Sparse Linear Contextual Bandits via Relevance Vector Machines

Davis Gilton and Rebecca Willett
Electrical and Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706
Email: gilton@wisc.edu, willett@discovery.wisc.edu

***Abstract*—This paper describes a linear multi-armed bandit algorithm that exploits sparsity in the underlying unknown weight vector controlling rewards. In linear multi-armed bandits, a user chooses a sequence of (slot machine) "arms" to pull, and each arm pull results in the user receiving a stochastic reward with mean equal to the inner product between a known feature vector associated with the arm and an unknown weight vector. While linear bandit algorithms have been widely considered in the literature, relatively little is known about how to exploit sparsity in the weight vector. This paper describes a novel approach that leverages ideas from linear Thompson sampling and relevance vector machines, resulting in a scalable approach that adapts to the unknown sparse support. Theoretical regret bounds highlight the proposed algorithm's performance as a function of the sparsity level, and simulations illustrate the advantages of the proposed method over several competing approaches.**

I. Introduction

Classical multi-armed Bandit (MAB) algorithms are often described in the context of a gambler facing an array of slot machines. Each time one of the machine's arms is pulled, it gives the gambler a reward; the rewards are typically stochastic and identically distributed for each arm, and each arm has its own unknown average payoff. The gambler must devise a strategy for pulling slot machine arms to maximize his/her total rewards from the machines, balancing between exploration (*i.e.,* estimating the average payoff of each arm) and exploitation (*i.e.,* pulling arms known to have large average payoffs). *Linear* MABs (LMABs) assume that each slot machine is associated with a known feature vector, and the expected payoff of the machine will be the inner product of this feature vector and an unknown weight vector. In this setting, a pull of any arm provides some information about the unknown weight vector and hence insight into the average payoff of all the other arms. Classical and linear MABs are applicable in a variety of settings, notably online advertising.

In many contexts, the unknown weight vector is *sparse* because only a few features of the arms are relevant to the expected rewards. However, standard linear bandit strategies do not exploit this sparsity and yield far smaller rewards than oracle LMAB algorithms with full knowledge of the support of the weight vector. Relatively few studies have focused on sparse LMAB. This paper describes a new approach to LMABs that leverages ideas from Linear Thompson Sampling [1] and Relevance Vector Machines [2]. We present our proposed algorithm, theoretical regret bounds, and simulation results illustrating how the proposed approach adapts to sparse weight vectors and outperforms competing algorithms.

A. Problem formulation

Consider a collection $\mathcal{X}$ of N arms that are represented by d-dimensional feature vectors denoted $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d$. Our interest is in settings with large d. Without loss of generality, assume the feature vectors $\boldsymbol{x} \in \mathcal{X}$ have ℓ_2 norm at most one, and that also $\|\boldsymbol{\theta}^*\|_2 \leq 1$. These assumptions alter any theoretical bounds on performance only by constants. At each time $t = 1, 2, \ldots$, the user selects an arm $\boldsymbol{x}_t \in \mathcal{X}$ and receives a reward $y_t \in \mathbb{R}$. We model the reward as an unknown, noisy linear function of $\boldsymbol{x}_t$; specifically,

$$y_t = \langle \boldsymbol{\theta}^*, \boldsymbol{x}_t \rangle + \eta_t \ , \tag{1}$$

where η_t is a noise term that is drawn from a zero-mean Gaussian distribution with variance σ^2 conditioned on $\boldsymbol{x}_{1:t}$ and $\eta_{1:(t-1)}$, and $\boldsymbol{\theta}^* \in \mathbb{R}^d$ is an unknown weight vector reflecting how much different features of the arms contribute to the reward. If we knew $\boldsymbol{\theta}^*$, then we would choose

$$\boldsymbol{x}^* := \arg\max_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{\theta}^*, \boldsymbol{x} \rangle$$

at each round to maximize expected rewards. However, $\boldsymbol{\theta}^*$ is unknown, and estimating $\boldsymbol{\theta}^*$ is non-trivial due to the high dimensionality of the arms. This paper assumes that $\boldsymbol{\theta}^*$ is s-sparse for some $s \ll d$, meaning that $\boldsymbol{\theta}^*$ is supported on at most s unknown indices.

We also define $\boldsymbol{X}_t := \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_t \end{bmatrix}^\top \in \mathbb{R}^{t \times d}$, $\boldsymbol{y}_t := \begin{bmatrix} y_1 & \cdots & y_t \end{bmatrix}^\top \in \mathbb{R}^t$, and let

$$\hat{\boldsymbol{\theta}}_t := \boldsymbol{\Sigma}_t^{-1} \boldsymbol{X}_t^\top \boldsymbol{y}_t \tag{2}$$

be the ridge regression estimator of $\boldsymbol{\theta}^*$ given the arms and rewards up to time t, where $\boldsymbol{\Sigma}_t := \boldsymbol{X}_t^\top \boldsymbol{X}_t + \lambda \boldsymbol{I} \in \mathbb{R}^{d \times d}$ is defined as above, and $\lambda \in \mathbb{R}$ is a regularizing constant.

Existing linear MAB algorithms provide analysis on their cumulative rewards up to time step T. Define $[T] := \{1, \ldots, T\}$. Instead of guaranteeing an absolute quantity on the cumulative rewards, these algorithms guarantee a relative quantity called *cumulative pseudo-regret*, which measures the

extra expected cumulative reward one could have received by pulling $\boldsymbol{x}^*$ at each round: 8

$$\mathcal{R}(T) := \sum_{t=1}^{\top} \langle \boldsymbol{\theta}^*, \boldsymbol{x}^* \rangle - \langle \boldsymbol{\theta}^*, \boldsymbol{x}_t \rangle \ , \tag{3}$$

which we refer to as *regret*.

II. Related work on linear bandits

The first formal linear stochastic MAB analysis appears in [3] but is under a more general family of problems. Dani *et al.* [4] focus on the linear stochastic MAB case and propose an algorithm based on confidence bounds with a regret bound that is later shown to be optimal up to logarithmic factors by [5].

A. Optimism in the Face of Uncertainty for Linear Rewards (OFUL)

Abbasi-Yadkori *et al.* [6] propose a significantly tighter confidence bound, which slightly improves the regret bound and improves the practical performance by orders of magnitude. This method is coined "Optimism in the Face of Uncertainty for Linear Rewards (OFUL)" [6]. estimator of $\boldsymbol{\theta}^*$ given the arms and rewards up to time t.

Assume $\|\boldsymbol{\theta}^*\|_2 \leq 1$ and let $\delta \in (0,1)$ be the target failure rate. Define

$$\sqrt{\beta_t} := \sigma \sqrt{2 \log \left(\frac{\det(\boldsymbol{\Sigma}_t)^{1/2} \det(\lambda \boldsymbol{I})^{-1/2}}{\delta} \right)} + \sqrt{\lambda}$$

Also denoting for some positive semidefinite $\mathbf{M} \in \mathbb{R}^{d \times d}$ and $\mathbf{w} \in \mathbb{R}^d$, $\|\mathbf{w}\|_{\mathbf{M}} := \sqrt{\mathbf{w}^\top \mathbf{M} \mathbf{w}}$, the confidence set at time t is given by:

$$C_t := \{ \boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\|_{\boldsymbol{\Sigma}_t} \leq \sqrt{\beta_t} \}, \tag{4}$$

OFUL chooses which arm to pull by solving the following optimization problem:

$$(\tilde{\boldsymbol{\theta}}_t, \boldsymbol{x}_t^{\text{OFUL}}) := \underset{\boldsymbol{\theta} \in C_{t-1}, \boldsymbol{x} \in \mathcal{X}}{\arg\max} \ \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle \ . \tag{5}$$

where $C_{t-1} \subseteq \mathbb{R}^d$ is a confidence set that contains $\boldsymbol{\theta}^*$ with high probability. This approach is "optimistic" in the following sense: we know $\boldsymbol{\theta}^* \in C_{t-1}$ with high probability, but nothing more, so we choose $\tilde{\boldsymbol{\theta}}_t$ to be the weight vector in C_{t-1} that would yield the maximum reward given the best arm pull, and we optimistically use that weight vector to choose $\boldsymbol{x}_t$.

The optimization problem (5) has a natural interpretation of balancing between an "exploitation" term that encourages arms aligned with the current ridge regression estimator $\hat{\boldsymbol{\theta}}_t$ and a second "exploration" term that encourages arms aligned with the directions that have been probed less so far.

OFUL achieves the near-optimal regret $\tilde{O}(d\sqrt{T})$ [5], which implies that in the long-term, OFUL will retrieve as large a cumulative reward as one could hope.

B. Linear Thompson Sampling

Linear Thompson Sampling (LTS) is a linear extension of the Thompson sampling algorithm [7] that is known to perform very well in practice. Departing from OFUL-based algorithms, LTS samples a vector $\tilde{\boldsymbol{\theta}}_t$ from a multivariate normal distribution $\mathcal{N}(\hat{\boldsymbol{\theta}}_t, v^2 \boldsymbol{\Sigma}_t^{-1})$ for $v \propto \sqrt{9 d \sigma^2 \log(T)} \in \mathbb{R}$ and then chooses the arm

$$\boldsymbol{x}_t^{\text{LTS}} = \underset{\boldsymbol{x} \in \mathcal{X}}{\arg\max} \langle \tilde{\boldsymbol{\theta}}_t, \boldsymbol{x} \rangle.$$

The regret bound of LTS is $\tilde{O}(d^{3/2}\sqrt{T})$ [8].

Algorithm 1 Linear Thompson Sampling (LTS) [8]

Initialize $\hat{\boldsymbol{\theta}}_1 = 0$ and $\boldsymbol{\Sigma}_1 = \boldsymbol{I}_d$
Let $\delta \in (0,1)$ and set $v = \sqrt{9 d \sigma^2 \log(T/\delta)}$
for $t = 1, 2, \ldots, T$ **do**
  Sample $\tilde{\boldsymbol{\theta}}_t$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\theta}}_t, v^2 \boldsymbol{\Sigma}_t^{-1})$
  Play arm $\boldsymbol{x}_t := \arg\max_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{x}, \tilde{\boldsymbol{\theta}}_t \rangle$
  Receive reward y_t
  Set $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t + \boldsymbol{x}_t \boldsymbol{x}_t^\top = \boldsymbol{I}_d + \boldsymbol{X}_t^\top \boldsymbol{X}_t$
  Set $\hat{\boldsymbol{\theta}}_{t+1} = \boldsymbol{\Sigma}_{t+1}^{-1} \boldsymbol{X}_t^\top \boldsymbol{y}_t$
end for

LTS and OFUL share some important features. Specifically, both algorithms begin by computing the ridge regression estimate $\hat{\boldsymbol{\theta}}_t$, and encouraging exploration in order to avoid the potential pitfalls of being greedy. OFUL chooses the most optimistic estimate from $\tilde{\boldsymbol{\theta}}_t \in C_{t-1}$, and LTS draws $\tilde{\boldsymbol{\theta}}_t$ from a Gaussian centered at $\hat{\boldsymbol{\theta}}_t$, which has C_{t-1} as a level set. Both algorithms choose the next arm $\boldsymbol{x}_t$ to maximize the inner product between the arm feature vector and their "proxy" weight vector. For large collections of arms, LTS is more computationally efficient, while OFUL has the lower regret bound for large D.

III. Sparse linear bandits

The above approaches do not leverage sparsity in the weight vector $\boldsymbol{\theta}^*$. While there is increasing interest in sparse models, there is no consensus on how to best incorporate them within bandit algorithms. Naïvely, we might consider computing a sparse estimate $\hat{\boldsymbol{\theta}}_t$ using LASSO or elastic net regression instead of the ridge regression estimate in (2). This approach can yield some empirical advantages, but it is difficult to characterize the confidence set C_t associated with the new sparsity-promoting estimate $\hat{\boldsymbol{\theta}}_t$. Without this confidence set, we cannot improve the scaling of the regret bounds.

Bounding techniques used in standard LASSO analysis rely upon the matrix $\boldsymbol{X}_t$ satisfying certain properties (*e.g.,* the restricted isometry property or restricted eigenvalue condition), but in the bandit setting $\boldsymbol{X}_t$ corresponds to the sequence of arms selected and is not guaranteed to have this property. Two approaches have been proposed to address this challenge.

The first approach [9] assumes the existence of an online learning algorithm that can be used to estimate $\hat{\boldsymbol{\theta}}_t$ (*e.g.,* online LASSO). This algorithm is used to predict y_t and it is assumed that there is a known upper bound on the prediction loss.

The predictions and the bound on the prediction error is used to determine a confidence set analogous to C_t in the OFUL algorithm. However, known bounds on online LASSO algorithms are too loose for this method to perform well in practice.

The second approach [10] operates in two epochs. For the first $O(\sqrt{T})$ rounds, random non-adaptive arms are pulled in a "support exploration phase"; based on the rewards from these rounds, the support of $\boldsymbol{\theta}^*$ is estimated, and for the remaining rounds the OFUL algorithm is run on the estimated support of $\boldsymbol{\theta}^*$. However, the support exploration phase is non-adaptive and hence this approach can yield smaller cumulative rewards than alternative methods, particularly with the support size s of $\boldsymbol{\theta}^*$ is small relative to $\sqrt{T}$. This is illustrated in the simulations below.

IV. Relevance vector machines

The relevance vector machine (RVM) is a Bayesian framework that is particularly useful in sparse regression and classification tasks, introduced in [2]. The RVM setup considered here assumes we observe

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta}^* + \boldsymbol{\eta},$$

where $\boldsymbol{\eta} \in \mathbb{R}^t$ is a Gaussian noise vector with distribution $\mathcal{N}(0, \sigma^2 I_t)$. The RVM assumes a prior on $\boldsymbol{\theta}^*$ of the form

$$\boldsymbol{\theta}^* \sim \mathcal{N}(0, \mathbf{A}^{-1}),$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$ is a diagonal matrix of hyperparameters with $A_{i,i} = \alpha_i$; we define $\boldsymbol{\alpha} := \begin{bmatrix}\alpha_1, & \alpha_2, & \ldots, & \alpha_d\end{bmatrix}$ to be the diagonal elements of $\mathbf{A}$. That is, the i^{th} element of $\boldsymbol{\theta}^*$ is a zero-mean Gaussian with variance α_i^{-1}.

The RVM further assigns a hyperprior on $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} \sim \prod_{i=1}^{d} \text{Gamma}(0, 0).$$

Using Bayes' rule, the posterior distribution of $\boldsymbol{\theta}^*$ is

$$\boldsymbol{\theta}^* | \boldsymbol{y}, \boldsymbol{\alpha} \sim \mathcal{N}(\hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}^{-1}) \tag{6a}$$

where

$$\boldsymbol{\Sigma} = \mathbf{A} + \sigma^{-2}\boldsymbol{X}^\top\boldsymbol{X} \tag{6b}$$

$$\hat{\boldsymbol{\theta}} = \sigma^{-2}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^\top\boldsymbol{y} \tag{6c}$$

If all the α_i's are equal to λ, then this estimate is equivalent to (2). Since $\boldsymbol{\alpha}$ is unknown and computing the full posterior is computationally intensive, [2] proposes choosing $\boldsymbol{\alpha}$ to maximize $p(\boldsymbol{y}|\boldsymbol{\alpha})$ and describes an iterative approach to solving this optimization problem, with computational speedups detailed in [11].

The Relevance Vector Machine is appealing for our task for two reasons. First, as noted in [2], the RVM tends to produce estimates of $\boldsymbol{\alpha}$ which are very sparse. Secondly, the form of $\hat{\boldsymbol{\theta}}$ in (6) is very similar to the ridge regression estimator of (2) that is used in LTS and OFUL, so some regret bounds might be feasible to propose.

V. Relevance vector machine linear Thompson sampling (RVM-LTS)

In the stochastic linear bandit regime, the RVM is interesting because the posterior distribution in (6) can be used within an LTS framework to draw $\tilde{\boldsymbol{\theta}}_t$ (instead of the original $\mathcal{N}(\hat{\boldsymbol{\theta}}_t, v^*\boldsymbol{\Sigma}_t^{-1})$ described in Section II-B). By using the RVM posterior this way, we essentially form a sampling distribution (with a level set corresponding to a confidence set C_t) that adapts to the unknown sparse support of $\boldsymbol{\theta}^*$.

A. Algorithm

Our proposed approach is in Algorithm 2; the overall structure is similar to the LTS algorithm described in Section II-B, with two key differences. First, in line 8, the covariance matrix is offset by diagonal matrix $\mathbf{A}$ instead of the conventional $\lambda\boldsymbol{I}_d$. Second, in line 7, $\mathbf{A}$ is updated adaptively based on the data acquired so far. It is this adaptation that allows the sparsity of $\boldsymbol{\theta}^*$ to be exploited. Note that $\mathbf{A}$ can be updated quickly by using the algorithm outlined by Tipping and Faul [11][1].

Algorithm 2 Relevance Vector Machine Linear Thompson Sampling (RVM-LTS)

1: Initialize $\hat{\boldsymbol{\theta}}_1 = 0$ and $\mathbf{A} = \boldsymbol{\Sigma}_1 = \boldsymbol{I}_d$
2: Let $\delta \in (0,1)$ and set $v = \sqrt{9d\sigma^2\log(T/\delta)}$
3: **for** $t = 1, 2, \ldots, T$ **do**
4: Sample $\tilde{\boldsymbol{\theta}}_t$ from the distribution $\mathcal{N}(\hat{\boldsymbol{\theta}}_t, v^2\boldsymbol{\Sigma}_t^{-1})$
5: Play arm $\boldsymbol{x}_t := \arg\max_{\boldsymbol{x}\in\mathcal{X}}\langle \boldsymbol{x}, \tilde{\boldsymbol{\theta}}_t\rangle$
6: Receive reward y_t
7: Update $\mathbf{A}$ using [11]
8: Set $\boldsymbol{\Sigma}_{t+1} = \boldsymbol{\Sigma}_t + \boldsymbol{x}_t\boldsymbol{x}_t^\top = \mathbf{A} + \boldsymbol{X}_t^\top\boldsymbol{X}_t$
9: Set $\hat{\boldsymbol{\theta}}_{t+1} = \boldsymbol{\Sigma}_{t+1}^{-1}\boldsymbol{X}_t^\top\boldsymbol{y}_t$
10: **end for**

To better understand the sparse case, consider an oracle with perfect knowledge of the support of $\boldsymbol{\theta}^*$, denoted $S := \text{supp}(\boldsymbol{\theta}^*)$. Then if $\alpha_i = \infty$ for $i \notin S$ and $\alpha_i = \lambda$ for $i \in S$, and if we omit updating $\mathbf{A}$ in line 7, then RVM-LTS amounts to LTS on the true support of $\boldsymbol{\theta}^*$ and ignoring all other coordinate axes. Of course, in a practical problem we do not have such oracle knowledge. To adapt to this, our method updates its estimate of $\boldsymbol{\alpha}$ using [11].

B. Regret bounds for fixed $\mathbf{A}$

The above algorithm is, on the surface, very similar to standard Linear Thompson Sampling. However, the addition of the $\mathbf{A}$ matrix instead of $\boldsymbol{I}_d$ has some significant theoretical consequences.

First, note that within the RVM updating scheme, for $i \notin \text{supp}(\boldsymbol{\theta}^*)$, $\alpha_i \to \infty$. Let

$$S_{\boldsymbol{\alpha}} := \{i : 1/\alpha_i = 0\}$$

[1]The authors of [11] allow α_i to go to infinity, which allows for the omission of that particular dimension from calculations of $\hat{\boldsymbol{\theta}}_t$. Our implementation, for numerical stability, instead sets all α_i that have been set to ∞ by the $\boldsymbol{\alpha}$-updating algorithm to some suitably large $\alpha_{\max}$.

denote the support of the diagonal of $\mathbf{A}^{-1}$ and let $s_{\boldsymbol{\alpha}} := |S_{\boldsymbol{\alpha}}|$ be the size of that support; in practice, we often see $s_{\boldsymbol{\alpha}}$ is close to the true support size of $\boldsymbol{\theta}^*$. Further, let $S = \text{supp}(\boldsymbol{\theta}^*)$ and $\boldsymbol{\alpha}_S$ be $\boldsymbol{\alpha}$ on S and zero elsewhere.

We then have the following regret bound for Algorithm 2 for fixed $\mathbf{A}$ (*i.e.,* when we omit line 7).

Theorem 1: Assume $\text{supp}(\boldsymbol{\theta}^*) \subseteq S_{\boldsymbol{\alpha}}$. For $\delta > 0$, with probability at least $1-\delta$, the pseudo-regret of Algorithm 2 for fixed $\mathbf{A}$ at time T is bounded by

$$\mathcal{R}(T) = O\left(\min\{\sqrt{s_{\boldsymbol{\alpha}}}, \sqrt{\log N}\} \times \left[\sigma \sqrt{s_{\boldsymbol{\alpha}} \log \frac{T^2 \bar{\boldsymbol{\alpha}} + T^3}{\delta \sqrt[s]{\boldsymbol{\alpha}}}} + \|\boldsymbol{\alpha}_S\|_2 \right] \right),$$

where $\bar{\boldsymbol{\alpha}} := \frac{1}{s_{\boldsymbol{\alpha}}} \sum_{i \in S_{\boldsymbol{\alpha}}} \alpha_i$ is the arithmetic mean of $\boldsymbol{\alpha}$ and $\sqrt[s]{\boldsymbol{\alpha}} = (\prod_{i \in S_{\boldsymbol{\alpha}}} \alpha_i)^{1/s_{\boldsymbol{\alpha}}}$ is the geometric mean of $\boldsymbol{\alpha}$, both on the nonzero support.

Sketch proof: The proof of this theorem follows the main steps of the LTS regret bound in [8, Theorem 1]. A key element of the proof is finding concentration results for the distribution of $x_t^\top \hat{\boldsymbol{\theta}}_t$, which requires bounding $|\boldsymbol{x}_t^\top \hat{\boldsymbol{\theta}}_t - \boldsymbol{x}_t^\top \boldsymbol{\theta}^*|$ for all t. Given that there is a bound on $|\boldsymbol{x}_t|_2$, we must find how close $\hat{\boldsymbol{\theta}}_t$ is to $\boldsymbol{\theta}^*$. It is straightforward to show that $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* = \boldsymbol{\Sigma}_{t-1}^{-1}(\xi_{t-1} - \mathbf{A}\boldsymbol{\theta}^*)$, where $\xi_t = \boldsymbol{X}_t^\top \boldsymbol{y}_t \in \mathbb{R}^d$, the vector of correlations between the reward vector $\boldsymbol{y}_t$ and each of the d features of the arms pulled so far, and $\boldsymbol{\Sigma}_t = \mathbf{A} + \boldsymbol{X}_t^\top \boldsymbol{X}_t \in \mathbb{R}^{d \times d}$, the sample covariance matrix plus $\mathbf{A}$. We first show that:

$$||\xi||_{\boldsymbol{\Sigma}_t^{-1}} \leq R \sqrt{d \log \frac{\bar{\boldsymbol{\alpha}} + t}{\delta \sqrt[s]{\boldsymbol{\alpha}}}}.$$

This allows us to bound $||\xi_{t-1} - \mathbf{A}\boldsymbol{\theta}^*||_{\boldsymbol{\Sigma}_{t-1}^{-1}}$ using the steps of [8, Lemma 1], which gives, with probability $1-\delta$ for $\delta > 0$:

$$||\xi_{t-1} - \mathbf{A}\boldsymbol{\theta}^*||_{\boldsymbol{\Sigma}_{t-1}^{-1}} \leq R \sqrt{d \log \frac{\bar{\boldsymbol{\alpha}} + t}{\delta \sqrt[s]{\boldsymbol{\alpha}}}} + \|\boldsymbol{\alpha}_S\|_2.$$

The remainder of the proof follows [8, Theorem 1]

VI. Simulations

In this section we present several simulations to demonstrate regret performance for the algorithm. First, plots of regret *vs.* number of nonzero elements in $\boldsymbol{\theta}^*(s)$ are provided. A comparison is done with state-of-the-art linear bandit algorithms and another sparse linear bandit algorithm. The effect of noise on the above algorithms is also investigated.

The four main algorithms that we compare are OFUL [6], as outlined in Section II-A, Linear Thompson Sampling [8], as outlined in Section II-B, our proposed RVM-Linear Thompson Sampling (Algorithm 2, and the sparse linear bandit algorithm of [10], outlined in , which we refer to as *Epoch SOFUL* (Epoch Sparse OFUL).

In Figure 1 we illustrate the per-round regret of these algorithms. The experimental setup consists of $N = 1000$ possible arms with unit-norm feature vectors in $\mathbb{R}^{100}$, which are 5-sparse. The bandit runs for 200 rounds, and the results are averaged over 100 trials. The RVM-LTS algorithm achieves favorable per-round regret, notably beginning its exploitation phase at the same time as the Oracle Thompson Sampling. Because they do not make use of the sparsity of $\boldsymbol{\theta}^*$, standard Thompson Sampling and OFUL require roughly d arm pulls to begin reducing the per-round regret, while Epoch-OFUL pulls arms for a much shorter time before beginning its exploitation phase.

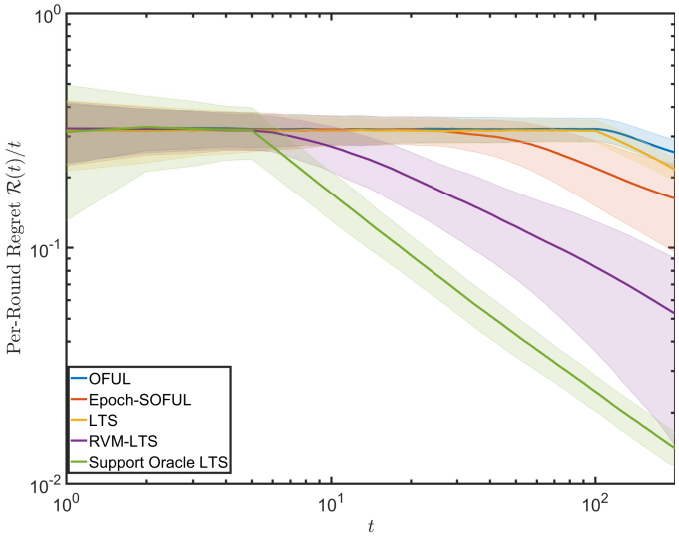

Figure 1. The per-round regret for several different bandit algorithms. Here, $s = 5$, $d = 100$, and the number of arms is $N = 1000$. Colorbars represent the standard deviation of the per-round regret, and the center lines are the mean over the 100 trials.

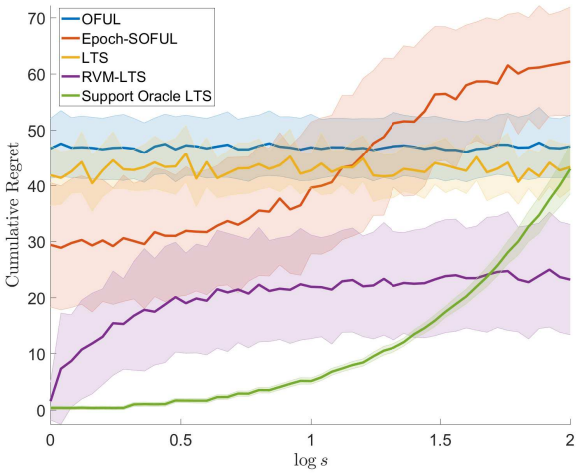

Figure 2. The total regret *vs.* s at $T = 200$ for several different bandit algorithms. Here, $d = 100$, the number of arms is $N = 1000$. Each algorithm was run 300 times for each value of s. The shaded regions represent the standard deviation of the cumulative regret, and the center lines are the mean over the 300 trials.

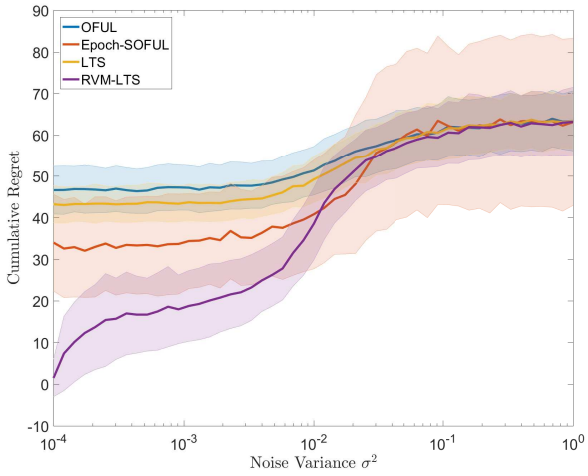

Figure 3. The total regret at $T = 200$ *vs.* noise power for several different bandit algorithms. Here, $d = 100$, the number of arms is $N = 1000$. Each algorithm was run 500 times for each value of σ^2. Colorbars represent the standard deviation of the cumulative regret, and the center lines are the mean over the 500 trials.

In Figure 2, the total regret is plotted *vs.* s, where $\boldsymbol{\theta}^*$ is s-sparse. The algorithms compared include the previously identified, plus an "Oracle Thompson Sampling" algorithm. In the oracle algorithm, the bandit is given the true support of $\boldsymbol{\theta}^*$, and runs Linear Thompson Sampling over that true support. As can be seen in Figure 2, the oracle should outperform any other algorithm for low s, but is identical to Linear Thompson Sampling when $s = d$. The RVM-LTS regret is closer to the Oracle Thompson Sampling bound for low s, implying that especially for low S, RVM-LTS is performing nearly as well as we could with full prior knowledge of the support.

As s grows, the Epoch SOFUL regret goes up significantly, which is expected of a sparse linear bandit algorithm. Interestingly, the RVM-LTS algorithm achieves lower regret than the oracle as s approaches d, indicating that the RVM-LTS algorithm may have some advantages even in the classical (non-sparse) linear bandit setting.

Finally, Figure 3 illustrates the regret response of the different linear bandit algorithms to increasing the variance of the zero-mean Gaussian noise in the rewards. Past $\sigma^2 = 1$, all the algorithms have roughly similar cumulative regret, so no algorithm under consideration appears to have an advantage in the presence of high-variance noise. However, in the low-noise regime, the RVM-LTS algorithm achieves very good regret.

VII. Conclusions

This paper describes an algorithm that combines the Relevance Vector Machine with Linear Thompson Sampling to yield a sparse linear bandits algorithm that empirically outperforms competing algorithms and is nearly as good as a support oracle. The RVM allows the sparse support of $\boldsymbol{\theta}^*$ to be estimated quickly and accurately, so that after relatively few pulls the algorithm performs as well as one operating in a lower-dimensional feature space. As a result, the algorithm is able to home in on the best arms more quickly than conventional algorithms. In addition, we have a preliminary regret bound for the fixed $\mathbf{A}$ case.

The empirical results demonstrate that this algorithm achieves better regret than the state-of-the-art linear bandit algorithms, and performs well compared to another sparse bandit algorithm. Perhaps more impressively, the RVM-LTS algorithm achieves regret only slightly worse than the regret obtained by Thompson Sampling over the true sparse support of $\boldsymbol{\theta}^*$ for relatively small s. The algorithm responds to noise relatively well, and in the worst noise case appears to be no worse than standard Thompson Sampling in terms of regret.

From a practical standpoint, one disadvantage of the RVM-LTS method is the time complexity. Because $\mathbf{A}$ changes with time, rank-one updates to $\boldsymbol{\Sigma}_t^{-1}$ are no longer possible. Especially in the case where d is very large, this can lead to long computation times. Further work on the updating process which takes advantage of the structure of the RVM updates could yield computational benefits.

Finally, note that the RVM-LTS algorithm has better regret and higher rewards than other methods even when s is close to d or $\boldsymbol{\theta}^*$ was dense. As $\boldsymbol{\theta}^*$ becomes less sparse, the behavior of the RVM updates is less well understood. The fact that the algorithm, at least in simulations, outperforms standard linear bandit methods indicates that further investigation is warranted.

References

[1] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs." in *ICML (3)*, 2013, pp. 127–135.

[2] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[3] P. Auer and M. Long, "Using Confidence Bounds for Exploitation-Exploration Trade-offs," *Journal of Machine Learning Research*, vol. 3, p. 2002, 2002.

[4] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic Linear Optimization under Bandit Feedback." in *Proceedings of the Conference on Learning Theory (COLT)*, 2008, pp. 355–366.

[5] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly Parameterized Bandits," *Math. Oper. Res.*, vol. 35, no. 2, pp. 395–411, 2010.

[6] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari, "Improved Algorithms for Linear Stochastic Bandits," *Advances in Neural Information Processing Systems (NIPS)*, pp. 1–19, 2011.

[7] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem." in *COLT*, 2012, pp. 39–1.

[8] ——, "Thompson Sampling for Contextual Bandits with Linear Payoffs," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2013, pp. 127–135. [Online]. Available: http://jmlr.org/proceedings/papers/v28/agrawal13.html

[9] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari, "Online-to-confidence-set conversions and application to sparse stochastic bandits." in *AISTATS*, vol. 22, 2012, pp. 1–9.

[10] A. Carpentier and R. Munos, "Bandit theory meets compressed sensing for high dimensional stochastic linear bandit," in *AISTATS*, 2012, pp. 190–198.

[11] M. E. Tipping, A. C. Faul *et al.*, "Fast marginal likelihood maximisation for sparse bayesian models." in *AISTATS*, 2003.