Bandit Theory meets Compressed Sensing for high dimensional Stochastic Linear Bandit

Alexandra Carpentier, Rémi Munos

Bandit Theory meets Compressed Sensing for high dimensional Stochastic Linear Bandit

Alexandra Carpentier **Remi Munos**
INRIA Lille - Nord Europe

Abstract

We consider a linear stochastic bandit problem where the dimension K of the unknown parameter θ is larger than the sampling budget n. In such cases, it is in general impossible to derive sub-linear regret bounds since usual linear bandit algorithms have a regret in $O(K\sqrt{n})$. In this paper we assume that θ is $S-$sparse, i.e. has at most $S-$non-zero components, and that the space of arms is the unit ball for the $||.||_2$ norm. We combine ideas from Compressed Sensing and Bandit Theory and derive algorithms with regret bounds in $O(S\sqrt{n})$. We detail an application to the problem of optimizing a function that depends on many variables but among which only a small number of them (initially unknown) are relevant.

Introduction

We consider linear stochastic bandit problem in very high dimension K. At each round t, from 1 to n, the player chooses x_t in a fixed set of arms and receives a reward $r_t = \langle x_t, \theta + \eta_t \rangle$, where $\theta \in \mathbb{R}^K$ is an unknown parameter and η_t is a noise term. Note that r_t is a (noisy) projection of θ on x_t. The goal is to maximize the sum of rewards.

We are interested in cases where the total number of rounds is much less than the dimension of the parameter, i.e. $n \ll K$. This is new in bandit literature but useful in practice, as illustrated by the problem of gradient ascent for a high-dimensional function, described later.

As $n \ll K$, it is in general impossible to even estimate θ in an accurate way. It is thus necessary to restrict the setting, and the assumption we consider here is that θ is *sparse*. We assume also that the set of arms to which x_t belongs is the unit ball with respect to the norm $||.||_2$ norm, induced by the inner product[1].

Bandit Theory meets Compressed Sensing Our problem asks in an urging way the fundamental question at the heart of bandit theory, namely the exploration[2] versus exploitation[3] dilemma. More precisely, when the dimension of the space K is smaller than the budget n, it is possible to project the parameter θ *at least once* on *each* directions of a basis (e.g. the canonical basis): it is easy to explore efficiently. In our setting, $K \gg n$ and it is not possible anymore to project *even once* on *each* directions of any basis of the space: we thus require a cleverer exploration technique.

Compressed Sensing provides us with ideas on how to explore, i.e. estimate θ, *provided that it is sparse*, with few measurements: it is thus possible to roughly estimate its support without spending too much of the budget. The idea is to project θ on random (isotropic) directions such that each reward sample provides equal information about *all* coordinates of θ. This is the reason why we emphasized the fact that the set of arm is the unit ball, as we need to be able to project θ in each direction of the space. Then using regularization (Hard Thresholding, Lasso, Dantzig selector...) enables to recover the support of the parameter. For some references on Compressed Sensing, see e.g. (Candes and Tao, 2007; Chen et al., 1999; Blumensath and Davies, 2009). Note however that such a technique

Preliminary work. Under review by AISTATS 2012. Do not distribute.

[1] The fact that one can consider the unit ball is not usual, as linear bandits applications involve most of the time constrained subsets of arms, for instance in order to model graphs. The setting described here has however interesting applications such as gradient ascent.

[2] It is important to explore the space in order to have a good estimate of θ and to know which ones are the best arms.

[3] It is important to exploit, i.e. to pull the empirical best arms in order to maximize the sum of rewards.

allows only to retrieve a rough estimate of θ and is not designed for the purpose of maximizing the sum of rewards.

Bandit Theory is then a good tool to address this second issue, namely maximizing the sum of rewards by efficiently balancing between exploration and exploitation. In our setting, once a rough estimate of the (restricted) support of θ is available, we use a linear bandit algorithm. References on linear stochastic bandits include the works of Rusmevichientong and Tsitsiklis (2008); Dani et al. (2008); Filippi et al. (2010) and the recent work by Y. Abbasi-yadkori (2011).

Contributions: Our contributions are the following.

- We provide two algorithms that mix ideas of Compressed Sensing and Bandit Theory for solving the exposed problem. The first algorithm has a regret[4] which is either of order $O(S\sqrt{n}/||\theta||_2)$ when $||\theta||_2 = \Omega(n^{-1/6})$[5] or of order $O(Sn^{2/3})$ when $||\theta||_2 = O(n^{-1/6})$[6]. If we additionally assume that η_t is sparse, it is possible to build another algorithm such that the regret is of order $O(S\sqrt{n})$ and *independent of* θ.
- We give a detailed example of an application of this setting to high dimensional gradient ascent when the gradient is sparse. We first explain why the setting of gradient ascent can be seen as a bandit problem. We then display numerical experiments supporting our belief that our algorithms are efficient ways for solving the problem of high dimensional gradient ascent on functions that depend only on a small number of relevant variables.

We formalize the setting in Section 1, and recall briefly what linear bandits can achieve when the dimension K is low. We then describe the algorithms we propose for this problem, and give the main results in Sections 2 and 3. We detail in Section 4 the application to gradient ascent and provide numerical experiments for illustration. The Supplementary Material provides proofs of the main Theorems.

1 Setting and a useful existing result

1.1 Description of the problem

We consider a linear bandit problem in dimension K. An algorithm (or strategy) $\mathcal{A}lg$ is given a budget of n pulls. At each round $1 \leq t \leq n$ it selects an arm x_t in the set of arms $\mathcal{B}_K$, which is the unit ball for the $||.||_2$-norm induced by the inner product. It then receives a reward

$$r_t = \langle x_t, \theta + \eta_t \rangle,$$

where $\eta_t \in \mathbb{R}^K$ is an i.i.d. white noise[7] that is independent from the past actions, i.e. from $\left\{(x_{t'})_{t' \leq t}\right\}$ and $\theta \in \mathbb{R}^K$ is an unknown parameter.

We define the *performance* of algorithm $\mathcal{A}lg$ as

$$L_n(\mathcal{A}lg) = \sum_{t=1}^{n} \langle \theta, x_t \rangle. \tag{1}$$

Note that $L_n(\mathcal{A}lg)$ differs from the sum of rewards $\sum_{t=1}^n r_t$. But if we assume for instance that each $\eta_{k,t}$ is bounded by $\frac{1}{2}\sigma_k$, we know by Hoeffding inequality (because x_t is independent of $\eta_{k,t}$) that with probability $1-\delta$, we have $\sum_{t=1}^n r_t = L_n(\mathcal{A}lg) + \sum_{t=1}^n \langle \eta_t, x_t \rangle \leq L_n(\mathcal{A}lg) + \sqrt{2\log(1/\delta)}||\sigma||_2\sqrt{n}$. Note that this result can be extended to sub-gaussian random variables $\eta_{k,t}$.

If the parameter θ was known, we could define an optimal fixed strategy $\mathcal{A}lg^*$ that always picks $x^* = \arg\max_{x \in \mathcal{B}_K} \langle \theta, x \rangle$ in order to maximize the performance. Here, $x^* = \frac{\theta}{||\theta||_2}$. The performance of $\mathcal{A}lg^*$ is given by

$$L_n(\mathcal{A}lg^*) = n||\theta||_2. \tag{2}$$

We define the *regret* of an algorithm $\mathcal{A}lg$ with respect to this optimal strategy as

$$R_n(\mathcal{A}lg) = L_n(\mathcal{A}lg^*) - L_n(\mathcal{A}lg). \tag{3}$$

We consider the class of algorithms that do not know the parameter θ. Our objective is to find an adaptive strategy $\mathcal{A}lg$, i.e. using the history $\{(x_1, r_1), \ldots, (x_{t-1}, r_{t-1})\}$ at time t to choose the next state x_t, in order to minimize the regret.

For a given t, we write $X_t = (x_1; \ldots; x_t)$ the matrix in $\mathbb{R}^{K \times t}$ of all chosen arms until time t, and $R_t = (r_1, \ldots, r_t)^T$ the vector in $\mathbb{R}^t$ of all rewards until time t.

In this paper, we consider the case where the dimension K is much larger than the budget, i.e. $n \ll K$. As already mentioned, it is impossible in general to estimate accurately the parameter and thus achieve a sub-linear regret. This is the reason why we make the assumption that θ is $S-$sparse (i.e. there are at most S components of θ which are not 0) with $S < n$.

[4] We define the notion of regret in Section 1.

[5] We use the conventional notation: $f(n) = \Omega(g(n))$ means that $\exists c / \forall n, f(n) \geq cg(n)$.

[6] We use the conventional notation: $f(n) = O(g(n))$ means that $\exists c / \forall n, f(n) \leq cg(n)$.

[7] This means that $\mathbb{E}_{\eta_t}(\eta_{k,t}) = 0$ for every (k,t), that the $(\eta_{k,t})_k$ are independent and that the $(\eta_{k,t})_t$ are i.i.d..

1.2 A useful algorithm for Linear Bandits

In this paper we use the algorithm $ConfidenceBall_2$, which we abbreviate by CB_2, and that is described in the article of Dani et al. (2008). We recall here briefly the algorithm and the corresponding regret bound.

This algorithm is designed for stochastic linear bandit in dimension d where d is *smaller* than the budget n. This is the reason why we can not immediatly apply this algorithm to the problem we described in the previous subsection. The bandit parameter θ here is in $\mathbb{R}^d$.

The pseudo-code of the algorithm is presented in Figure 1. The idea is to build an ellipsoid of confidence for the parameter θ, namely $B_t = \{\nu : ||\nu - \hat{\theta}_t||_{2,A_t} \leq \sqrt{\beta_t}\}$ where $||u||_{2,A_t} = u^T A u$, and to pull the arm with largest inner product with a vector in B_t.

Note that this algorithm is intended for general shapes of the set of arms, and that we can apply it in the particular case where the set of arms is the unit ball for the $||.||_2$ norm in $\mathbb{R}^d$, i.e. $\mathcal{B}_d$. This case is simple. At first, it is easier to find a span in the set of arms: we can just take the canonical basis of $\mathbb{R}^d$. Then we need to find the point of the confidence ellipsoid B_t with largest norm in order to compute the upper confidence bound: the maximization problem is simplified. Note also that we present here a simplified variant where the temporal horizon n is known: the original version of the algorithm in (Dani et al., 2008) is anytime.

We recall here Theorem 2 of (Dani et al., 2008).

Theorem 1 ($ConfidenceBall_2$) *Assume that the η_t is an i.i.d. white noise, independent of the $(x_{t'})_{t' \leq t}$ and that for all $k = \{1, \ldots, d\}$, $\exists \sigma_k$ such that for all t, $|\eta_{t,k}| \leq \frac{1}{2}\sigma_k$. If n is large enough, we have with probability $1 - \delta$ the following bound for the regret of $ConfidenceBall_2(\mathcal{B}_{2,d}, \delta)$:*

$$R_n(\mathcal{A}lg_{CB_2}) \leq 64d\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n}.$$

2 The algorithm SL-UCB

We present here a first algorithm which can address the general case for the noise. We call it *Sparse Linear Upper Confidence Bound* (SL-UCB).

2.1 Presentation of the algorithm

SL-UCB is divided in two main parts, (i) a first unadaptive phase, using compressed sensing ideas and referred to as *support exploration phase* where we project θ on isotropic random vectors in order to select the arms that belong to what we call the *active set* $\mathcal{A}$ and (ii) a phase that we call *restricted linear bandit phase* where we apply a linear bandit algorithm to the active set $\mathcal{A}$ in order to balance exploration and exploitation and further minimize the regret. Note that the length of the support exploration phase is problem dependent.

This algorithm takes as parameters: $\bar{\sigma}_2$ and $\bar{\theta}_2$ which are upper bounds respectively on $||\sigma||_2$ and $||\theta||_2$, and δ which is a (small) probability.

First, we define an *exploring set* as

$$\mathcal{E}_{xploring} = \frac{1}{\sqrt{K}}\{-1, +1\}^K. \tag{4}$$

Note that $\mathcal{E}_{xploring} \subset \mathcal{B}_K$. We sample this set uniformly during the support exploration phase. This gives us some insight about the directions on which the parameter θ is sparse, using very simple concentration tools[8]: at the end of this phase, the algorithm selects a set of coordinates $\mathcal{A}$, named *active set*, which are the directions where θ is likely to be non-zero. The length of this phase is problem dependent and is either $n^{2/3}$ or $\sqrt{n}$: if the problem is *difficult*, i.e. $||\theta||_2 = O(n^{-1/6})$, we need a longer support exploration phase than if the problem is *simple*, i.e. $||\theta||_2 = \Omega(n^{-1/6})$. Note that the algorithm automatically adapts the length of this phase and that no lower bound on $||\theta||_2$ is needed.

We then exploit the information collected in the first phase, i.e. the active set $\mathcal{A}$, by doing a linear bandit algorithm on the intersection of the unit ball B_K and the small vector subspace generated by the active set $\mathcal{A}$, i.e. $Vec(\mathcal{A})$. Here we choose to use the algorithm CB_2 described in (Dani et al., 2008). See Subsection 1.2 for an adaptation of this algorithm to our specific case: the set of arms is indeed the unit ball for the $||.||_2$ norm in the vector subspace $Vec(\mathcal{A})$.

The algorithm is described in Figure 2.

Note that the algorithm computes $\hat{\theta}_{k,\sqrt{n}}$, as well as $\hat{\theta}_{k,n^{2/3}}$, using

$$\hat{\theta}_{k,t} = \frac{K}{t}\Big(\sum_{i=1}^{t} x_{k,i} r_i\Big) = \big(\frac{K}{t} X_t R_t\big)_k. \tag{5}$$

2.2 Main Results

We first state an assumption on the noise.

Assumption 1 *$(\eta_{k,t})_{k,t}$ is an i.i.d. white noise and $\exists \sigma_k$ s.t. $|\eta_{k,t}| \leq \frac{1}{2}\sigma_k$.*

[8] Note that this idea is very similar to the one of compressed sensing.

Input: $\mathcal{B}_{2,d}$, δ
Initialization:
$A_1 = I_d$, $\hat{\theta}_1 = 0$, $\beta_t = 128d(\log(n^2/\delta))^2$.
for $t = 1, \ldots, n$ **do**
  $B_t = \{\nu : ||\nu - \hat{\theta}_t||_{2,A_t} \leq \sqrt{\beta_t}\}$
  $x_t = \arg\max_{x \in \mathcal{B}_d} \max_{\nu \in B_t} \langle \nu, x \rangle$.
  Observe $r_t = \langle x_t, \theta + \eta_t \rangle$.
  $A_{t+1} = A_t + x_t x_t'$, $\hat{\theta}_{t+1} = A_{t+1}^{-1} X_t R_t$.
end for

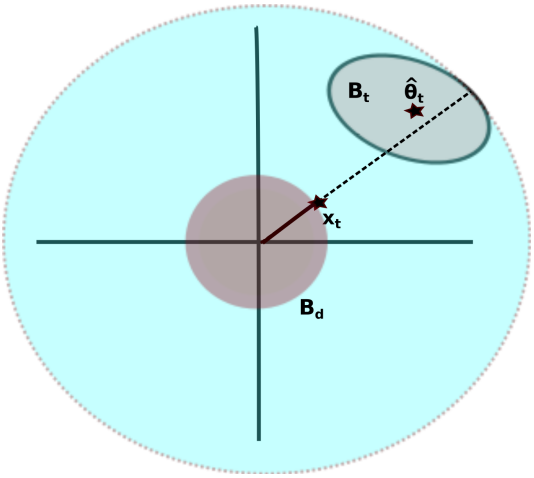

Figure 1: Algorithm $ConfidenceBall_2$ (CB_2) adapted for an action set of the form $\mathcal{B}_d$ (Left), and illustration of the maximization problem defining x_t (Right).

Input: parameters $\bar{\sigma}_2$, $\bar{\theta}_2$,δ.
Initialize: Set $b = (\bar{\theta}_2 + \bar{\sigma}_2)\sqrt{2\log(2K/\delta)}$.
Support Exploration Phase:
For $t = 1, \ldots, \sqrt{n}$, pull uniformly randomly an arm x_t in $\mathcal{E}_{xploring}$ (defined in Equation 4) and observe r_t
Compute $\hat{\theta}_{k,\sqrt{n}}$ using Equation 5.
if $\max_k(|\hat{\theta}_{k,\sqrt{n}}|) \geq \frac{b}{n^{1/6}}$ **then**
  Set $\mathcal{A} = \left\{k : \hat{\theta}_{k,\sqrt{n}} \geq \frac{2b}{n^{1/4}}\right\}$
  Restricted Linear Bandit Phase:
  For $t = \sqrt{n}+1, \ldots, n$, apply $CB_2(\mathcal{B}_K \cap Vec(\mathcal{A}), \delta)$ and collect the r_t.
else
  Support Exploration Phase bis:
  For $t = \sqrt{n}+1, \ldots, n^{2/3}$, pull randomly an arm x_t in $\mathcal{E}_{xploring}$ (defined in Equation 4) and observe r_t
  Compute $\hat{\theta}_{n^{2/3}}$ using Equation 5.
  Set $\mathcal{A} = \left\{k : \hat{\theta}_{k,n^{2/3}} \geq \frac{2b}{n^{1/3}}\right\}$
  Restricted Linear Bandit Phase:
  For $t = n^{2/3}+1, \ldots, n$, apply $CB_2(\mathcal{B}_K \cap Vec(\mathcal{A}), \delta)$ and collect the r_t.
end if

Figure 2: The pseudo-code of the SL-UCB algorithm.

Note that this assumption is made for simplicity and that it could easily be generalized to, for instance, sub-Gaussian noise.

This algorithm behaves differently depending on how *difficult* the problem is, which means here on how difficult it is to find the support. Note that the larger $||\theta||_2$, the easier it is to identify the principal non-zeros components.

Theorem 2 *Under Assumption 1, if we choose* $\bar{\sigma}_2 \geq ||\sigma||_2$, *and* $\bar{\theta}_2 \geq ||\theta||_2$, *the regret of SL-UCB is bounded with probability at least* $1 - 5\delta$

(i) when $||\theta||_2 \geq \frac{2b\sqrt{S}}{n^{1/6}}$, *i.e. when the problem is easy*

$$R_n(\mathcal{A}lg_{SL-UCB}) \leq 168S\Big(\sqrt{\bar{\theta}_2} + \frac{\bar{\sigma}_2}{\sqrt{||\theta||_2}}\Big)^2(\log(2n^2K/\delta))^2\sqrt{n}.$$

(ii) when $||\theta||_2 \leq \frac{2b\sqrt{S}}{n^{1/6}}$, *i.e. when it is difficult,*

$$R_n(\mathcal{A}lg_{SL-UCB}) \leq 336S\big(\bar{\theta}_2 + 10(\bar{\sigma}_2 + 1)^2\big)(\log(2n^2K/\delta))^2n^{2/3}.$$

We report full proofs of this Theorem in the Supplementary Material (Appendix A).

The algorithm SL-UCB uses at first an idea of compressed sensing: it explores by performing random projection and builds an estimate of θ. It then selects the support as soon as the uncertainty is small enough, and applies CB_2 to the selected support. The particularity of this algorithm is that the length of the support exploration phase adjusts to the difficulty of finding the support. More precisely, if $||\theta||_2$ is small, i.e. of order $O(n^{-1/6})$ it is difficult to find the biggest components of the support. In that case, the regret deteriorates because the length of the support exploration phase has to be $n^{2/3}$ in order to determine the biggest principal non-zeros entries of θ. This subsequently leads to a regret of order $O(Sn^{2/3})$. On the other hand, if $||\theta||_2$ is big, i.e. of order $\Omega(n^{-1/6})$, it is easy to find the biggest components of the support. It is sufficient to have a support exploration phase of length $\sqrt{n}$. This leads to a regret of order $O(\frac{S\sqrt{n}}{||\theta||_2})$.

3 The algorithm S^{2}L-UCB

When $||\theta||_2$ is too small, i.e. $||\theta||_2 = O(n^{-1/6})$, the algorithm SL-UCB needs $n^{2/3} (\gg \sqrt{n})$ time steps to find the support, and in this case the regret is of order $O(Sn^{2/3}) (\gg O(S\sqrt{n}))$. Now, we make a stronger assumption on the noise, namely that it is sparse. Under this assumption, we can build an algorithm such that the regret is in $O(S\sqrt{n})$ *whatever the value of* $||\theta||_2$.

We call the corresponding algorithm *Sparse Square Linear Upper Confidence Bound* (S^{2}L-UCB).

3.1 Presentation of the algorithm

Again, the S^{2}L-UCB algorithm is divided in two parts, the *support exploration phase* where we sample the function in order to choose which arms belong to the *active set* $\mathcal{A}(t)$ and the *Restricted Linear Bandit Phase* where we apply a linear bandit algorithm to the active set $\mathcal{A}(t)$. Note that the active set $\mathcal{A}(t)$ evolves in time for S^{2}L-UCB.

This algorithm takes as parameters: S, an upper bound on the sparsity of θ, and δ which is a (small) probability.

The design of the support exploration phase for this algorithm is very different from the one for SL-UCB. Here, the length of the support exploration phase is fixed, but the way we explore the support evolves in time. It is divided in $n_1 = \lfloor \log(K/2S)(S+1) \rfloor + 1$ phases. Some indexes are removed from the active set $\mathcal{A}(t)$ at the end of each of those n_1 phases[9]. During each of those phases, the algorithm chooses randomly $n_2 = \lfloor \log(1/\delta) \exp(1) \rfloor + 1$ arms x drawn from $Ł(\mathcal{A}(t))$, where $Ł(\mathcal{A}(t))$ is a probability distribution defined later in this Subsection. And the algorithm pulls $n_3 = \lfloor \log(1/\delta)\sqrt{n} \rfloor + 1$ times each of those chosen arm x. If for a given x, the observed reward samples are always zero, all the indexes k such that $x_k \neq 0$ are removed from the active set. Note that the length of the support exploration phase is $n_1 n_2 n_3 = O(S \log(K/2S)\sqrt{n})$.

We define the probability distribution $Ł(\mathcal{A})$ for any $\mathcal{A} \subset \{1, \dots, K\}$. $x \sim Ł(\mathcal{A})$ is generated from $x = \frac{u}{||u||_2}$ where $u \in \mathbb{R}^K$ is generated according to:

- For every $k \in \mathcal{A}$, we set $u_k = 0$ with probability $\frac{2S}{2S+1}$ and $u_k \sim \mathcal{N}(0,1)$ with probability $\frac{1}{2S+1}$.
- For $k \in \mathcal{A}^c$, where $\mathcal{A}^c$ is the complementary of $\mathcal{A}$, i.e. $\{1, \dots, K\} \setminus \mathcal{A}$, we set $u_k = 0$.

We then exploit the information collected in the first phase, i.e. the active set at time $n_1 n_2 n_3$, by applying the linear bandit algorithm CB_2 on the small selected subset. The pseudo-code of the algorithm is described in Figure 3.

Input: parameters S, δ.
Initialize: Set $n_1 = \lfloor \log(K/2S)(S+1) \rfloor + 1$, $n_2 = \lfloor \log(1/\delta)\exp(1) \rfloor + 1$ and $n_3 = \lfloor \log(1/\delta)\sqrt{n} \rfloor + 1$
Initialize: Set $t = 1$, $\mathcal{A}(t) = \{1, \dots, K\}$
Support exploration phase:
for $i = 0, \dots, n_1 - 1$ **do**
  $v = 0$
  for $j = 0, \dots, n_2 - 1$ **do**
    Pull randomly an arm $x \sim Ł(\mathcal{A}(t))$
    for $k = 0, \dots, n_3$ **do**
      Collect r_t with $x_t = x$
      Set $\mathcal{A}(t+1) = \mathcal{A}(t)$
      if $r_t = 0$ **then**
        $v = x_t$
      end if
      $t = t + 1$
    end for
  end for
  if $v \neq 0$ **then**
    $\mathcal{A}(t+1) = \mathcal{A}(t) \setminus \{k : v_k \neq 0\}$
  end if
end for
Restricted Linear Bandit Phase:
For $t = n_1 n_2 n_3, \dots, n$, run $CB_2(\mathcal{B}_K \cap Vec(\mathcal{A}(n_1 n_2 n_3)), \delta)$ and collect the r_t

Figure 3: The pseudo-code of the S^{2}L-UCB algorithm.

3.2 Main Result

We make a more restrictive assumption on the noise

Assumption 2 *The vector σ such that $|\eta_{k,t}| \leq \frac{1}{2}\sigma_k$ is a $S-$sparse vector.*

We provide here the expression of the regret for algorithm S^{2}L-UCB. Again, the proof of this result can be found in the Supplementary Material (Appendix B).

Theorem 3 *Under Assumption 2, and if S is an upper bound on the sparsity of θ, the regret of S^{2}L-UCB is bounded with probability at least $1 - \delta$ as*

$$R_n(\mathcal{A}lg_{S^2L-UCB}) \leq 298 S \log(16KSn^2/\delta^2)^4 (||\theta||_2 + ||\sigma||_2)\sqrt{n}. \quad (6)$$

When the noise is sparse, it is possible to retrieve the support of θ with a *number of samples of order* $O(S\sqrt{n})$ even when the noise is arbitrarily big and θ is arbitrarily small. The idea is to detect the coordinates of the space for which the projection of the vector $\theta + \eta_t$ is non-zero: note that there are at most $2S$ indexes such that the vector is non-zero. To detect the non-zero coordinates, we project on vectors x

[9] Note that $\mathcal{A}(1) = \{1, \dots, K\}$.

that contain a certain proportion of non-zero coordinates whereas the other coordinates of the vector are 0. With a non-zero probability, all the non-zero coordinates of $\theta+\eta_t$ will be at the same position as the zeros in x and we observe in those cases $r_t = \langle \theta+\eta_t, x\rangle = 0$. In this case, we can remove all the non-zero coordinates of x from the active set[10]. As we observe $r_t = 0$ with non-zero probability, we know that if we sample a large enough number of different i.i.d. x, we will receive $r_t = 0$ several times with high probability and thus remove from the active set many coordinates: at the end of the process, the size of the active set $\mathcal{A}(t)$ is smaller than a constant times S.

4 The gradient ascent as a bandit problem

The aim of this section is to propose a local optimization technique to maximize a function $f : \mathbb{R}^K \to \mathbb{R}$ *when the dimension K is very high* and *when we can only sample n times this function with $n \ll K$*. We assume that the function f depends on a small number of relevant variables: it corresponds to the assumption that the gradient of f is sparse.

A well-known local optimization technique is gradient ascent, that is to say compute the gradient $\nabla f(u)$ of f at point u, go in the direction of the gradient, and then iterate n times. See for instance the book of Bertsekas (1999) for an exhaustive survey on, among other things, gradient methods.

4.1 Formalization

The objective is to apply gradient ascent to the differentiable function f. Assume that we are allowed to do only n queries to the function. We call u_t the $t-$th point where we sample f, and choose it such that $||u_{t+1} - u_t||_2 = \epsilon$, where ϵ is the gradient step.

Note that by the theorem of intermediate values

$$\begin{aligned} f(u_n) - f(u_0) &= \sum_{t=1}^{n} f(u_t) - f(u_{t-1}) \\ &= \sum_{k} \langle (u_t - u_{t-1}), \nabla f(w_t)\rangle, \end{aligned}$$

where w_t is an appropriate barycentre of u_t and u_{t-1}.

We can thus model the problem of efficient gradient ascent by a linear bandit problem where the reward is what we gain/loose by going from point u_{t-1} to point u_t, i.e. $f(u_t) - f(u_{t-1})$. More precisely, if we want to rewrite the problem with previous notations, we would have $\theta + \eta_t = \nabla f(w_t)$[11], and $x_t = u_t - u_{t-1}$. We illustrate this modelisation in Figure 4.

If we assume that the function f is (locally) linear and that there are some i.i.d. measurement errors, we are exactly in the setting of Section 1. The objective of minimizing the regret, can then be rewritten

$$R_n(\mathcal{A}lg) = \max_{x \in \mathcal{B}_2(u_0, n\epsilon)} f(x) - f(u_n),$$

where u_n is the terminal point of algorithm $\mathcal{A}lg$. The regret would be either in $O(S\epsilon\frac{\sqrt{n}}{||\theta||_2})$ (easy problems) or $O(S\epsilon n^{2/3})$ (difficult problems) for SL-UCB. If the noise is $S-$sparse as well, the regret would be in $O(S\epsilon\sqrt{n})$ for S^{2}L-UCB.

Remark on the noise: Assumption 1, which states that the noise added to the function is of the form $\langle u_t - u_{t-1}, \eta_t\rangle$ is specially suitable for gradient ascent because it corresponds to the cases where the noise is an approximation error and depends on the gradient step.

Assumption 2, which states that the noise is sparse, is specially suitable for cases where we measure non-noisy samples of a function exactly constant in $K - S$ directions: the observation noise corresponds to the linear approximation error and is thus in the directions where the function is not constant.

Remark on the linearity assumption: Matching the stochastic bandit model in Section 1 to the problem of gradient ascent corresponds to assuming that the function is (locally) linear in a neighborhood of u_0, and that we have in this neighborhood $f(u_{t+1}) - f(u_t) = \langle u_{t+1} - u_t, \nabla f(u_0) + \eta_{t+1}\rangle$, where the noise η_{t+1} is i.i.d. This setting is very restrictive: we made it in order to offer a first, simple solution for the problem. When the function is not linear, there is an additional approximation error.

4.2 Numerical experiment

In order to illustrate the mechanism of our algorithms, we apply SL-UCB to a quadratic function in dimension 100 where only two dimensions are informative (for better visibility on a 2-D graph). Figure 5 shows the trajectory of the algorithm, projected in the subspace of dimension 2 where the function is not constant.

[10] Note however that in order to remove coordinates from the active set, we need to project many times on a given x: this is necessary in order to be sure that we do not remove by accident a coordinate where $\theta_k = -\eta_{k,t} \neq 0$.

[11] Note that in order for the model in Section 1 to hold, we need to relax the assumption that η is i.i.d..

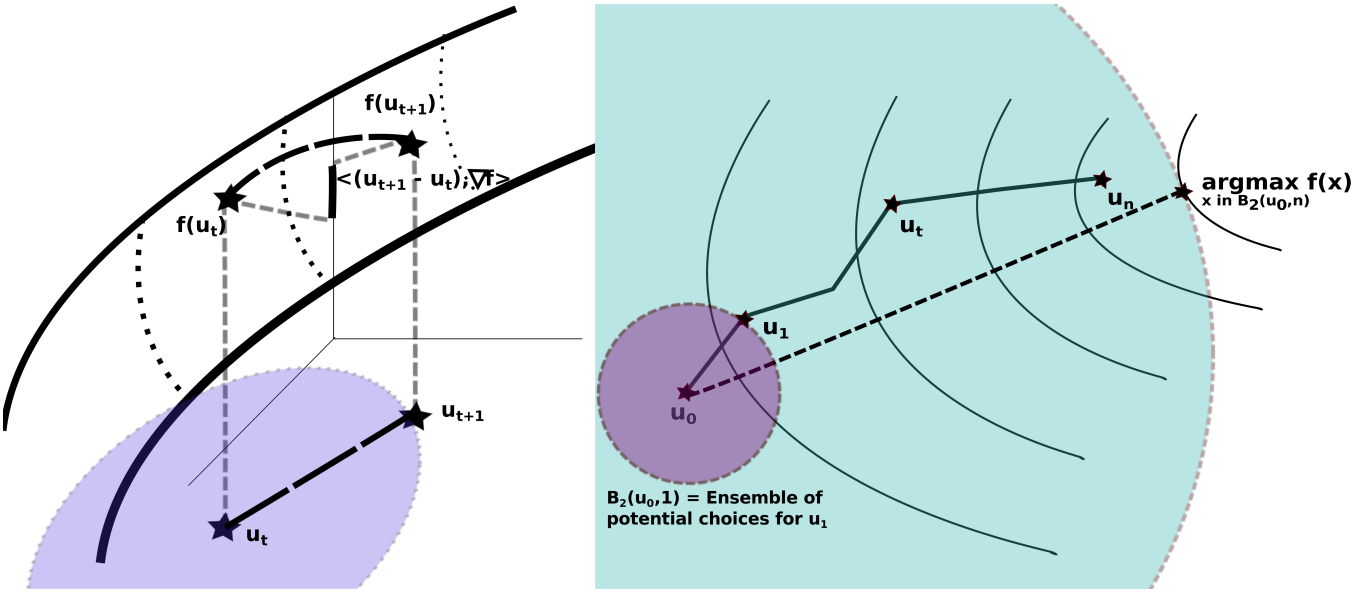

Figure 4: The gradient ascent: the first picture illustrates the problem written as a linear bandit problem with rewards and the second picture illustrates the regret.

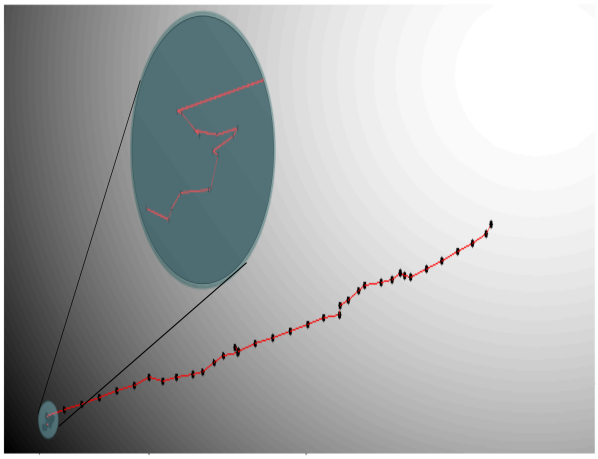

Figure 5: Trajectory of algorithm SL-UCB with a budget $n = 50$, with a zoom at the beginning of the trajectory to illustrate the support exploration phase. The levels of gray correspond to the contours of function.

Note that at the beginning of the ascent, the projection of the steps on relevant directions are very small because we search for the good support and thus also move in other directions of the space than the subspace of dimension 2 where the gradient lies. However, the algorithm quickly concentrates on the good support of the gradient.

We now want to illustrate the performances of SL-UCB and S^{2}L-UCB. We fix the number of pulls to 100, and we try different values of K, in order to have results for different values of $\frac{K}{n}$. The higher this quantity, the more difficult the problem. We choose a quadratic function varying in $S = 10$ directions[12].

We compare our two algorithms SL-UCB and S^{2}L-UCB with two strategies: the "oracle" gradient strategy (OGS), i.e. a gradient algorithm with access to the *entire* gradient of the function[13], and what we call random best direction (BRD), that is to say a strategy that, at a given point, chooses a random direction, observes the value of the function a step further in this direction, and goes to that point if the value of the function at this point is better than what it was before. We report the difference between the value at the final point of the algorithm and the value at the beginning of the algorithm, i.e. the *regret* of the algorithm. The results are available in Figure 6.

Note that the performances of our algorithms are

[12]We keep the same function when K varies. It is the quadratic function $f(x) = \sum_{k=1}^{10} -20(x_k - 25)^2$.

[13]Each of the 100 pulls corresponds to an access to the entire gradient of the function at a chosen point.

K/n	OGS	SL-UCB	S^{2}L-UCB	BRD
2	$1.875\ 10^5$	$1.723\ 10^5$	$1.823\ 10^5$	$2.934\ 10^4$
10	$1.875\ 10^5$	$1.657\ 10^5$	$1.726\ 10^5$	$1.335\ 10^4$
100	$1.875\ 10^5$	$1.552\ 10^5$	$1.648\ 10^5$	$5.675\ 10^3$

Figure 6: We report, for different values of $\frac{K}{n}$ and different strategies, the value of $f(u_n) - f(u_0)$.

worse than the "oracle" gradient strategy. This is not surprising because SL-UCB and S^{2}L-UCB are only given partial information on the gradient. However they perform way better than the random best direction. Note that the bigger $\frac{K}{n}$, the more impressive the improvements of SL-UCB and S^{2}L-UCB over the random best direction strategy. This can be explained by the fact that the larger $\frac{K}{n}$, the less probable it is that the random direction strategy picks a direction of interest, whereas our algorithms are build for dealing with such problems.

Note also that S^{2}L-UCB performs slightly better than SL-UCB in this specific case. This is not surprising as we did not add any noise to the function on the experiment, and the only perturbation to the linear gradient problem comes from the non-linearity of the function we choose: the "noise" in that case is thus sparse, and S^{2}L-UCB is built to deal with such a setting.

Conclusion

In the paper we provided two algorithms for sparse linear bandits in very high dimension. Both of them are designed using ideas from compressed sensing and bandit theory. Compressed sensing intuitions are useful in a first support exploration phase, when we want to figure out what the support of the parameter is. We then apply the linear bandit algorithm from (Dani et al., 2008) to the small dimensional subspace obtained from the extracted support of the parameter. The algorithm SL-UCB provides a regret (i) of order $O(Sn^{2/3})$ if the problem is *difficult*, i.e. $||\theta||_2$ is small and (ii) of order $O(S\sqrt{n})$ if the problem is *simple*, i.e. $||\theta||_2$ is big. Note that there is a gap with respect to n between those two rates. However, by making the additional assumption that the noise η is sparse, we derived an algorithm called S^{2}L-UCB whose regret is of order $O(S\sqrt{n})$. Note that all our bounds scale with the sparsity S of the unknown parameter θ instead of the dimension K of the parameter (as is usually the case in linear bandits). We then provided an example of application for this setting, the optimization of a function in very high dimension.

There are to our minds three main directions for further researches.

- An interesting direction of research is to deal with the case when θ is not fixed. It is probably impossible to achieve a sub linear regret in the general case. But it would be nice to have some assumptions on the way it has to change so that we are able to achieve sub linear regret. One idea would be to use techniques developed for *adversarial bandits* (see (Abernethy et al., 2008; Bartlett et al., 2008; Cesa-Bianchi and Lugosi, 2009; Koolen et al., 2010; Audibert et al., 2011), but also (Flaxman et al., 2005) for a more gradient-specific modeling) or also from *restless/switching bandits* (see e.g. (Whittle, 1988; Nino-Mora, 2001; Slivkins and Upfal, 2008; A. Garivier, 2011) and many others). This would be particularly interesting to model gradient ascent on e.g. convex function where the gradient evolves (but not too much) in space.

- Finally, what we believe to be very interesting is to investigate in more detail the complementarity of compressed sensing and bandit theory for retrieving information. It is in our opinion very relevant to use the methods at the same time, because while compressed sensing samples the space in an unadaptive way to find relevant features, bandits focus on those parts of the space that seem interesting and exploit them in a good way. This direction would deserve further investigations.

References

E. Moulines A. Garivier. On upper-confidence bound policies for non-stationary bandit problems. In *Algorithmic Learning Theory (ALT)*, 2011.

J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, volume 3. Citeseer, 2008.

J.Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. *Arxiv preprint arXiv:1105.4871*, 2011.

P.L. Bartlett, V. Dani, T. Hayes, S.M. Kakade, A. Rakhlin, and A. Tewari. High-probability regret bounds for bandit online linear optimization. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008)*, pages 335–342. Citeseer, 2008.

D.P. Bertsekas. *Nonlinear programming*. Athena Scientific Belmont, MA, 1999.

T. Blumensath and M.E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT 09)*. Citeseer, 2009.

S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1999.

V. Dani, T.P. Hayes, and S.M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*. Citeseer, 2008.

S. Filippi, O. Cappé, A. Garivier, and C. Szepesvári. Parametric bandits: The generalized linear case. 2010.

A.D. Flaxman, A.T. Kalai, and H.B. McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

W.M. Koolen, M.K. Warmuth, and J. Kivinen. Hedging structured concepts. 2010.

J. Nino-Mora. Restless bandits, partial conservation laws and indexability. *Advances in Applied Probability*, 33(1):76–98, 2001.

P. Rusmevichientong and J.N. Tsitsiklis. Linearly parameterized bandits. *Arxiv preprint arXiv:0812.3465*, 2008.

A. Slivkins and E. Upfal. Adapting to a changing environment: The brownian restless bandits. In *Proc. 21st Annual Conference on Learning Theory*, pages 343–354. Citeseer, 2008.

P. Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, pages 287–298, 1988.

C. Szepesvari Y. Abbasi-yadkori, D. Pal. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, 2011.

Supplementary Material: Bandit Theory meets Compressed Sensing for high dimensional Stochastic Linear Bandit

A Analysis of algorithm SL-UCB

A.1 Expression of the regret of the algorithm

If we can find an event ξ of probability at least $1-p_\delta$ such that on ξ, the support exploration phase lasts at most T times and selects an active set which contains at least the elements in $\mathcal{A}$ and at most d elements, and then plays $ConfidenceBall_2(\mathcal{B}_K \cap \mathcal{A}, p_\delta)$ on that set, we can bound the regret of this algorithm with probability $1-2p_\delta$ by

$$R_n \leq 2T||\theta||_2 + \tilde{R}_n(\mathcal{A}, p_\delta) + n\Big(\max_{x\in\mathcal{B}_K}\langle x,\theta\rangle - \max_{x\in\mathcal{B}_K\cap Vect(\mathcal{A})}\langle x,\theta\rangle\Big), \quad (7)$$

where $\tilde{R}_n(\mathcal{A}, p_\delta)$ is the regret of $ConfidenceBall_2(\mathcal{B}_K \cap \mathcal{A}, p_\delta)$, and if the set $\mathcal{A}$ contains at most d elements, then $\tilde{R}_n(\mathcal{A}, p_\delta) \leq 64d\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n}$.

A.2 Expression of the θ_t in order to create an event ξ of interest

Note that as $x_{k,t} = \frac{1}{\sqrt{K}}$ or $x_{k,t} = -\frac{1}{\sqrt{K}}$, we have

$$\begin{aligned}\hat{\theta}_{k,T} &= \frac{K}{T}\Big(\sum_{t=1}^{T} x_{k,t} r_t\Big)\\ &= \frac{K}{T}\Big(\sum_{t=1}^{T} x_{k,t}\sum_{k'=1}^{K} x_{k',t}(\theta_{k'}+\eta_{k',t})\Big)\\ &= \frac{K}{T}\sum_{t=1}^{T} x_{k,t}^2\theta_k + \frac{K}{T}\sum_{t=1}^{T} x_{k,t}\sum_{k'\neq k} x_{k',t}\theta_{k'}\\ &+ \frac{K}{T}\sum_{t=1}^{T} x_{k,t}\sum_{k'=1}^{K} x_{k',t}\eta_{k',t}\\ &= \theta_k + \frac{1}{T}\sum_{t=1}^{T}\sum_{k'\neq k} b_{k,k',t}\theta_{k'} + \frac{1}{T}\sum_{t=1}^{T}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t},\end{aligned}$$

where $b_{k,k',t} = K x_{k,t}x_{k',t}$.

Note that as the $x_{k,t}$ are i.i.d. random variables such that $x_{k,t} = \frac{1}{\sqrt{K}}$ with probability $1/2$ and $x_{k,t} = -\frac{1}{\sqrt{K}}$ with probability $1/2$, the $(b_{k,k',t})_{k'\neq k,t}$ are i.i.d. rademacher random variables, and $b_{k,k,t} = 1$.

Step 1: Study of the first term. Let us first study $\frac{1}{T}\sum_{t=1}^{T}\sum_{k'\neq k} b_{k,k',t}\theta_{k'}$.

Note that the $b_{k,k',t}\theta_{k'}$ are $(K-1)T$ zero-mean random independent variables and that among them, $\forall k' \in \{1,...,K\}$, T of them are bounded by $\theta_{k'}$, i.e. the $(b_{k,k',t}\theta_{k'})_t$.

By Hoeffding, we thus have with probability $1-\delta$

$$|\frac{1}{T}\sum_{t=1}^{T}\sum_{k'\neq k}^{K} b_{k,k',t}\theta_{k'}| \leq \frac{||\theta||_2\sqrt{2\log(2/\delta)}}{\sqrt{T}}.$$

Now by doing a union bound on all the $k = \{1,\dots,K\}$, we have that with probability $1-\delta$, $\forall k$,

$$|\frac{1}{T}\sum_{t=1}^{T}\sum_{k'\neq k} b_{k,k',t}\theta_{k'}| \leq \frac{||\theta||_2\sqrt{2\log(2K/\delta)}}{\sqrt{T}}. \quad (8)$$

Step 2: Study of the second term. Let us now study $\frac{1}{T}\sum_{t=1}^{T}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t}$.

Note that the $(b_{k,k',t}\eta_{k',t})_{k',t}$ are KT independent zero-mean random variables, and that among these variables, $\forall k \in \{1,...,K\}$, T of them are bounded by $\frac{1}{2}\sigma_k$. By Hoeffding, we thus have with probability $1-\delta$

$$|\frac{1}{T}\sum_{t=1}^{T}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t}| \leq \frac{||\sigma||_2\sqrt{2\log(2/\delta)}}{\sqrt{T}}.$$

Now by doing a union bound on all the k, we have that with probability $1-\delta$, $\forall k$,

$$|\frac{1}{T}\sum_{t=1}^{T}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t}| \leq \frac{||\sigma||_2\sqrt{2\log(2K/\delta)}}{\sqrt{T}}. \quad (9)$$

Step 3: Global bound Finally for a given T, with probability $1-2\delta$, $\forall k = \{1,\dots,K\}$, we have by Equations 8 and 9

$$|\hat{\theta}_{k,T} - \theta_k| \leq \frac{(||\theta||_2 + ||\sigma||_2)\sqrt{2\log(2K/\delta)}}{\sqrt{T}}, \quad (10)$$

which is equivalent, with probability $1-2\delta$, to

$$||\hat{\theta}_T - \theta||_\infty \leq \frac{(||\theta||_2 + ||\sigma||_2)\sqrt{2\log(2K/\delta)}}{\sqrt{T}}$$

Step 4: Definition of the event of interest Now we consider the event ξ such that

$$\xi = \left\{\omega \in \Omega / ||\theta - \frac{K}{\sqrt{n}} X_{\sqrt{n}} R_{\sqrt{n}}||_\infty \leq \frac{b}{n^{1/4}}\right\} \cup \left\{\omega \in \Omega / ||\theta - \frac{K}{n^{2/3}} X_{n^{2/3}} R_{n^{2/3}}||_\infty \leq \frac{b}{n^{1/3}}\right\}, \quad (11)$$

where $b = (\bar{\theta}_2 + \bar{\sigma}_2)\sqrt{2\log(2K/\delta)}$.

Now by just using Equation 10 and a simple union bound, we have that $\mathbb{P}(\xi) \geq 1 - 4\delta$.

A.3 A first problem dependent bound when $||\theta||_\infty$ is big.

We assume in the sequel that $||\theta||_\infty \geq \frac{2b}{n^{1/6}}$, i.e. $\max_k |\theta_k| \geq \frac{2b}{n^{1/6}}$.

Step 1: The support exploration phase stops after $\sqrt{n}$ time steps. Let us call $k^* = \arg\max_k \theta_k$.

On the event ξ, we know that for $t = \sqrt{n}$,

$$|\hat{\theta}_{k^*,\sqrt{n}} - \theta_{k^*}| \leq \frac{b}{n^{1/4}}.$$

This means that, on ξ,

$$|\hat{\theta}_{k^*,\sqrt{n}}| \geq |\theta_{k^*}| - \frac{b}{n^{1/4}} \geq \frac{2b}{n^{1/6}} - \frac{b}{n^{1/4}} \geq \frac{b}{n^{1/6}}.$$

This means that on ξ, the algorithm will stop pulling randomly arms in $\mathcal{E}_{xploring}$ and start using the linear bandit algorithm after $\sqrt{n}$ time steps.

Step 2: Description of the set $\mathcal{A}$ The set $\mathcal{A}$ is defined as $\mathcal{A} = \left\{k : |\hat{\theta}_{k,\sqrt{n}}| \geq \frac{2b}{n^{1/4}}\right\}$.

Let us consider an arm k such that $|\theta_k| \geq \frac{3b}{n^{1/4}}$.

Then on ξ, we know that

$$|\hat{\theta}_{k,\sqrt{n}}| \geq |\theta_k| - \frac{b}{n^{1/4}} \geq \frac{3b}{n^{1/4}} - \frac{b}{n^{1/4}} \geq \frac{2b}{n^{1/4}}.$$

This means that $k \in \mathcal{A}$ on ξ.

Now let us consider an arm k such that $|\theta_k| < \frac{b}{n^{1/4}}$

Then on ξ, we know that

$$|\hat{\theta}_{k,\sqrt{n}}| < |\theta_k| + \frac{b}{n^{1/4}} < \frac{b}{n^{1/4}} + \frac{b}{n^{1/4}} < \frac{2b}{n^{1/4}}.$$

This means that $k \in \mathcal{A}^c$ on ξ.

Finally, the only θ_k we do not know for sure if or if not they belong to $\mathcal{A}$ on ξ are the ones such that $\frac{b}{n^{1/4}} < |\theta_k| \leq \frac{3b}{n^{1/4}}$. Note that this means also that $|\mathcal{A}| \leq S$ because θ is $S-$sparse.

Step 3: Comparison of the best element on $\mathcal{A}$ and on $\mathcal{B}_K$. Now let us compare $\max_{x_t \in Vec(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle$ and $\max_{x_t \in \mathcal{B}_K} \langle \theta, x_t \rangle$.

At first, note that $\max_{x_t \in \mathcal{B}_K} \langle \theta, x_t \rangle = ||\theta||_2$ and that $\max_{x_t \in Vec(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle = ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2$.

This means that

$$\begin{aligned}
&\max_{x_t \in \mathcal{B}_K} \langle \theta, x_t \rangle - \max_{x_t \in Vec(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle \\
&= ||\theta||_2 - ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2 \\
&= \frac{||\theta||_2^2 - ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2^2}{||\theta||_2 + ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2} \\
&\leq \frac{\sum_{k \in \mathcal{A}^c} \theta_k^2}{||\theta||_2}.
\end{aligned}$$

We now use the results of the Step 2, that is to say that on ξ, if k is in $\mathcal{A}^c$, then $|\theta_k| < \frac{3b}{n^{1/4}}$. As the vector is S sparse, the last equation gives us

$$\max_{x_t \in \mathcal{B}_K} \langle \theta, x_t \rangle - \max_{x_t \in Vec(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle \leq \frac{9Sb^2}{||\theta||_2 \sqrt{n}}. \quad (12)$$

Step 4: Expression of the regret By using Equation 16, the length of the support exploration phase deduced in Step 1, and Equation 12, we obtain

$$
\begin{aligned}
R_n &\leq 2||\theta||_2\sqrt{n} + \tilde{R}_n(\mathcal{A},\delta) + n\frac{9Sb^2}{||\theta||_2\sqrt{n}} \\
&\leq \Big(66S\big(||\theta||_2 + ||\sigma||_2\big)(\log(n^2/\delta))^2 + \frac{9Sb^2}{||\theta||_2}\Big)\sqrt{n} \\
&\leq 84S\big(||\theta||_2 + ||\sigma||_2 + \frac{(\bar{\theta}_2+\bar{\sigma}_2)^2}{||\theta||_2}\big) \\
&\times (\log(2n^2K/\delta))^2\sqrt{n} \\
&\leq 84S\big(2\bar{\theta}_2 + 3\bar{\sigma}_2 + \frac{\bar{\sigma}_2^2}{||\theta||_2}\big)(\log(2n^2K/\delta))^2\sqrt{n} \\
&\leq 168S\big(\sqrt{\bar{\theta}_2} + \frac{\bar{\sigma}_2}{\sqrt{||\theta||_2}}\big)^2(\log(2n^2K/\delta))^2\sqrt{n} \\
&\leq O(\frac{\sqrt{n}S}{||\theta||_2}),
\end{aligned}
$$

by using $b = (\bar{\theta}_2+\bar{\sigma}_2)\sqrt{2\log(2K/\delta)}$ for the third step.

A.4 A problem independent bound when $||\theta||_\infty$ is small.

We assume in the sequel that $||\theta||_\infty \leq \frac{b}{10n^{1/6}}$, i.e. $\max_k |\theta_k| \leq \frac{b}{10n^{1/6}}$.

Step 1: The support exploration phase stops after $n^{2/3}$ time steps. On the event ξ, we know that for $t = \sqrt{n}$,

$$
|\hat{\theta}_{k,\sqrt{n}} - \theta_k| \leq \frac{b}{n^{1/4}}.
$$

This means that

$$
|\hat{\theta}_{k,\sqrt{n}}| \leq \max_k |\theta_k| + \frac{b}{n^{1/4}} \leq \frac{b}{10n^{1/6}} + \frac{b}{n^{1/4}} < \frac{b}{4n^{1/6}}.
$$

if $n \geq 4$, because of then $\frac{b}{n^{1/4}} \leq \frac{9b}{10n^{1/6}}$.

This means that on ξ, the algorithm will continue pulling randomly arms in $\mathcal{E}_{xploring}$ until $t = n^{2/3}$.

Step 2: Description of the set $\mathcal{A}$ The set $\mathcal{A}$ is then defined as $\mathcal{A} = \Big\{k : |\hat{\theta}_{k,n^{2/3}}| \geq \frac{2b}{n^{1/3}}\Big\}$.

Let us consider an arm k such that $|\theta_k| \geq \frac{3b}{n^{1/3}}$.

Then on ξ, we know that

$$
|\hat{\theta}_{k,n^{2/3}}| \geq |\theta_k| - \frac{b}{n^{1/3}} \geq \frac{3b}{n^{1/3}} - \frac{b}{n^{1/3}} \geq \frac{2b}{n^{1/3}}.
$$

This means that $k \in \mathcal{A}$ on ξ.

Now let us consider an arm k such that $|\theta_k| < \frac{b}{n^{1/3}}$

Then on ξ, we know that

$$
|\hat{\theta}_{k,n^{2/3}}| < |\theta_k| + \frac{b}{n^{1/3}} < \frac{b}{n^{1/3}} + \frac{b}{n^{1/3}} < \frac{2b}{n^{1/3}}.
$$

This means that $k \in \mathcal{A}^c$ on ξ.

Finally, the only θ_k we do not know for sure if or if not they belong to $\mathcal{A}$ on ξ are the ones such that $\frac{b}{n^{1/3}} < |\theta_k| \leq \frac{3b}{n^{1/3}}$. Note that this means also that $|\mathcal{A}| \leq S$ because of θ is $S-$sparse.

Step 3: Comparison of the best element on $\mathcal{A} \cap \mathcal{B}_K$ and on $\mathcal{B}_K$. Now let us compare $\max_{x_t \in Vect(\mathcal{A})\cap\mathcal{B}_K}\langle\theta, x_t\rangle$ and $\max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle$.

At first, note that $\max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle = ||\theta||_2$ and that $\max_{x_t \in Vect(\mathcal{A})\cap\mathcal{B}_K}\langle\theta,x_t\rangle = ||\theta\mathbb{I}\{k\in\mathcal{A}\}||_2$.

This means that

$$
\begin{aligned}
&\max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle - \max_{x_t\in Vect(\mathcal{A})\cap\mathcal{B}_K}\langle\theta,x_t\rangle \\
&= ||\theta||_2 - ||\theta\mathbb{I}\{k\in\mathcal{A}\}||_2 \\
&= \frac{||\theta||_2^2 - ||\theta\mathbb{I}\{k\in\mathcal{A}\}||_2^2}{||\theta||_2 + ||\theta\mathbb{I}\{k\in\mathcal{A}\}||_2} \\
&\leq \frac{\sum_{k\in\mathcal{A}^c}\theta_k^2}{||\theta||_2}.
\end{aligned}
$$

We now use the results of the Step 2, that is to say that on ξ, if k is in $\mathcal{A}^c$, then $|\theta_k| < \frac{3b}{n^{1/3}}$. As the vector is S sparse, the last equation gives us

$$
\begin{aligned}
&\max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle - \max_{x_t\in Vect(\mathcal{A})\cap\mathcal{B}_K}\langle\theta,x_t\rangle \\
&\leq \frac{9Sb^2}{||\theta||_2 n^{2/3}}.
\end{aligned} \tag{13}
$$

Now note also that we trivially have

$$
\begin{aligned}
&\max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle - \max_{x_t\in Vect(\mathcal{A})\cap\mathcal{B}_K}\langle\theta,x_t\rangle \\
&\leq \max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle = ||\theta||_2.
\end{aligned} \tag{14}
$$

Finally, we have by combining Equations 13 and 14 that

$$
\begin{aligned}
&\max_{x_t\in\mathcal{B}_K}\langle\theta,x_t\rangle - \max_{x_t\in Vect(\mathcal{A})\cap\mathcal{B}_K}\langle\theta,x_t\rangle \\
&\leq \min\Big(||\theta||_2, \frac{9Sb^2}{||\theta||_2 n^{2/3}}\Big).
\end{aligned}
$$

If we consider the worst case when $||\theta||_2 = \frac{1}{3b\sqrt{S}n^{1/3}}$, we get

$$\max_{x_t \in \mathcal{B}_K} \langle \theta, x_t \rangle - \max_{x_t \in Vect(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle \leq \frac{3b\sqrt{S}}{n^{1/3}}. \quad (15)$$

Step 4: Expression of the regret By using the Equation 16, the support exploration phase length obtained in Step 1 and Equation 15 we obtain

$$\begin{aligned} R_n &\leq 2||\theta||_2 n^{2/3} + \tilde{R}_n(\mathcal{A}, \delta) + n\frac{3b\sqrt{S}}{n^{1/3}} \\ &\leq \Big(66S\big(||\theta||_2 + ||\sigma||_2\big)(\log(n^2/\delta))^2 + 3b\sqrt{S}\Big)n^{2/3} \\ &\leq 69S\big(\bar{\theta}_2 + \bar{\sigma}_2\big)(\log(2Kn^2/\delta))^2 n^{2/3} \\ &\leq O(n^{2/3}S), \end{aligned}$$

by using $b = (\bar{\theta}_2 + \bar{\sigma}_2)\sqrt{2\log(2K/\delta)}$ for the third step.

A.5 And when $||\theta||_\infty$ is middle

We assume now that $\frac{b}{10n^{1/6}} \leq \max_k |\theta_k| \leq \frac{2b}{n^{1/6}}$.

In that case, we are not sure whether the algorithm will pull arms in $\mathcal{E}_{xploring}$ for $\sqrt{n}$ or $n^{2/3}$ time steps. But the regret will anyways be smaller than what happens in the worst case, i.e.

$$\begin{aligned} R_n \leq & \\ &\max\Bigg[168S\big(\sqrt{\bar{\theta}_2} + \frac{\bar{\sigma}_2}{\sqrt{||\theta||_2}}\big)^2(\log(2n^2K/\delta))^2\sqrt{n} \\ &, 69S\big(\bar{\theta}_2 + \bar{\sigma}_2\big)(\log(2Kn^2/\delta))^2 n^{2/3}\Bigg] \\ &\leq 336S\big(\bar{\theta}_2 + 10(\bar{\sigma}_2 + 1)^2\big)(\log(2n^2K/\delta))^2 n^{2/3} \\ &\leq O(n^{2/3}S), \end{aligned}$$

by using Step 4 of, respectively, A.3 and A.4 for the first inequality.

$b = (\bar{\theta}_2 + \bar{\sigma}_2)\sqrt{2\log(2K/\delta)}$

A.6 Rewriting the results with the $||.||_2$ norm

Note first that as the vector θ is $S-$sparse, we have $\frac{||\theta||_2}{\sqrt{S}} \leq ||\theta||_\infty \leq ||\theta||_2$.

This means that $||\theta||_\infty \geq \frac{2b}{n^{1/6}}$ is implied by $\frac{||\theta||_2}{\sqrt{S}} \geq \frac{2b}{n^{1/6}}$. Similarly, $||\theta||_\infty \leq \frac{2b}{n^{1/6}}$ is implied by $||\theta||_2 \leq \frac{2b}{n^{1/6}}$.

We thus have that when the problem is *easy*, i.e. when $||\theta||_2 \geq \frac{2b\sqrt{S}}{n^{1/6}}$

$$\begin{aligned} &R_n(\mathcal{A}lg_{SL-UCB}) \\ &\leq 168S\big(\sqrt{\bar{\theta}_2} + \frac{\bar{\sigma}_2}{\sqrt{||\theta||_2}}\big)^2(\log(2n^2K/\delta))^2\sqrt{n}. \end{aligned}$$

And when the problem is difficult, i.e. $||\theta||_2 \leq \frac{2b}{n^{1/6}}$,

$$\begin{aligned} &R_n(\mathcal{A}lg_{SL-UCB}) \\ &\leq 69S\big(\bar{\theta}_2 + \bar{\sigma}_2\big)(\log(2Kn^2/\delta))^2 n^{2/3}. \end{aligned}$$

For the junction between the two, we take the maximum of those two bounds, i.e. when $\frac{2b\sqrt{S}}{n^{1/6}} \leq ||\theta||_2 \leq \frac{2b}{n^{1/6}}$

$$\begin{aligned} &R_n(\mathcal{A}lg_{SL-UCB}) \\ &\leq 336S\big(\bar{\theta}_2 + 10(\bar{\sigma}_2 + 1)^2\big)(\log(2n^2K/\delta))^2 n^{2/3}. \end{aligned}$$

This leads to the desired regret.

B Analysis of S^{2}L-UCB

B.1 Some additional notations

Let us denote by $Supp(\theta) = \{k : \theta_k \neq 0\} \cup \{k : \sigma_k \neq 0\}$. Note that $|Supp(\theta)| \leq 2S$.

Let us now call $p_k = \mathbb{P}_{\eta_{k,t}}(\theta_k + \eta_{k,t} \neq 0)$.

We also note $\widetilde{Supp}(\theta) = Supp(\theta) \cap \{k : p_k \geq \frac{1}{\sqrt{n}}\}$.

Let us write $Supp(x) = \{k : x_k \neq 0\}$.

B.2 Expression of the regret of the algorithm

If we can find an event ξ of probability at least $1 - p_\delta$ such that on ξ, the support exploration phase lasts at most T times and selects an active set which contains at least the elements in $\mathcal{A}$ and at most d elements, and then plays $ConfidenceBall_2(\mathcal{B}_K \cap \mathcal{A}, p_\delta)$ on that set, we can bound the regret of this algorithm with probability $1 - 2p_\delta$ by

$$\begin{aligned} R_n \leq\ & 2T||\theta||_2 + \tilde{R}_n(\mathcal{A}, p_\delta) \\ &+ n\Big(\max_{x \in \mathcal{B}_K} \langle x, \theta \rangle - \max_{x \in \mathcal{B}_K \cap Vect(\mathcal{A})} \langle x, \theta \rangle\Big), \end{aligned} \quad (16)$$

where $\tilde{R}_n(\mathcal{A}, p_\delta)$ is the regret of $ConfidenceBall_2(\mathcal{B}_K \cap \mathcal{A}, p_\delta)$, and if the

set $\mathcal{A}$ contains at most d elements, then $\tilde{R}_n(\mathcal{A}, p_\delta) \leq 64d\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n}$.

B.3 Probability of observing $r_t = 0$ when $Supp(\theta) \cap Supp(x) \neq \emptyset$

Let us assume that we are at time t and in the support exploration phase ($t \leq n_1 n_2 n_3$).

Let us assume that we pulled an arm x from $\mathcal{E}_{xploring}(t)$. Note that the algorithm will pull this arm n_3 times.

At first, note that as the $(x_k)_{k \in Supp(x)}$ are $|Supp(x)|$ i.i.d. gaussians and as the other x_k are equal to 0, we have

$$\begin{aligned}\mathbb{P}_x(r_t = 0) &= \mathbb{P}_x(\sum_{k=1}^{K} x_k(\theta_k + \eta_{k,t}) = 0) \\ &= \mathbb{P}_x(\sum_{k \in Supp(x)} x_k(\theta_k + \eta_{k,t}) = 0) = 0,\end{aligned} \tag{17}$$

if at least one of the components $(\theta_k + \eta_{k,t})_{k \in Supp(x)}$ is not 0.

Let us assume that $Supp(\theta) \cap Supp(x) \neq \emptyset$. It means that there is (at least) a k such that $\theta_k \neq 0$ or $\sigma_k \neq 0$, and $x_k \neq 0$.

Because of Equation 17, we have

$$\mathbb{P}_{x/Supp(\theta)\cap Supp(x)\neq\emptyset,\eta}(r_t = 0) \leq \mathbb{P}_\eta(\eta_k + \theta_k = 0) = 1 - p_k.$$

Now note that the algorithm pulls arm k (a bit) more than $n_3 = \log(1/\delta)\sqrt{n}$ times.

If $p_k \geq \frac{1}{\sqrt{n}}$, the probability P_1 of observing $r_t = 0$ all those n_3 times is

$$P_1 \leq (1-p)^{n_3} \leq (1-p)^{\log(1/\delta)\sqrt{n}} \leq \delta \tag{18}$$

because $(1 + \frac{x}{n})^n \leq \exp(x)$.

This means by just doing an union bound over the $n_1 n_2$ times where a different x is chosen that with probability at least $1 - n_1 n_2 \delta$, if for a chosen x we have $\max_{k \in Supp(x)} |\theta_k| + \sigma_k \neq 0$, we do not observe for this x $r_t = 0$ all the n_3 times.

Now if $p_k = \mathbb{P}_\eta(\theta_k + \eta_{k,t} = 0) \leq \frac{1}{\sqrt{n}}$, it means that $\theta_k = \mathbb{E}(\theta_k + \eta_{k,t}) \leq \frac{|\theta_k| + \sigma_k}{\sqrt{n}} \leq \frac{||\theta||_2 + ||\sigma||_2}{\sqrt{n}}$. Even if we have a non-neglectible probability of removing index k from the active set, it is not a problem because of θ_k is very small.

Finally, if we pick x such that $k \in \widetilde{Supp}(\theta) \cap Supp(x)$, we do not observe $r_t = 0$ for all n_3 pulls of this x with probability at least $1 - n_1 n_2 \delta$.

B.4 Probability of choosing a x such that $Supp(x) \cap Supp(\theta) = \emptyset$.

Let us assume that we are at time t and in the support exploration phase ($t \leq n_1 n_2 n_3$).

Let us assume that $|\mathcal{A}(t)| = k$. This means that the probability of choosing x such that $Supp(x) \cap Supp(\theta) = \emptyset$ is $(\frac{2S}{2S+1})^{2S} \geq \frac{2S+1}{2S}\exp(-1)(1 - \frac{1}{2(S+1)}) \geq e^{-1}$, because $(1 + \frac{x}{n})^n \geq \exp(x)(1 - \frac{x^2}{2n})$.

Note that we pick n_2 different vectors in $\mathcal{E}_{xploring}(t)$. The probability P_2 that none of those n_2 vectors are such that $Supp(x) \cap Supp(\theta) = \emptyset$ is such as

$$P_2 \leq (1 - \exp(-1))^{n_2} \leq \delta. \tag{19}$$

because $(1 + \frac{x}{n})^n \leq \exp(x)$. This means by just doing a union bound over the n_1 times where the support is updated that with probability at least $1 - n_1\delta$, we will pull an arm x at each phase such that $Supp(x) \cap Supp(\theta) = \emptyset$.

B.5 Probability of picking the good support at the end

B.3 tells us that with probability at least $1 - n_1 n_2 \delta$ if we choose an x such that $\widetilde{Supp(x)} \cap Supp(\theta) \neq \emptyset$, then we observe at least a $r_t \neq 0$ among the n_3 times we pull this x.

B.4 tells us that with probability $1 - n_1\delta$, we will pull a $x \in \mathcal{E}_{xploring}(t)$ such that $Supp(x) \cap Supp(\theta) = \emptyset$.

Combining those two results allows us to state that for each of the n_1 phases where we are allowed to change the active set, with probability at least $1 - n_1\delta - n_1 n_2 \delta \geq 1 - 2n_1 n_2 \delta$, we will remove from the active set the indexes of one vector x among the n_2 different vector that were picked before changing set $\mathcal{A}(t)$, such that $Supp(x) \cap \widetilde{Supp}(\theta) = \emptyset$.

We know that we have n_1 trials and that during each of those trials, there is at least one x such that $Supp(x) \cap \widetilde{Supp}(\theta) = \emptyset$. We know also that we erase all the non 0 of the so selected vectors x from the active set. The probability that we do not erase an index from the active set is thus the probability that this index is 0 in all the n_1 instances of x, that is to say $(\frac{S}{S+1})^{n_1}$. The number of indexes N_{n_1} that are remaining in the active set is thus the sum of K Bernouil-

lis of parameter $(\frac{S}{S+1})^{n_1}$. Note that the means of it is $K(\frac{S}{S+1})^{n_1} \leq 2S$ because $n_1 \geq (S+1)\log(K/2S)$. We can now apply Bernstein inequality (for variable bounded by 1) to obtain that with probability $1-\delta$, we have $N_{n_1} - 2S \leq \sqrt{4S\log(1/\delta)} + \frac{\log(1/\delta)}{3}$, because $\mathbb{V}(N_{n_1}) = K(\frac{S}{S+1})^{n_1}(1-(\frac{S}{S+1})^{n_1}) \leq 2S$.

This means that at the end, with probability $1 - 2n_1n_2\delta - \delta$, the active set is such that $\widetilde{Supp}(\theta) \subset \mathcal{A}_{n_1n_2n_3}$ and that $|A_{n_1n_2n_3}| = N_{n_1} \leq 2S + 2\sqrt{S\log(1/\delta)} + \frac{\log(1/\delta)}{3}$.

B.6 Regret

Let us pose $S' = 2S + 2\sqrt{S\log(1/\delta)} + \frac{\log(1/\delta)}{3}$ that is to say the upper bound in high probability on the size of the active set. We have with probability $1-2n_1n_2\delta-2\delta$

$$\begin{aligned}
R_n &\leq n_1n_2n_3(2||\theta||_2) \\
&\quad + 64S'\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n} \\
&\quad + 2S(||\theta||_2 + ||\sigma||_2)\sqrt{n},
\end{aligned}$$

where the first term is the what we lose during the support exploration phase, the second term is the regret of $CB_2(\widetilde{Supp}(\theta), \delta)$ and the last term is the loss incurred for not taking into account the small elements of $Supp(\theta) - \widetilde{Supp}(\theta)$.

Finally, we have with probability at least $1 - 4n_1n_2\delta$

$$\begin{aligned}
R_n &\leq 16\log(1/\delta)^2\log(K/S)S||\theta||_2\sqrt{n} \\
&\quad + 280S\log(1/\delta)\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n} \\
&\quad + 2S(||\theta||_2 + ||\sigma||_2)\sqrt{n} \\
&\quad \leq 298S\log(n^2/\delta)^3\log(K/S)(||\theta||_2 + ||\sigma||_2)\sqrt{n}.
\end{aligned}$$

By simplifying a bit more, we obtain with probability at least $1-\delta$

$$R_n \leq 298S\log(16KSn^2/\delta^2)^4(||\theta||_2 + ||\sigma||_2)\sqrt{n}.$$