Adaptive Feature Selection: Computationally Efficient Online Sparse Linear Regression under RIP

Satyen Kale [1] **Zohar Karnin** [2] **Tengyuan Liang** [3] **Dávid Pál** [4]

Abstract

Online sparse linear regression is an online problem where an algorithm repeatedly chooses a subset of coordinates to observe in an adversarially chosen feature vector, makes a real-valued prediction, receives the true label, and incurs the squared loss. The goal is to design an online learning algorithm with sublinear regret to the best sparse linear predictor in hindsight. Without any assumptions, this problem is known to be computationally intractable. In this paper, we make the assumption that data matrix satisfies restricted isometry property, and show that this assumption leads to computationally efficient algorithms with sublinear regret for two variants of the problem. In the first variant, the true label is generated according to a sparse linear model with additive Gaussian noise. In the second, the true label is chosen adversarially.

1. Introduction

In modern real-world sequential prediction problems, samples are typically high dimensional, and construction of the features may itself be a computationally intensive task. Therefore in sequential prediction, due to the computation and resource constraints, it is preferable to design algorithms that compute only a limited number of features for each new data example. One example of this situation, from (Cesa-Bianchi et al., 2011), is medical diagnosis of a disease, in which each feature is the result of a medical test on the patient. Since it is undesirable to subject a patient to a battery of medical tests, we would like to adaptively design diagnostic procedures that rely on only a few, highly informative tests.

Online sparse linear regression (OSLR) is a sequential prediction problem in which an algorithm is allowed to see only a small subset of coordinates of each feature vector. The problem is parameterized by 3 positive integers: d, the dimension of the feature vectors, k, the sparsity of the linear regressors we compare the algorithm's performance to, and k_0, a budget on the number of features that can be queried in each round by the algorithm. Generally we have $k \ll d$ and $k_0 \geq k$ but not significantly larger (our algorithms need[1] $k_0 = \tilde{O}(k)$).

In the OSLR problem, the algorithm makes predictions over a sequence of T rounds. In each round t, nature chooses a feature vector $x_t \in \mathbb{R}^d$, the algorithm chooses a subset of $\{1, 2, \ldots, d\}$ of size at most k' and observes the corresponding coordinates of the feature vector. It then makes a prediction $\widehat{y}_t \in \mathbb{R}$ based on the observed features, observes the true label y_t, and suffers loss $(y_t - \widehat{y}_t)^2$. The goal of the learner is to make the cumulative loss comparable to that of the best k-sparse linear predictor w in hindsight. The performance of the online learner is measured by the *regret*, which is defined as the difference between the two losses:

$$\mathrm{Regret}_T = \sum_{t=1}^{T} (y_t - \widehat{y}_t)^2 - \min_{w:\ \|w\|_0 \leq k} \sum_{t=1}^{T} (y_t - \langle x_t, w \rangle)^2 .$$

The goal is to construct algorithms that enjoy regret that is sub-linear in T, the total number of rounds. A sub-linear regret implies that in the asymptotic sense, the average per-round loss of the algorithm approaches the average per-round loss of the best k-sparse linear predictor.

Sparse regression is in general a computationally hard problem. In particular, given k, $x_1, x_2, \ldots, x_T$ and $y_1, y_2, \ldots, y_T$ as inputs, the offline problem of finding a k-sparse w that minimizes the error $\sum_{t=1}^{T} (y_t - \langle x_t, w \rangle)^2$ does not admit a polynomial time algorithm under standard complexity assumptions (Foster et al., 2015). This hard-

[1] Google Research, New York. [2] Amazon, New York. [3] University of Chicago, Booth School of Business, Chicago. [4] Yahoo Research, New York. Work done while the authors were at Yahoo Research, New York. Correspondence to: Satyen Kale <satyenkale@google.com>, Zohar Karnin <zkarnin@gmail.com>, Tengyuan Liang <Tengyuan.Liang@chicagobooth.edu>, Dávid Pál <dpal@yahoo-inc.com>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1] In this paper, we use the $\tilde{O}(\cdot)$ notation to suppress factors that are polylogarithmic in the natural parameters of the problem.

ness persists even under the assumption that there exists a k-sparse w^* such that $y_t = \langle x_t, w^* \rangle$ for all t. Furthermore, the computational hardness is present even when the solution is required to be only $\widetilde{\mathcal{O}}(k)$-sparse solution and has to minimize the error only approximately; see (Foster et al., 2015) for details. The hardness result was extended to online sparse regression by (Foster et al., 2016). They showed that for all $\delta > 0$ there exists no polynomial-time algorithm with regret $\mathcal{O}(T^{1-\delta})$ unless $NP \subseteq BPP$.

Foster et al. (2016) posed the open question of what additional assumptions can be made on the data to make the problem tractable. In this paper, we answer this open question by providing efficient algorithms with sublinear regret under the assumption that the matrix of feature vectors satisfies the *restricted isometry property* (RIP) (Candes & Tao, 2005). It has been shown that if RIP holds and there exists a sparse linear predictor w^* such that $y_t = \langle x_t, w^* \rangle + \eta_t$ where η_t is independent noise, the offline sparse linear regression problem admits computationally efficient algorithms, e.g., (Candes & Tao, 2007). RIP and related Restricted Eigenvalue Condition (Bickel et al., 2009) have been widely used as a standard assumption for theoretical analysis in the compressive sensing and sparse regression literature, in the offline case. In the online setting, it is natural to ask whether sparse regression avoids the computational difficulty under an appropriate form of the RIP condition. In this paper, we answer this question in a positive way, both in the realizable setting and in the agnostic setting. As a by-product, we resolve the adaptive feature selection problem as the efficient algorithms we propose in this paper adaptively choose a different "sparse" subset of features to query at each round. This is closely related to attribute-efficient learning (see discussion in Section 1.2) and online model selection.

1.1. Summary of Results

We design polynomial-time algorithms for online sparse linear regression for two models for the sequence $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$. The first model is called the *realizable* and the second is called *agnostic*. In both models, we assume that, after proper normalization, for all large enough t, the matrix X_t formed from the first t feature vectors $x_1, x_2, \ldots, x_t$ satisfies the restricted isometry property. The two models differ in the assumptions on y_t. The realizable model assumes that $y_t = \langle x_t, w^* \rangle + \eta_t$ where w^* is k-sparse and η_t is an independent noise. In the agnostic model, y_t can be arbitrary, and therefore, the regret bounds we obtain are worse than in the realizable setting. The models and corresponding algorithms are presented in Sections 2 and 3 respectively. Interestingly enough, the algorithms and their corresponding analyses are completely different in the realizable and agnostic case.

Our algorithms allow for somewhat more flexibility than the problem definition: they are designed to work with a budget k_0 on the number of features that can be queried that may be larger than the sparsity parameter k of the comparator. The regret bounds we derive improve with increasing values of k_0. In the case when $k_0 \approx k$, the dependence on d in the regret bounds is polynomial, as can be expected in limited feedback settings (this is analogous to polynomial dependence on d in *bandit* settings). In the extreme case when $k_0 = d$, i.e. we have access to *all* the features, the dependence on the dimension d in the regret bounds we prove is only *logarithmic*. The interpretation is that if we have full access to the features, but the goal is to compete with just k sparse linear regressors, then the number of data points that need to be seen to achieve good predictive accuracy has only logarithmic dependence on d. This is analogous to the (offline) compressed sensing setting where the sample complexity bounds, under RIP, only depend logarithmically on d.

A major building block in the solution for the realizable setting (Section 2) consists of identifying the best k-sparse linear predictor for the past data at any round in the prediction problem. This is done by solving a sparse regression problem on the observed data. The solution of this problem cannot be obtained by a simple application of say, the Dantzig selector (Candes & Tao, 2007) since we do not observe the data matrix X, but rather a subsample of its entries. Our algorithm is a variant of the Dantzig selector that incorporates random sampling into the optimization, and computes a near-optimal solution by solving a linear program. The resulting algorithm has a regret bound of $\widetilde{\mathcal{O}}(\log T)$. This bound has optimal dependence on T, since even in the full information setting where all features are observed there is a lower bound of $\Omega(\log T)$ (Hazan & Kale, 2014).

The algorithm for the agnostic setting relies on the theory of submodular optimization. The analysis in (Boutsidis et al., 2015) shows that the RIP assumption implies that the set function defined as the minimum loss achievable by a linear regressor restricted to the set in question satisfies a property called *weak supermodularity*. Weak supermodularity is a relaxation of standard supermodularity that is still strong enough to show performance bounds for the standard greedy feature selection algorithm for solving the sparse regression problem. We then employ a technique developed by Streeter & Golovin (2008) to construct an online learning algorithm that mimics the greedy feature selection algorithm. The resulting algorithm has a regret bound of $\widetilde{\mathcal{O}}(T^{2/3})$. It is unclear if this bound has the optimal dependence on T: it is easy to prove a lower bound of $\Omega(\sqrt{T})$ on the regret using standard arguments for the multiarmed bandit problem.

1.2. Related work

A related setting is attribute-efficient learning (Cesa-Bianchi et al., 2011; Hazan & Koren, 2012; Kukliansky & Shamir, 2015). This is a batch learning problem in which the examples are generated i.i.d., and the goal is to simply output a linear regressor using only a limited number of features per example with bounded excess risk compared to the optimal linear regressor, when given *full access* to the features at test time. Since the goal is not prediction but simply computing the optimal linear regressor, efficient algorithms exist and have been developed by the aforementioned papers.

Without any assumptions, only inefficient algorithms for the online sparse linear regression problem are known Zolghadr et al. (2013); Foster et al. (2016). Kale (2014) posed the open question of whether it is possible to design an efficient algorithm for the problem with a sublinear regret bound. This question was answered in the negative by Foster et al. (2016), who showed that efficiency can only be obtained under additional assumptions on the data. This paper shows that the RIP assumption yields tractability in the online setting just as it does in the batch setting.

In the realizable setting, the linear program at the heart of the algorithm is motivated from Dantzig selection (Candes & Tao, 2007) and error-in-variable regression (Rosenbaum & Tsybakov, 2010; Belloni et al., 2016). The problem of finding the best sparse linear predictor when only a sample of the entries in the data matrix is available is also discussed by Belloni et al. (2016) (see also the references therein). In fact, these papers solve a more general problem where we observe a matrix Z rather than X that is an unbiased estimator of X. While we can use their results in a black-box manner, they are tailored for the setting where the variance of each Z_{ij} is constant and it is difficult to obtain the exact dependence on this variance in their bounds. In our setting, this variance can be linear in the dimension of the feature vectors, and hence we wish to control the dependence on the variance in the bounds. Thus, we use an algorithm that is similar to the one in (Belloni et al., 2016), and provide an analysis for it (in the supplementary material). As an added bonus, our algorithm results in solving a linear program rather than a conic or general convex program, hence admits a solution that is more computationally efficient.

In the agnostic setting, the computationally efficient algorithm we propose is motivated from (online) supermodular optimization (Natarajan, 1995; Boutsidis et al., 2015; Streeter & Golovin, 2008). The algorithm is computationally efficient and enjoys sublinear regret under an RIP-like condition, as we will show in Section 3. This result can be contrasted with the known computationally prohibitive algorithms for online sparse linear regression (Zolghadr et al., 2013; Foster et al., 2016), and the hardness result without RIP (Foster et al., 2015; 2016).

1.3. Notation and Preliminaries

For $d \in \mathbb{N}$, we denote by $[d]$ the set $\{1, 2, \ldots, d\}$. For a vector in $x \in \mathbb{R}^d$, denote by $x(i)$ its i-th coordinate. For a subset $S \subseteq [d]$, we use the notation $\mathbb{R}^S$ to indicate the vector space spanned by the coordinate axes indexed by S (i.e. the set of all vectors w supported on the set S). For a vector $x \in \mathbb{R}^d$, denote by $x(S) \in \mathbb{R}^d$ the projection of x on $\mathbb{R}^S$. That is, the coordinates of $x(S)$ are

$$x(S)(i) = \begin{cases} x(i) & \text{if } i \in S, \\ 0 & \text{if } i \notin S, \end{cases} \qquad \text{for } i = 1, 2, \ldots, d.$$

Let $\langle u, v \rangle = \sum_i u(i) \cdot v(i)$ be the inner product of vectors u and v.

For $p \in [0, \infty]$, the ℓ_p-norm of a vector $x \in \mathbb{R}^d$ is denoted by $\|x\|_p$. For $p \in (0, \infty)$, $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$, $\|x\|_\infty = \max_i |x_i|$, and $\|x\|_0$ is the number of non-zero coordinates of x.

The following definition will play a key role:

Definition 1 (Restricted Isometry Property (Candes & Tao, 2007))**.** *Let $\epsilon \in (0, 1)$ and $k \geq 0$. We say that a matrix $X \in \mathbb{R}^{n \times d}$ satisfies* restricted isometry property *(RIP) with parameters (ϵ, k) if for any $w \in \mathbb{R}^d$ with $\|w\|_0 \leq k$ we have*

$$(1 - \epsilon) \|w\|_2 \leq \frac{1}{\sqrt{n}} \|Xw\|_2 \leq (1 + \epsilon) \|w\|_2 .$$

One can show that RIP holds with overwhelming probability if $n = \Omega(\epsilon^{-2} k \log(ed/k))$ and each row of the matrix is sampled independently from an isotropic sub-Gaussian distribution. In the realizable setting, the sub-Gaussian assumption can be relaxed to incorporate heavy tail distribution via the "small ball" analysis introduced in Mendelson (2014), since we only require one-sided lower isometry property.

1.4. Proper Online Sparse Linear Regression

We introduce a variant of online sparse regression (OSLR), which we call *proper online sparse linear regression (POSLR)*. The adjective "proper" is to indicate that the algorithm is required to output a weight vector in each round and its prediction is computed by taking an inner product with the feature vector.

We assume that there is an underlying sequence $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)$ of *labeled examples* in $\mathbb{R}^d \times \mathbb{R}$. In each round $t = 1, 2, \ldots, T$, the algorithm behaves according to the following protocol:

1. Choose a vector $w_t \in \mathbb{R}^d$ such that $\|w_t\|_0 \leq k$.

2. Choose $S_t \subseteq [d]$ of size at most k_0.

3. Observe $x_t(S_t)$ and y_t, and incur loss $(y_t - \langle x_t, w_t \rangle)^2$.

Essentially, the algorithm makes the prediction $\widehat{y}_t := \langle x_t, w_t \rangle$ in round t. The regret after T rounds of an algorithm with respect to $w \in \mathbb{R}^d$ is

$$\text{Regret}_T(w) = \sum_{t=1}^{T} (y_t - \langle x_t, w_t \rangle)^2 - \sum_{t=1}^{T} (y_t - \langle x_t, w \rangle)^2 \ .$$

The regret after T rounds of an algorithm with respect to the best k-sparse linear regressor is defined as

$$\text{Regret}_T = \max_{w:\ \|w\|_0 \le k} \text{Regret}_T(w) \, .$$

Note that any algorithm for POSLR gives rise to an algorithm for OSLR. Namely, if an algorithm for POSLR chooses w_t and S_t, the corresponding algorithm for OSLR queries the coordinates $S_t \cup \{i \ : w_t(i) \neq 0\}$. The algorithm for OSLR queries at most $k_0 + k$ coordinates and has the same regret as the algorithm for POSLR.

Additionally, POSLR allows parameters settings which do not have corresponding counterparts in OSLR. Namely, we can consider the sparse "full information" setting where $k_0 = d$ and $k \ll d$.

We denote by X_t the $t \times d$ matrix of first t unlabeled samples i.e. rows of X_t are $x_1^T, x_2^T, \ldots, x_t^T$. Similarly, we denote by $Y_t \in \mathbb{R}^t$ the vector of first t labels $y_1, y_2, \ldots, y_t$. We use the shorthand notation X, Y for X_T and Y_T respectively.

In order to get computationally efficient algorithms, we assume that that for all $t \ge t_0$, the matrix X_t satisfies the restricted isometry condition. The parameter t_0 and RIP parameters k, ϵ will be specified later.

2. Realizable Model

In this section we design an algorithm for POSLR for the realizable model. In this setting we assume that there is a vector $w^* \in \mathbb{R}^d$ such that $\|w^*\|_0 \le k$ and the sequence of labels $y_1, y_2, \ldots, y_T$ is generated according to the linear model

$$y_t = \langle x_t, w^* \rangle + \eta_t \ , \tag{1}$$

where $\eta_1, \eta_2, \ldots, \eta_T$ are independent random variables from $N(0, \sigma^2)$. We assume that the standard deviation σ, or an upper bound of it, is given to the algorithm as input. We assume that $\|w^*\|_1 \le 1$ and $\|x_t\|_\infty \le 1$ for all t.

For convenience, we use η to denote the vector $(\eta_1, \eta_2, \ldots, \eta_T)$ of noise variables.

2.1. Algorithm

The algorithm maintains an unbiased estimate $\widehat{X}_t$ of the matrix X_t. The rows of $\widehat{X}_t$ are vectors $\widehat{x}_1^T, \widehat{x}_2^T, \ldots, \widehat{x}_t^T$ which are unbiased estimates of $x_1^T, x_2^T, \ldots, x_t^T$. To construct the estimates, in each round t, the set $S_t \subseteq [d]$ is chosen uniformly at random from the collection of all subsets of $[d]$ of size k_0. The estimate is

$$\widehat{x}_t = \frac{d}{k_0} \cdot x_t(S_t). \tag{2}$$

To compute the predictions of the algorithm, we consider the linear program

$$\text{minimize } \|w\|_1 \text{ s.t. } \left\| \frac{1}{t} \widehat{X}_t^T \left(Y_t - \widehat{X}_t w \right) + \frac{1}{t} \widehat{D}_t w \right\|_\infty \le C \sqrt{\frac{d \log(td/\delta)}{t k_0}} \left(\sigma + \frac{d}{k_0} \right). \tag{3}$$

Here, $C > 0$ is a universal constant, and $\delta \in (0,1)$ is the allowed failure probability. $\widehat{D}_t$, defined in equation (5), is a diagonal matrix that offsets the bias on the $\text{diag}(\widehat{X}_t^T \widehat{X}_t)$.

The linear program (3) is called the Dantzig selector. We denote its optimal solution by $\widehat{w}_{t+1}$. (We define $\widehat{w}_1 = 0$.)

Based on $\widehat{w}_t$, we construct $\widetilde{w}_t \in \mathbb{R}^d$. Let $|\widehat{w}_t(i_1)| \ge |\widehat{w}_t(i_2)| \ge \cdots \ge |\widehat{w}_t(i_d)|$ be the coordinates sorted according to the their absolute value, breaking ties according to their index. Let $\widetilde{S}_t = \{i_1, i_2, \ldots, i_k\}$ be the top k coordinates. We define $\widetilde{w}_t$ as

$$\widetilde{w}_t = \widehat{w}_t(\widetilde{S}_t). \tag{4}$$

The actual prediction w_t is either zero if $t \le t_0$ or $\widetilde{w}_s$ for some $s \le t$ and it gets updated whenever t is a power of 2.

The algorithm queries at most $k + k_0$ features each round, and the linear program can be solved in polynomial time using simplex method or interior point method. The algorithm solves the linear program only $\lceil \log_2 T \rceil$ times by using the same vector in the rounds $2^s, \ldots, 2^{s+1} - 1$. This lazy update improves both the computational aspects of the algorithm and the regret bound.

2.2. Main Result

The main result in this section provides a logarithmic regret bound under the following assumptions [2]

- The feature vectors have the property that for any $t \ge t_0$, the matrix X_t satisfies the RIP condition with $(\frac{1}{5}, 3k)$, with $t_0 = \mathcal{O}(k \log(d) \log(T))$.

[2] A more precise statement with the exact dependence on the problem parameters can be found in the supplementary material.

Algorithm 1 Dantzig Selector for POSLR

```
Require: $T$, $\sigma$, $t_0$, $k$, $k_0$
for $t = 1, 2, \dots, T$ do
    if $t \le t_0$ then
        Predict $w_t = 0$
    else if $t$ is a power of 2 then
        Let $\widehat{w}_t$ be the solution of linear program (3)
        Compute $\widetilde{w}_t$ according to (4)
        Predict $w_t = \widetilde{w}_t$
    else
        Predict $w_t = w_{t-1}$
    end if
    Let $S_t \subseteq [d]$ be a random subset of size $k_0$
    Observe $x_t(S_t)$ and $y_t$
    Construct estimate $\widehat{x}_t$ according to (2)
    Append $\widehat{x}_t^T$ to $\widehat{X}_{t-1}$ to form $\widehat{X}_t \in \mathbb{R}^{t \times d}$
end for
```

- The underlying POSLR online prediction problem has a sparsity budget of k and observation budget k_0.
- The model is realizable as defined in equation (1) with i.i.d unbiased Gaussian noise with standard deviation $\sigma = \mathcal{O}(1)$.

Theorem 2. *For any $\delta > 0$, with probability at least $1 - \delta$, Algorithm 1 satisfies*

$$\text{Regret}_T = \mathcal{O}\left(k^2 \log(d/\delta)(d/k_0)^3 \log(T)\right).$$

The theorem asserts that an $\mathcal{O}(\log T)$ regret bound is efficiently achievable in the realizable setting. Furthermore when $k_0 = \Omega(d)$ the regret scales as $\log(d)$ meaning that we do not necessarily require $T \ge d$ to obtain a meaningful result. We note that the complete expression for arbitrary t_0, σ is given in (13) in the supplementary material.

The algorithm can be easily understood via the error-in-variable equation

$$\begin{aligned} y_t &= \langle x_t, w^* \rangle + \eta_t\ , \\ \widehat{x}_t &= x_t + \xi_t. \end{aligned}$$

with $\mathbf{E}[\xi_t] = \mathbf{E}[\widehat{x}_t - x_t] = 0$, where the expectation is taken over random sampling introduced by the algorithm when performing feature exploration. The learner observes y_t as well as the "noisy" feature vector $\widehat{x}_t$, and aims to recover w^*.

As mentioned above, we (implicitly) need an unbiased estimator of $X_t^T X_t$. By taking $\widehat{X}_t^T \widehat{X}_t$ it is easy to verify that the off-diagonal entries are indeed unbiased however this is not the case for the diagonal. To this end we define $D_t \in \mathbb{R}^{d \times d}$ as the diagonal matrix compensating for the sampling bias on the diagonal elements of $\widehat{X}_t^T \widehat{X}_t$

$$D_t = \left(\frac{d}{k_0} - 1\right) \cdot \text{diag}\left(X_t^T X_t\right)$$

and the estimated bias from the observed data is

$$\widehat{D}_t = \left(1 - \frac{k_0}{d}\right) \cdot \text{diag}\left(\widehat{X}_t^T \widehat{X}_t\right). \tag{5}$$

Therefore, program (1) can be viewed as Dantzig selector with plug-in unbiased estimates for $X_t^T Y_t$ and $X_t^T X_t$ using limited observed features.

2.3. Sketch of Proof

The main building block in proving Theorem 2 is stated in Lemma 3. It proves that the sequence of solutions $\widehat{w}_t$ converges to the optimal response w^* based on which the signal y_t is created. More accurately, ignoring all second order terms, it shows that $\|\widehat{w}_t - w^*\|_1 \le \mathcal{O}(1/\sqrt{t})$. In Lemma 4 we show that the same applies for the sparse approximation w_t of $\widehat{w}_t$. Now, since $\|x_t\|_\infty \le 1$ we get that the difference between our response $\langle x_t, w_t \rangle$ and the (almost) optimal response $\langle x_t, w^* \rangle$ is bounded by $1/\sqrt{t}$. Given this, a careful calculation of the difference of losses leads to a regret bound w.r.t. w^*. Specifically, an elementary analysis of the loss expression leads to the equality

$$\text{Regret}_T(w^*) = \sum_{t=1}^{T} 2\eta_t \langle x_t, w^* - w_t \rangle + (\langle x_t, w^* - w_t \rangle)^2$$

A bound on both summands can clearly be expressed in terms of $|\langle x_t, w^* - w_t \rangle| = \mathcal{O}(1/\sqrt{t})$. The right summand requires a martingale concentration bound and the left is trivial. For both we obtain a bound of $\mathcal{O}(\log(T))$.

We are now left with two technicalities. The first is that w^* is not necessarily the empirically optimal response. To this end we provide, in Lemma 16 in the supplementary material, a constant (independent of T) bound on the regret of w^* compared to the empirical optimum. The second technicality is the fact that we do not solve for $\widehat{w}_t$ in every round, but in exponential gaps. This translates to an added factor of 2 to the bound $\|w_t - w^*\|_1$ that affects only the constants in the $\mathcal{O}(\cdot)$ terms.

Lemma 3 (Estimation Rates)**.** *Assume that the matrix $X_t \in \mathbb{R}^{t \times d}$ satisfies the RIP condition with $(\epsilon, 3k)$ for some $\epsilon < 1/5$. Let $\widehat{w}_{n+1} \in \mathbb{R}^d$ be the optimal solution of program (3). With probability at least $1 - \delta$,*

$$\|\widehat{w}_{t+1} - w^*\|_2 \le C \cdot \sqrt{\frac{d}{k_0} \cdot \frac{k \log(d/\delta)}{t}} \left(\sigma + \frac{d}{k_0}\right),$$

$$\|\widehat{w}_{t+1} - w^*\|_1 \le C \cdot \sqrt{\frac{d}{k_0} \frac{k^2 \log(d/\delta)}{t}} \left(\sigma + \frac{d}{k_0}\right).$$

Here $C > 0$ is some universal constant and σ is the standard deviation of the noise.

Note the $\widehat{w}_t$ may not be sparse; it can have many non-zero coordinates that are small in absolute value. However, we take the top k coordinates of $\widehat{w}_t$ in absolute value. Thanks to the Lemma 4 below, we lose only a constant factor $\sqrt{3}$.

Lemma 4. *Let $\widehat{w} \in \mathbb{R}^d$ be an arbitrary vector and let $w^* \in \mathbb{R}^d$ be a k-sparse vector. Let $\widetilde{S} \subseteq [d]$ be the top k coordinates of $\widehat{w}$ in absolute value. Then,*

$$\left\|\widehat{w}(\widetilde{S}) - w^*\right\|_2 \leq \sqrt{3}\,\|\widehat{w} - w^*\|_2\,.$$

3. Agnostic Setting

In this section we focus on the agnostic setting, where we don't impose any distributional assumption on the sequence. In this setting, there is no "true" sparse model, but the learner — with limited access to features — is competing with the best k-sparse model defined using full information $\{(x_t, y_t)\}_{t=1}^T$.

As before, we do assume that x_t and y_t are bounded. Without loss of generality, $\|x_t\|_\infty \leq 1$, and $|y_t| \leq 1$ for all t. Once again, without any regularity condition on the design matrix, Foster et al. (2016) have shown that achieving a sub-linear regret $\mathcal{O}(T^{1-\delta})$ is in general computationally hard, for any constant $\delta > 0$ unless NP $\subseteq$ BPP.

We give an efficient algorithm that achieves sub-linear regret under the assumption that the design matrix of any (sufficiently long) block of consecutive data points has bounded *restricted condition number*, which we define below:

Definition 5 (Restricted Condition Number)**.** *Let $k \in \mathbb{N}$ be a sparsity parameter. The* restricted condition number for sparsity *k of a matrix $X \in \mathbb{R}^{n\times d}$ is defined as*

$$\sup_{\substack{v,w:\ \|v\|=\|w\|=1,\\ \|v\|_0, \|w\|_0 \leq k}} \frac{\|Xv\|}{\|Xw\|}.$$

It is easy to see that if a matrix X satisfies RIP with parameters (ϵ, k), then its restricted condition number for sparsity k is at most $\frac{1+\epsilon}{1-\epsilon}$. Thus, having bounded restricted condition number is a weaker requirement than RIP.

We now define the *Block Bounded Restricted Condition Number Property* (BBRCNP):

Definition 6 (Block Bounded Restricted Condition Number Property)**.** *Let $\kappa > 0$ and $k \in \mathbb{N}$. A sequence of feature vectors $x_1, x_2, \ldots, x_T$ satisfies* BBRCNP *with parameters (κ, K) if there is a constant t_0 such that for any sequence of consecutive time steps $\mathcal{T}$ with $|\mathcal{T}| \geq t_0$, the restricted condition number for sparsity k of X, the design matrix of the feature vectors x_t for $t \in \mathcal{T}$, is at most κ.*

Note that in the random design setting where x_t, for $t \in [T]$, are isotropic sub-Gaussian vectors, $t_0 = O(\log T + k \log d)$ suffices to satisfy BBRCNP with high probability, where the $O(\cdot)$ notation hides a constant depending on κ.

We assume in this section that the sequence of feature vectors satisfies BBRCNP with parameters (κ, K) for some $K = \mathcal{O}(k \log(T))$ to be defined in the course of the analysis.

3.1. Algorithm

The algorithm in the agnostic setting is of distinct nature from that in the stochastic setting. Our algorithm is motivated from literature on maximization of sub-modular set function (Natarajan, 1995; Streeter & Golovin, 2008; Boutsidis et al., 2015). Though the problem being NP-hard, greedy algorithm on sub-modular maximization provides provable good approximation ratio. Specifically, (Streeter & Golovin, 2008) considered online optimization of super/sub-modular set functions using expert algorithm as sub-routine. (Natarajan, 1995; Boutsidis et al., 2015) cast the sparse linear regression as maximization of weakly supermodular function. We will introduce an algorithm that blends various ideas from referred literature, to attack the online sparse regression with limited features.

First, let's introduce the notion of a weakly supermodular function.

Definition 7. *For parameters $k \in \mathbb{N}$ and $\alpha \geq 1$, a set function $g : [d] \to \mathbb{R}$ is (k, α)-weakly supermodular if for any two sets $S \subseteq T \subseteq [d]$ with $|T| \leq k$, the following two inequalities hold:*

1. **(monotonicity)** *$g(T) \leq g(S)$, and*
2. **(approximately decreasing marginal gain)**

$$g(S) - g(T) \leq \alpha \sum_{i \in T\setminus S} [g(S) - g(S \cup \{i\})].$$

The definition is slightly stronger than that in (Boutsidis et al., 2015). We will show that sparse linear regression can be viewed as weakly supermodular minimization in Definition 7 once the design matrix has bounded restricted condition number.

Now we outline the algorithm (see Algorithm 2). We divide the rounds $1, 2, \ldots, T$ into mini-batches of size B each (so there are T/B such batches). The b-th batch thus consists of the examples (x_t, y_t) for $t \in \mathcal{T}_b := \{(b-1)B + 1, (b-1)B + 1, \ldots, bB\}$. Within the b-th batch, our algorithm queries the same subset of features of size at most k_0.

The algorithm consists of few key steps. First, one can show that under BBRCNP, as long as B is large enough, the loss within batch b defines a weakly supermodular set function

$$g_t(S) = \frac{1}{B} \inf_{w \in \mathbb{R}^S} \sum_{t \in \mathcal{T}_b} (y_t - \langle x_t, w \rangle)^2.$$

Therefore, we can formulate the original online sparse regression problem into online weakly supermodular minimization problem. For the latter problem, we develop an online greedy algorithm along the lines of (Streeter & Golovin, 2008). We employ $k_1 = \mathcal{O}^*(k)$ budgeted experts algorithms (Amin et al., 2015), denoted BEXP, with budget parameter[3] $\frac{k_0}{k_1}$. The precise characteristics of BEXP are given in Theorem 8 (adapted from Theorem 2 in (Amin et al., 2015)).

Theorem 8. *For the problem of prediction from expert advice, let there be d experts, and let $k \in [d]$ be a budget parameter. In each prediction round t, the BEXP algorithm chooses an expert j_t and a set of experts U_t containing j_t of size at most k, obtains as feedback the losses of all the experts in U_t, suffers the loss of expert j_t, and guarantees an expected regret bound of $2\sqrt{\frac{d\log(d)}{k}T}$ over T prediction rounds.*

At the beginning of each mini-batch b, the BEXP algorithms are run. Each BEXP algorithm outputs a set of coordinates of size $\frac{k_0}{k_1}$ as well as a special coordinate in that set. The union of all of these sets is then used as the set of features to query throughout the subsequent mini-batch. Within the mini-batch, the algorithm runs the standard Vovk-Azoury-Warmuth algorithm for linear prediction with square loss *restricted* to set of special coordinates output by all the BEXP algorithms.

At the end of the mini-batch, every BEXP algorithm is provided carefully constructed losses for each coordinate that was output as feedback. These losses ensure that the set of special coordinates chosen by the BEXP algorithms mimic the greedy algorithm for weakly supermodular minimization.

3.2. Main Result

In this section, we will show that Algorithm 2 achieves sublinear regret under BBRCNP.

Theorem 9. *Suppose the sequence of feature vectors satisfies BBRCNP with parameters $(\kappa, k_1 + k)$ for $k_1 = \frac{1}{3}\kappa^2 k \log(T)$, and assume that T is large enough so that $t_0 \le (\frac{k_0 T}{\kappa^2 dk})^{1/3}$. Then if Algorithm 2 is run with parameters $B = (\frac{k_0 T}{\kappa^2 dk})^{1/3}$ and k_1 as specified above, its expected regret is at most $\tilde{O}((\frac{\kappa^8 dk^4}{k_0})^{1/3} T^{2/3})$.*

Proof. The proof relies on a number of lemmas whose proofs can be found in the supplementary material. We begin with the connection between sparse linear regression, weakly supermodular function and RIP, formally stated in Lemma 10. This lemma is a direct consequence of Lemma 5 in (Boutsidis et al., 2015).

Algorithm 2 Online Greedy Algorithm for POSLR

```
Require: Mini-batch size $B$, sparsity parameters $k_0$ and $k_1$
Set up $k_1$ budgeted prediction algorithms $\mathsf{BEXP}^{(i)}$ for $i \in [k_1]$, each using the coordinates in $[d]$ as "experts" with a per-round budget of $\frac{k_0}{k_1}$.
for $b = 1, 2, \ldots, T/B$ do
    For each $i \in [k_1]$, obtain a coordinate $j_b^{(i)}$ and subset of coordinates $U_b^{(i)}$ from $\mathsf{BEXP}^{(i)}$ such that $j_b^{(i)} \in U_b^{(i)}$.
    Define $V_b^{(0)} = \emptyset$ and for each $i \in [k_1]$ define $V_b^{(i)} = \{j_b^{(i')} \mid i' \le i\}$.
    Set up the Vovk-Azoury-Warmuth ($\mathsf{VAW}$) algorithm for predicting using the features in $V_b^{(k_1)}$.
    for $t \in \mathcal{T}_b$ do
        Set $S_t = \bigcup_{i \in [k_1]} U_b^{(i)}$, obtain $x_t(S_t)$, and pass $x_t(V_b^{(k_1)})$ to $\mathsf{VAW}$.
        Set $w_t$ to be the weight vector output by $\mathsf{VAW}$.
        Obtain the true label $y_t$ and pass it to $\mathsf{VAW}$.
    end for
    Define the function
        $$g_b(S) = \frac{1}{B} \inf_{w \in \mathbb{R}^S} \sum_{t \in \mathcal{T}_b} (y_t - \langle x_t, w \rangle)^2. \quad (6)$$
    For each $j \in U_b^{(i)}$, compute $g_b(V_b^{(i-1)} \cup \{j\})$ and pass it $\mathsf{BEXP}^{(i)}$ as the loss for expert $j$.
end for
```

Lemma 10. *Consider a sequence of examples $(x_t, y_t) \in \mathbb{R}^d \times \mathbb{R}$ for $t = 1, 2, \ldots, B$, and let X be the design matrix for the sequence. Consider the set function associated with least squares optimization:*

$$g(S) = \inf_{w \in \mathbb{R}^S} \frac{1}{B} \sum_{t=1}^{B} (y_t - \langle x_t, w \rangle)^2.$$

Suppose the restricted condition number of X for sparsity k is bounded by κ. Then $g(S)$ is (k, κ^2)-weakly supermodular.

Even though minimization of weakly supermodular functions is NP-hard, the greedy algorithm provides a good approximation, as shown in the next lemma.

Lemma 11. *Consider a (k, α)-weakly supermodular set function $g(\cdot)$. Let $j^* := \arg\min_j g(\{j\})$. Then, for any*

[3] We assume, for convenience, that k_0 is divisible by k_1.

subset V of size at most k, we have

$$g(\{j^*\}) - g(V) \leq \left(1 - \tfrac{1}{\alpha|V|}\right)[g(\emptyset) - g(V)].$$

The BEXP algorithms essentially implement the greedy algorithm in an online fashion. Using the properties of the BEXP algorithm, we have the following regret guarantee:

Lemma 12. *Suppose the sequence of feature vectors satisfies BBRCNP with parameters $(\epsilon, k_1 + k)$. Then for any set V of coordinates of size at most k, we have*

$$\mathbf{E}\left[\sum_{b=1}^{T/B} g_b(V_b^{(k_1)}) - g_b(V)\right]$$
$$\leq \sum_{b=1}^{T/B}\left(1 - \tfrac{1}{\kappa^2|V|}\right)^{k_1}[g_b(\emptyset) - g_b(V)] + 2\kappa^2 k\sqrt{\tfrac{dk_1\log(d)T}{k_0 B}}.$$

Finally, within every mini-batch, the VAW algorithm guarantees the following regret bound, an immediate consequence of Theorem 11.8 in Cesa-Bianchi & Lugosi (2006):

Lemma 13. *Within every batch b, the VAW algorithm generates weight vectors w_t for $t \in \mathcal{T}_b$ such that*

$$\sum_{t\in\mathcal{T}_b}(y_t - \langle x_t, w_t\rangle)^2 - Bg_b(V_b^{(k_1)}) \leq O(k_1\log(B)).$$

We can now prove Theorem 9. Combining the bounds of lemma 12 and 13, we conclude that for any subset of coordinates V of size at most k, we have

$$\mathbf{E}\left[\sum_{t=1}^{T}(y_t - \langle x_t, w_t\rangle)^2\right] \tag{7}$$
$$\leq \sum_{b=1}^{T/B} Bg_b(V) + B(1 - \tfrac{1}{\kappa^2|V|})^{k_1}[g_b(\emptyset) - g_b(V)] \tag{8}$$
$$+ O\left(\kappa^2 k\sqrt{\tfrac{dk_1\log(d)BT}{k_0}} + \frac{T}{B}k_1\log(B)\right). \tag{9}$$

Finally, note that

$$\sum_{b=1}^{T/B} Bg_b(V) \leq \inf_{w\in\mathbb{R}^V}\sum_{t=1}^{T}(y_t - \langle x_t, w\rangle)^2,$$

and

$$\sum_{b=1}^{T/B} B(1 - \tfrac{1}{\kappa^2|V|})^{k_1}[g_b(\emptyset) - g_b(V)] \leq T\cdot\exp(-\tfrac{k_1}{\kappa^2 k}),$$

because $g_b(\emptyset) \leq 1$. Using these bounds in (9), and plugging in the specified values of B and k_1, we get the stated regret bound. □

4. Conclusions and Future Work

In this paper, we gave computationally efficient algorithms for the online sparse linear regression problem under the assumption that the design matrices of the feature vectors satisfy RIP-type properties. Since the problem is hard without any assumptions, our work is the first one to show that assumptions that are similar to the ones used to sparse recovery in the batch setting yield tractability in the online setting as well.

Several open questions remain in this line of work and will be the basis for future work. Is it possible to improve the regret bound in the agnostic setting? Can we give matching lower bounds on the regret in various settings? Is it possible to relax the RIP assumption on the design matrices and still have efficient algorithms? Some obvious weakenings of the RIP assumption we have made don't yield tractability. For example, simply assuming that the final matrix X_T satisfies RIP rather than every intermediate matrix X_t for large enough t is not sufficient; a simple tweak to the lower bound construction of Foster et al. (2016) shows this. This tweak consists of simply padding the construction with enough dummy examples which are well-conditioned enough to overcome the ill-conditioning of the original construction so that RIP is satisfied by X_T. We note however that in the realizable setting, our analysis can be easily adapted to work under weaker conditions such as irrepresentability (Zhao & Yu, 2006; Javanmard & Montanari, 2013).

References

Amin, Kareem, Kale, Satyen, Tesauro, Gerald, and Turaga, Deepak S. Budgeted prediction with expert advice. In *AAAI*, pp. 2490–2496, 2015.

Belloni, Alexandre, Rosenbaum, Mathieu, and Tsybakov, Alexandre B. Linear and conic programming estimators in high dimensional errors-in-variables models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016. ISSN 1467-9868.

Bickel, Peter J, Ritov, Ya'acov, and Tsybakov, Alexandre B. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, pp. 1705–1732, 2009.

Boutsidis, Christos, Liberty, Edo, and Sviridenko, Maxim. Greedy minimization of weakly supermodular set functions. *arXiv preprint arXiv:1502.06528*, 2015.

Candes, Emmanuel and Tao, Terence. The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, pp. 2313–2351, 2007.

Candes, Emmanuel J and Tao, Terence. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

Cesa-Bianchi, Nicolò and Lugosi, Gábor. *Prediction, learning, and games*. Cambridge University Press, 2006.

Cesa-Bianchi, Nicolò, Shalev-Shwartz, Shai, and Shamir, Ohad. Efficient learning with partially observed attributes. *Journal of Machine Learning Research*, 12 (Oct):2857–2878, 2011.

Foster, Dean, Karloff, Howard, and Thaler, Justin. Variable selection is hard. In *COLT*, pp. 696–709, 2015.

Foster, Dean, Kale, Satyen, and Karloff, Howard. Online sparse linear regression. In *COLT*, 2016.

Hazan, Elad and Kale, Satyen. Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. *Journal of Machine Learning Research*, 15(1):2489–2512, 2014.

Hazan, Elad and Koren, Tomer. Linear regression with limited observation. In *ICML*, 2012.

Javanmard, Adel and Montanari, Andrea. Model selection for high-dimensional regression under the generalized irrepresentability condition. In *NIPS*, pp. 3012–3020, 2013.

Kale, Satyen. Open problem: Efficient online sparse regression. In *COLT*, pp. 1299–1301, 2014.

Kukliansky, Doron and Shamir, Ohad. Attribute efficient linear regression with distribution-dependent sampling. In *ICML*, pp. 153–161, 2015.

Mendelson, Shahar. Learning without concentration. In *COLT*, pp. 25–39, 2014.

Natarajan, Balas Kausik. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

Rosenbaum, Mathieu and Tsybakov, Alexandre B. Sparse recovery under matrix uncertainty. *The Annals of Statistics*, 38(5):2620–2651, 2010.

Streeter, Matthew J. and Golovin, Daniel. An online algorithm for maximizing submodular functions. In *NIPS*, pp. 1577–1584, 2008.

Zhao, Peng and Yu, Bin. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov): 2541–2563, 2006.

Zolghadr, Navid, Bartók, Gábor, Greiner, Russell, György, András, and Szepesvári, Csaba. Online learning with costly features and labels. In *NIPS*, pp. 1241–1249, 2013.