other methods on MNIST and CIFAR10 data sets. This result validates the effectiveness of our method.

4.3 Robustness of CBSCFD

Finally, we investigate the robustness of our method. We compare the sensitivity to parameter λ between our method and baseline approaches. Since CBRAP always sets $\lambda = 1$, we do not include it as baseline. In this experiment, we fix the parameter $\beta = 0.01$ and $m = 10$. Then we perform experiments on MNIST data set with $\{2 \times 10^{-4}, 2 \times 10^{-3}, \ldots, 2 \times 10^{4}\}$. We present the result in Figure 3. This figure shows that the performance of CBSCFD only changes slightly when λ is changed. But the performance of other two methods significantly depend on the choice of λ. This result validates our theoretical analysis in Property 2 and shows that our method is robust than others.

4.4 Discussion

It is surprising to see that our method outperforms LinUCB, which adopts no approximations. This happens because our experiments focuses on high dimensional setting, i.e., $d \approx T$.



MNIST, $m = 10$

CIFAR, $m = 20$

Figure 2: The comparison of online mistakes among different contextual bandit algorithms on real-world data sets with different sketch sizes.

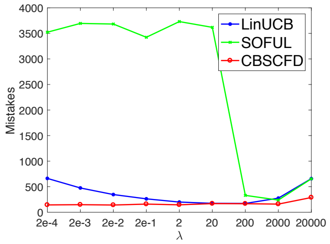


Figure 3: Total mistakes with different choice of parameter λ.

When the iteration round $t < d$, the matrix $\mathbf{X}_t\mathbf{X}_t^\top$ is singular. LinUCB uses a constant regularization and enhances the spectrum by λ. When the regulrization parameter λ is very small, the matrix $\mathbf{V}_t$ is ill-conditioned, which makes LinUCB unstable. On the other hand, CBSCFD enhances the spectrum by $\Delta_t + \lambda$, where Δ_t is updated in each round and no more than the smallest nonzero singular values of $\mathbf{X}_t\mathbf{X}_t^\top$. Thus, the approximated covariance matrix may have better performance than the original one. Also, the covariance matrix of CBSCFD is more well-conditioned than that of LinUCB, which makes the algorithm more stable and less sensitive to λ.

5 Conclusions

In this paper, we present a variant of FD, which is called SCFD, for matrix sketching. We then propose CBSCFD algorithm for high-dimensional linear contextual bandits based on SCFD. Our method is much more efficient than LinUCB and require less space. Compared with previous methods which use matrix sketching to accelerate linear contextual bandits, our method has better upper regret bound and more robust. Finally, the experiments demonstrated the efficiency, effectiveness and robustness of CBSCFD.

Acknowledgments

Luo Luo was partially supported by Baidu academic collaboration program. The corresponding authors Yong Yu and Weinan Zhang thank the support of NSFC 61702327 and 61772333.

References

[Abbasi-Yadkori *et al.*, 2011] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[Agrawal and Goyal, 2013] Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 127–135, 2013.

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

[Calandriello *et al.*, 2019] Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Proceedings of the 32nd Annual Conference on Learning Theory*, pages 533–557, 2019.

[Carpentier and Munos, 2012] Alexandra Carpentier and Rémi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Proceedings of the 15th Artificial Intelligence and Statistics*, 2012.

[Chu *et al.*, 2011] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.

[Dani *et al.*, 2008] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, 2008.

[Desai *et al.*, 2016] Amey Desai, Mina Ghashami, and Jeff M Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.

[Deshpande and Montanari, 2012] Yash Deshpande and Andrea Montanari. Linear bandits in high dimension and recommendation systems. In *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1750–1754. IEEE, 2012.

[Dumitrascu *et al.*, 2018] Bianca Dumitrascu, Karen Feng, and Barbara Engelhardt. Pg-ts: Improved thompson sampling for logistic contextual bandits. In *Advances in Neural Information Processing Systems*, pages 4629–4638, 2018.

[Filippi *et al.*, 2010] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, pages 586–594, 2010.

[Ghashami *et al.*, 2016] Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009.

[Kuzborskij *et al.*, 2019] Ilja Kuzborskij, Leonardo Cella, and Nicolò Cesa-Bianchi. Efficient linear bandits through matrix sketching. *Proceedings of the 22th Artificial Intelligence and Statistics*, 2019.

[Langford and Zhang, 2008] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems*, pages 817–824, 2008.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2010] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670. ACM, 2010.

[Li *et al.*, 2017] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2071–2080, 2017.

[Liberty, 2013] Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 581–588. ACM, 2013.

[Luo *et al.*, 2016] Haipeng Luo, Alekh Agarwal, Nicolo Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems*, pages 902–910, 2016.

[Luo *et al.*, 2019] Luo Luo, Cheng Chen, Zhihua Zhang, Wu-Jun Li, and Tong Zhang. Robust frequent directions with application in online learning. *Journal of Machine Learning Research*, 20(45):1–41, 2019.

[Russo and Van Roy, 2014] Daniel Russo and Benjamin Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 2014.

[Tang *et al.*, 2013] Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 1587–1594. ACM, 2013.

[Tewari and Murphy, 2017] Ambuj Tewari and Susan A Murphy. From ads to interventions: Contextual bandits in mobile health. In *Mobile Health*. Springer, 2017.

[Yu *et al.*, 2017] Xiaotian Yu, Michael R Lyu, and Irwin King. Cbrap: Contextual bandits with random projection. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017.

[Zhao *et al.*, 2014] Tong Zhao, Julian McAuley, and Irwin King. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*. ACM, 2014.