Bandit Theory meets Compressed Sensing for high dimensional Stochastic Linear Bandit

Alexandra Carpentier
Sequel team, INRIA Lille - Nord Europe

Remi Munos
Sequel team, 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 an algorithm with a regret bound 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 a linear stochastic bandit problem in 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 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 $||.||_2$ norm, induced by the inner product.

Bandit Theory meets Compressed Sensing Our problem asks in an urging way the fundamental question at the heart of bandit theory, namely the exploration[1] versus exploitation[2] 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 thus possible to explore efficiently. In our setting we assume that $K \gg n$ and it is thus 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 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 exploita-

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

[1] It is important to explore the space in order to build a good estimate of all components of θ in order to know which arms are the best ones.

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

tion. 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 Abbasi-yadkori et al. (2011).

Our contributions are the following.

- We provide an algorithm that mixes ideas of Compressed Sensing and Bandit Theory for solving the exposed problem. It has a regret[3] which is of order $O(S\sqrt{n})$[4].
- 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 algorithm provides an efficient way for solving the problem of high dimensional gradient ascent on functions that depend only on a small number of relevant variables.

A similar setting is also studied in the paper (Abbasi-yadkori et al., 2012), published simultaneously. Their assumption on the noise is however radically different: unlike in our setting they consider a noise to signal ratio that is different depending on whether the direction where they sample is flat or not. It is thus difficult to compare the results.

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 algorithm we propose for this problem, and give the main results in Section 2. We detail in Section 3 the application to gradient ascent and provide numerical experiments.

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[5] 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 is close in high probability. For example if we assume that each $\eta_{k,t}$ is bounded by $\frac{1}{2}\sigma_k$, we know by Azuma inequality (because x_t depends only of $(\eta_{k,s})_{s \leq t}$) 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, 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 $S < n$.

1.2 A useful algorithm for Linear Bandits

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

This algorithm is designed for stochastic linear bandit in dimension d, i.e. the bandit parameter θ is in $\mathbb{R}^d$, and d is *smaller* than the budget n. This is the reason

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

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

[5] 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..

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

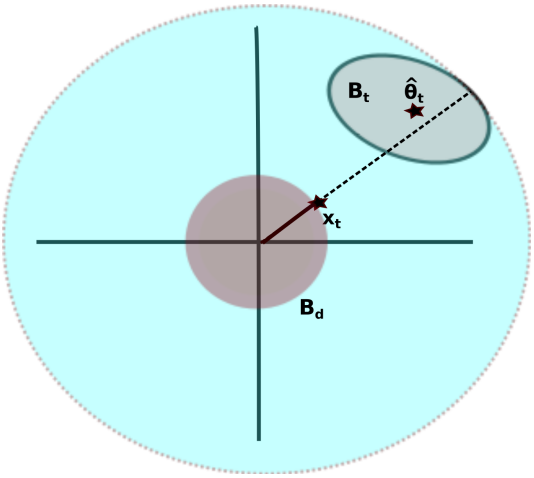

Figure 1: Algorithm $ConfidenceBall_2$ (CB_2) adapted for an action set of the form $\mathcal{B}_d$ (Left), and illustration of the maximization problem that defines x_t (Right).

why we can not immediately apply this algorithm to the problem described in the previous subsection.

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 $B_t = \{\nu : ||\nu - \hat{\theta}_t||_{2,A} \leq \sqrt{\beta_t}\}$ where $||u||_{2,A_t} = u^T A u$ and $\hat{\theta}_{t+1} = A_{t+1}^{-1} X_t R_t$, and to pull the arm with largest inner product with a vector in B_t, i.e. the arm such that $x_t = \arg\max_{x \in \mathcal{B}_d} \max_{\nu \in B_t} \langle \nu, x \rangle$.

Note that this algorithm is intended for general shapes of the set of arms. We can thus 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. $\mathcal{B}_d$ and this case is simpler. 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. 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 $k = \{1, \dots, d\}$, $\exists \sigma_k$ 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

Now we come back to our setting where $n \ll K$. We present here an algorithm, called *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, that uses compressed sensing ideas and which is 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[6]: at the end of this phase, the algorithm selects a set of coordinates $\mathcal{A}$, named *active set*, which

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

are the directions where θ is likely to be non-zero. The length of this phase is problem dependent and is built to be of order $\frac{\sqrt{n}}{||\theta||_2}$: if the problem is *difficult*, i.e. $||\theta||_2$ is small, we need a longer support exploration phase than if the problem is *simple*, i.e. $||\theta||_2$ big. Note that the algorithm automatically adapts the length of this phase and that no lower bound on $||\theta||_2$ is needed. The Support Exploration Phase stops at the first time t such that (i) $\max_k |\hat{\theta}_{k,t}| - \frac{2b}{\sqrt{t}} > 0$ for a well-defined b and (ii) $t \geq \frac{\sqrt{n}}{\max_k |\hat{\theta}_{k,t}| - \frac{b}{\sqrt{t}}}$.

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 vector subspace spanned 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.

Input: parameters $\bar{\sigma}_2$, $\bar{\theta}_2$,δ.
Initialize: Set $b = (\bar{\theta}_2 + \bar{\sigma}_2)\sqrt{2\log(2K/\delta)}$.
Pull randomly an arm x_1 in $\mathcal{E}_{xploring}$ (defined in Equation 4) and observe r_1
Support Exploration Phase:
while (i) $\max_k |\hat{\theta}_{k,t}| - \frac{2b}{\sqrt{t}} < 0$ or (ii) $t < \frac{\sqrt{n}}{\max_k |\hat{\theta}_{k,t}| - \frac{b}{\sqrt{t}}}$ **do**
  Pull randomly an arm x_t in $\mathcal{E}_{xploring}$ (defined in Equation 4) and observe r_t
  Compute $\hat{\theta}_t$ using Equation 5
  $t \leftarrow t+1$
end while
Call T the length of the Support Exploration Phase
Set $\mathcal{A} = \left\{k : \hat{\theta}_{k,T} \geq \frac{2b}{\sqrt{T}}\right\}$
Restricted Linear Bandit Phase:
For $t = T+1, \ldots, n$, apply $CB_2(\mathcal{B}_K \cap Vec(\mathcal{A}), \delta)$ and collect the r_t.

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

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

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

2.2 Main Result

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. $|\eta_{k,t}| \leq \frac{1}{2}\sigma_k$.*

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

Under this assumption, we have the following bound on the regret.

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 $1 - 5\delta$, as*

$$R_n(\mathcal{A}lg_{SL-UCB}) \leq 118(\bar{\theta}_2 + \bar{\sigma}_2)^2 \log(2K/\delta) S\sqrt{n}.$$

The proof of this result is available in Section 4.

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: the length of this phase is of order $O(\frac{\sqrt{n}}{||\theta||_2})$. More precisely, the smaller $||\theta||_2$, the more difficult the problem is (as it is difficult to find the biggest components of the support), and the longer the Support Exploration Phase. But note that the regret does not deteriorate, as the smaller $||\theta||_2$, the smaller also the loss at each step.

3 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, where one computes the gradient $\nabla f(u)$ of f at point u, move in the direction of the gradient, and then iterate n times. See for instance the book of Bertsekas (1999) for an exhaustive survey on gradient methods.

3.1 Formalization

The objective is to apply gradient ascent to a 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

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

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 $\theta + \eta_t = \nabla f(w_t)$[7], and $x_t = u_t - u_{t-1}$. We illustrate this model in Figure 3.

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 is in $O(S\epsilon\sqrt{n})$ for SL-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.

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.

3.2 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 (we represent in Figure 4 the projection of the function in these two informative directions). Figure 4 shows the trajectory of the algorithm, projected in the subspace of dimension 2 where the function is not constant.

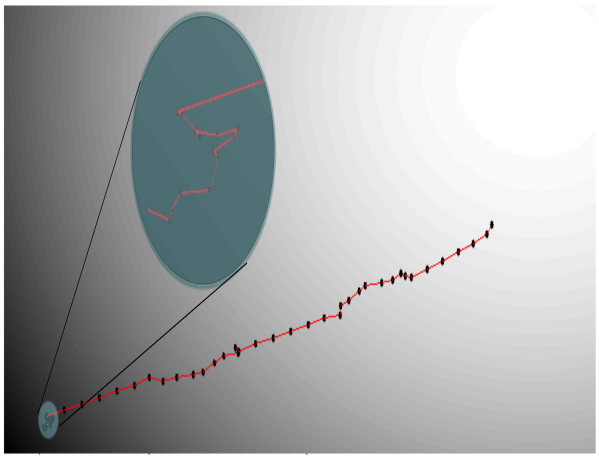

Figure 4: Illustration of the trajectory of algorithm SL-UCB with a budget $n = 50$, with a zoom at the beginning of the trajectory to illustrate the support exploration phase. The levels of gray correspond to the contours of function.

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. 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[8].

We compare our algorithm SL-UCB with two strategies: the "oracle" gradient strategy (OGS), i.e. a gradient algorithm with access to the *full* gradient of the function[9], 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

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

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

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

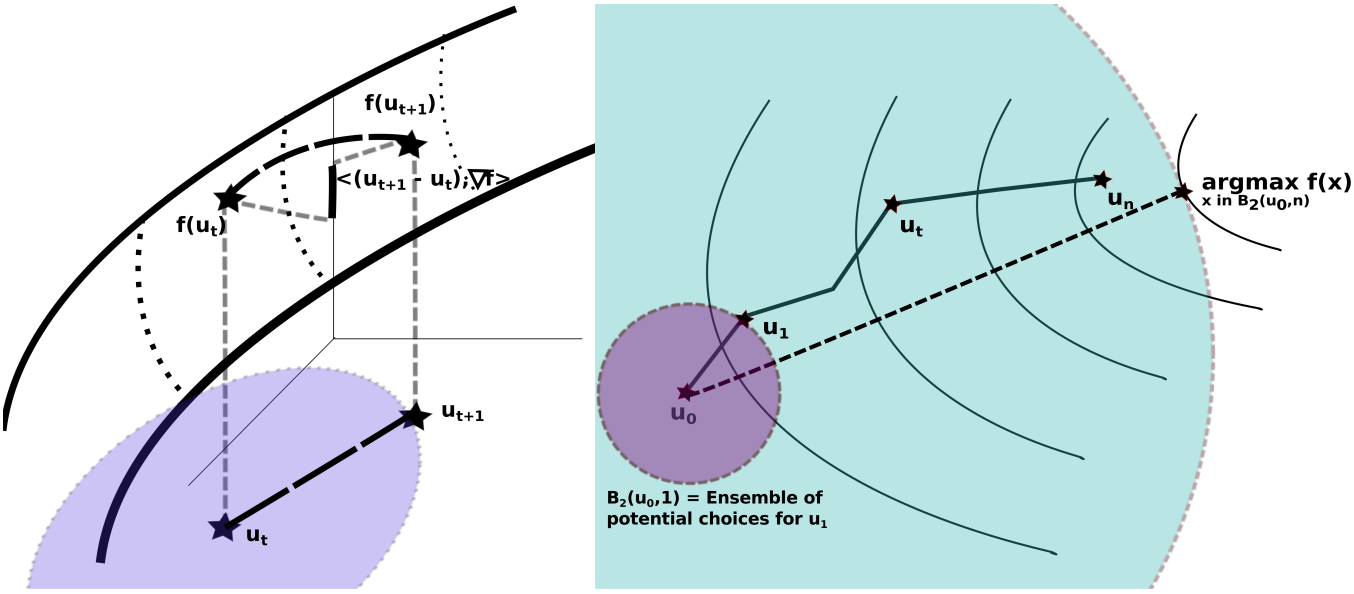

Figure 3: The gradient ascent: the first picture illustrates the problem written as a linear bandit problem with rewards and the second picture illustrates the regret.

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 5.

K/n	OGS	SL-UCB	BRD
2	$1.875\ 10^5$	$1.723\ 10^5$	$2.934\ 10^4$
10	$1.875\ 10^5$	$1.657\ 10^5$	$1.335\ 10^4$
100	$1.875\ 10^5$	$1.552\ 10^5$	$5.675\ 10^3$

Figure 5: We report, for different values of $\frac{K}{n}$ and different strategies, the value of $f(u_n) - f(u_0)$.

Note that the performances of our algorithm is worse than the "oracle" gradient strategy. This is not surprising because SL-UCB is only given partial information on the gradient. However it performs much better than the random best direction. Note that the bigger $\frac{K}{n}$, the more impressive the improvements of SL-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 algorithm is built for dealing with such problems.

4 Analysis of the SL-UCB algorithm

4.1 Event ξ of interest

Step 0: Bound on the variations of $\hat{\theta}_t$ around its mean during the Support Exploration Phase Note that as $x_{k,t} = \frac{1}{\sqrt{K}}$ or $x_{k,t} = -\frac{1}{\sqrt{K}}$ during the Support Exploration Phase, the estimate $\hat{\theta}_t$ of θ during the Support Exploration Phase is such that, for any $t_0 \leq T$ and any k

$$\begin{aligned}\hat{\theta}_{k,t_0} &= \frac{K}{t_0}\Big(\sum_{t=1}^{t_0} x_{k,t} r_t\Big) = \frac{K}{t_0}\Big(\sum_{t=1}^{t_0} x_{k,t} \sum_{k'=1}^{K} x_{k',t}(\theta_{k'} + \eta_{k',t})\Big) \\ &= \frac{K}{t_0}\sum_{t=1}^{t_0} x_{k,t}^2\theta_k + \frac{K}{t_0}\sum_{t=1}^{t_0} x_{k,t}\sum_{k'\neq k} x_{k',t}\theta_{k'} \\ &\quad + \frac{K}{t_0}\sum_{t=1}^{t_0} x_{k,t}\sum_{k'=1}^{K} x_{k',t}\eta_{k',t} \\ &= \theta_k + \frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'\neq k} b_{k,k',t}\theta_{k'} + \frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t},\end{aligned} \tag{6}$$

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 $\frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'\neq k} b_{k,k',t}\theta_{k'}$.

Note that the $b_{k,k',t}\theta_{k'}$ are $(K-1)T$ zero-mean independent random variables and that among them, $\forall k' \in \{1, ..., K\}$, t_0 of them are bounded by $\theta_{k'}$, i.e. the $(b_{k,k',t}\theta_{k'})_t$. By Hoeffdings inequality, we thus have with probability $1-\delta$ that $|\frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'\neq k}^{K} b_{k,k',t}\theta_{k'}| \leq$

$\frac{||\theta||_2\sqrt{2\log(2/\delta)}}{\sqrt{t_0}}$. Now by using an union bound on all the $k = \{1, \dots, K\}$, we have w.p. $1-\delta$, $\forall k$,

$$|\frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'\neq k} b_{k,k',t}\theta_{k'}| \leq \frac{||\theta||_2\sqrt{2\log(2K/\delta)}}{\sqrt{t_0}}. \quad (7)$$

Step 2: Study of the second term. Let us now study $\frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t}$.

Note that the $(b_{k,k',t}\eta_{k',t})_{k',t}$ are Kt_0 independent zero-mean random variables, and that among these variables, $\forall k \in \{1, ..., K\}$, t_0 of them are bounded by $\frac{1}{2}\sigma_k$. By Hoeffdings inequality, we thus have with probability $1-\delta$ that $|\frac{1}{t_0}\sum_{t=1}^{t_0}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t}| \leq \frac{||\sigma||_2\sqrt{2\log(2/\delta)}}{\sqrt{t_0}}$. Thus by an union bound with probability $1-\delta$, $\forall k$,

$$|\frac{1}{T}\sum_{t=1}^{t_0}\sum_{k'=1}^{K} b_{k,k',t}\eta_{k',t}| \leq \frac{||\sigma||_2\sqrt{2\log(2K/\delta)}}{\sqrt{t_0}}. \quad (8)$$

Step 3: Global bound Finally for a given t_0, with probability $1-2\delta$, we have by Equations 6, 7 and 8

$$||\hat{\theta}_T - \theta||_\infty \leq \frac{(||\theta||_2 + ||\sigma||_2)\sqrt{2\log(2K/\delta)}}{\sqrt{T}}. \quad (9)$$

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

$$\xi = \bigcap_{t=1,\dots,n}\left\{\omega \in \Omega / ||\theta - \frac{K}{t}X_t R_t||_\infty \leq \frac{b}{\sqrt{t}}\right\}, \quad (10)$$

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

From Equation 9 and an union bound over time, we deduce that $\mathbb{P}(\xi) \geq 1 - 2n\delta$.

4.2 Length of the Support Exploration Phase

The Support Exploration Phase stops at the first time t such that (i) $\max_k |\hat{\theta}_{k,t}| - \frac{2b}{\sqrt{t}} > 0$ and (ii) $t \geq \frac{\sqrt{n}}{\max_k |\hat{\theta}_{k,t}| - \frac{b}{\sqrt{t}}}$.

Step 1: A result on the empirical best arm On the event ξ, we know that for any t and any k, $|\theta_k| - \frac{b}{\sqrt{t}} \leq |\hat{\theta}_{k,t}| \leq |\theta_k| + \frac{b}{\sqrt{t}}$. In particular for $k^* = \arg\max_k |\theta_k|$ we have

$$|\theta_{k^*}| - \frac{b}{\sqrt{t}} \leq \max_k |\hat{\theta}_{k,t}| \leq |\theta_{k^*}| + \frac{b}{\sqrt{t}}. \quad (11)$$

Step 2: Maximum length of the Support Exploration Phase If $|\theta_{k^*}| - \frac{3b}{\sqrt{t}} > 0$ then by Equation 11, the first (i) criterion is verified on ξ. If $t \geq \frac{1}{\theta_{k^*} - \frac{3b}{\sqrt{t}}}\sqrt{n}$ then by Equation 11, the second (ii) criterion is verified on ξ.

Note that both those conditions are thus verified if $t \geq \max\left(\frac{9b^2}{|\theta_{k^*}|^2}, \frac{4\sqrt{n}}{3|\theta_{k^*}|}\right)$. The Support Exploration Phase stops thus before this moment. Note that as the budget of the algorithm is n, we have on ξ that $T \leq \max\left(\frac{9b^2}{|\theta_{k^*}|^2}, \frac{4\sqrt{n}}{3|\theta_{k^*}|}, n\right) \leq \frac{9\sqrt{S}b^2}{||\theta||_2}\sqrt{n}$. We write $T_{\max} = \frac{9\sqrt{S}b^2}{||\theta||_2}\sqrt{n}$.

Step 3: Minimum length of the Support Exploration Phase If the first (i) criterion is verified then on ξ by Equation 11 $|\theta_{k^*}| - \frac{b}{\sqrt{t}} > 0$. If the second (ii) criterion is verified then on ξ by Equation 11 we have $t \geq \frac{\sqrt{n}}{|\theta_{k^*}|}$.

Combining those two results, we have on the event ξ that $T \geq \max\left(\frac{b^2}{\theta_{k^*}^2}, \frac{\sqrt{n}}{|\theta_{k^*}|}\right) \geq \frac{b^2}{||\theta||_2}\sqrt{n}$. We write $T_{\min} = \frac{b^2}{||\theta||_2}\sqrt{n}$.

4.3 Description of the set $\mathcal{A}$

The set $\mathcal{A}$ is defined as $\mathcal{A} = \left\{k : |\hat{\theta}_{k,T}| \geq \frac{2b}{\sqrt{T}}\right\}$.

Step 1: Arms that are in $\mathcal{A}$ Let us consider an arm k such that $|\theta_k| \geq \frac{3b\sqrt{||\theta||_2}}{n^{1/4}}$. Note that $T \geq T_{\min} = \frac{b^2}{||\theta||_2}\sqrt{n}$ on ξ. We thus know that on ξ

$$|\hat{\theta}_{k,T}| \geq |\theta_k| - \frac{b}{\sqrt{T}} \geq \frac{3b\sqrt{||\theta||_2}}{n^{1/4}} - \frac{b\sqrt{||\theta||_2}}{n^{1/4}} \geq \frac{2b}{\sqrt{T}}.$$

This means that $k \in \mathcal{A}$ on ξ. We thus know that $|\theta_k| \geq \frac{3b\sqrt{||\theta||_2}}{n^{1/4}}$ implies on ξ that $k \in \mathcal{A}$.

Step 2: Arms that are not in $\mathcal{A}$ Now let us consider an arm k such that $|\theta_k| < \frac{b}{2\sqrt{n}}$. Then on ξ, we know that

$$|\hat{\theta}_{k,T}| < |\theta_k| + \frac{b}{\sqrt{T}} < \frac{b}{2\sqrt{n}} + \frac{b}{\sqrt{T}} < \frac{3b}{2\sqrt{T}} < \frac{2b}{\sqrt{T}}.$$

This means that $k \in \mathcal{A}^c$ on ξ.
This implies that on ξ, if $|\theta_k| = 0$, then $k \in \mathcal{A}^c$.

Step 3: Summary Finally, we know that $\mathcal{A}$ is composed of all the $|\theta_k| \geq \frac{3b\sqrt{||\theta||_2}}{n^{1/4}}$, and that it contains only positives θ_k, i.e. at most S elements since θ is $S-$sparse. We write $\mathcal{A}_{\min} = \{k : |\theta_k| \geq \frac{3b\sqrt{||\theta||_2}}{n^{1/4}}\}$

4.4 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 $\max_{x_t \in Vec(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle = ||\theta_{\mathcal{A}}||_2 = \sqrt{\sum_{k=1}^K \theta_k^2 \mathbb{I}\{k \in \mathcal{A}\}}$, where $\theta_{\mathcal{A},k} = \theta_k$ if $k \in \mathcal{A}$ and $\theta_{\mathcal{A},k} = 0$ otherwise. This means that

$$\begin{aligned}
&\max_{x_t \in \mathcal{B}_K} \langle \theta, x_t \rangle - \max_{x_t \in Vec(\mathcal{A}) \cap \mathcal{B}_K} \langle \theta, x_t \rangle \\
&= ||\theta||_2 - ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2 = \frac{||\theta||_2^2 - ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2^2}{||\theta||_2 + ||\theta \mathbb{I}\{k \in \mathcal{A}\}||_2} \\
&\leq \frac{\sum_{k \in \mathcal{A}^c} \theta_k^2}{||\theta||_2} \leq \frac{\sum_{k \in \mathcal{A}_{\min}^c} \theta_k^2}{||\theta||_2} \leq \frac{9Sb^2}{\sqrt{n}}. \qquad (12)
\end{aligned}$$

4.5 Expression of the regret of the algorithm

Assume that we launch algorithm $ConfidenceBall_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, T)$ at time T where $\mathcal{A} \subset Supp(\theta)$ with a budget of $n_1 = n - T$ samples. In paper (Dani et al., 2008), they prove that on an event $\xi_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, T)$ of probability $1 - \delta$ the regret of algorithm $ConfidenceBall_2$ is bounded by $R_n(\mathcal{A}lg_{CB_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, T)}) \leq 64|\mathcal{A}|\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n_1}$.

Note now that as $\mathcal{A} \subset Supp(\theta)$, there is $\xi_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, T) \subset \xi_2(Vec(Supp(\theta)) \cap \mathcal{B}_K, \delta, T)$ (see paper (Dani et al., 2008) for more details on the event ξ_2). We thus now that, conditional to T, with probability $1 - \delta$, the regret is bounded for any $\mathcal{A} \subset Supp(\theta)$ as $R_n(\mathcal{A}lg_{CB_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, T)}) \leq 64S\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^2/\delta))^2\sqrt{n_1}$.

By doing an union bound on all possible values for T (i.e. from 1 to n), we obtain that on an event ξ_2 whose probability is higher than $1 - \delta$, $R_n(\mathcal{A}lg_{CB_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, T)}) \leq 64S\Big(||\theta||_2 + ||\sigma||_2\Big)(\log(n^3/\delta))^2\sqrt{n}$.

We thus have on $\xi \bigcup \xi_2$, i.e. with probability higher than $1 - 2\delta$ that

$$\begin{aligned}
R_n(\mathcal{A}lg_{SL-UCB}, \delta) \leq\ & 2T_{\max}||\theta||_2 \\
&+ \max_t R_n(\mathcal{A}lg_{CB_2(Vec(\mathcal{A}) \cap \mathcal{B}_K, \delta, t)}) \\
&+ n\Big(\max_{x \in \mathcal{B}_K} \langle x, \theta \rangle - \max_{x \in \mathcal{B}_K \cap Vect(\mathcal{A}_{\min})} \langle x, \theta \rangle\Big).
\end{aligned}$$

By using this Equation, the maximal length of the support exploration phase $T_{\max}$ deduced in Step 2 of Subsection 4.2, and Equation 12, we obtain on ξ

$$\begin{aligned}
R_n &\leq 64S\big(||\theta||_2 + ||\sigma||_2\big)(\log(n^2/\delta))^2\sqrt{n} \\
&\quad + 18Sb^2\sqrt{n} + 9Sb^2\sqrt{n} \\
&\leq 118(\bar{\theta}_2 + \bar{\sigma}_2)^2 \log(2K/\delta) S\sqrt{n}.
\end{aligned}$$

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

Conclusion

In the paper we provided an algorithm for sparse linear bandits in high dimension. It has been designed using ideas from compressed sensing and bandit theory. Compressed sensing is used in the support exploration phase, in order to select the support of the parameter. A linear bandit algorithm from (Dani et al., 2008) is then applied to the small dimensional subspace defined by the first phase. The algorithm SL-UCB provides a regret of order $O(S\sqrt{n})$. Note that all the bound scales 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 high dimension. There are to our minds two main directions for further researches.

- It could be interesting to deal with the case when the support of θ evolves in time: 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 support of the gradient changes in space.
- The bound that we obtain is in $O(S\sqrt{n})$, and it would be interesting to see if it is possible to build an algorithm that has a regret in $O(\sqrt{Sn})$, which is a theoretic lower bound for this problem. Note that when an upper bound S' on the sparsity is available, it seems possible to obtain such a regret by replacing condition (ii) in the algorithm by $t < \frac{\sqrt{n}}{||\Big(\hat{\theta}_{t,k}\mathbb{I}\Big\{\hat{\theta}_{t,k} \geq \frac{b}{\sqrt{t}}\Big\}\Big)_k||_2 - \frac{\sqrt{S'}b}{\sqrt{t}}}$, and using for the Exploitation phase the algorithm in (Rusmevichientong and Tsitsiklis, 2008). The regret of such an algorithm would be in $O(\sqrt{S'n})$. But it is not clear whether it is possible to obtain such results when no upper bound on S is available.

Acknowledgements
This research was partially supported by Region Nord-Pas-de-Calais Regional Council, French ANR EXPLORA (ANR-08-COSI-004), the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement 231495 (project CompLACS), and by Pascal-2.

References

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

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

Y. Abbasi-yadkori, D. Pal, and C. Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, 2012.

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.