JMLR: Workshop and Conference Proceedings vol 49:1–11, 2016

Online Sparse Linear Regression

Dean Foster
Amazon
DEAN@FOSTER.NET

Satyen Kale
Yahoo Research
SATYEN@YAHOO-INC.COM

Howard Karloff
HOWARD@CC.GATECH.EDU

Abstract

We consider the online sparse linear regression problem, which is the problem of sequentially making predictions observing only a limited number of features in each round, to minimize regret with respect to the best sparse linear regressor, where prediction accuracy is measured by square loss. We give an *inefficient* algorithm that obtains regret bounded by $\tilde{O}(\sqrt{T})$ after T prediction rounds. We complement this result by showing that no algorithm running in polynomial time per iteration can achieve regret bounded by $O(T^{1-\delta})$ for any constant $\delta > 0$ unless $\mathsf{NP} \subseteq \mathsf{BPP}$. This computational hardness result resolves an open problem presented in COLT 2014 (Kale, 2014) and also posed by Zolghadr et al. (2013). This hardness result holds even if the algorithm is allowed to access more features than the best sparse linear regressor up to a logarithmic factor in the dimension.

1. Introduction

In various real-world scenarios, features for examples are constructed by running some computationally expensive algorithms. With resource constraints, it is essential to be able to make predictions with only a limited number of features computed per example. One example of this scenario, from (Cesa-Bianchi et al., 2011), is medical diagnosis of a disease, in which each feature corresponds to a medical test that the patient in question can undergo. Evidently, it is undesirable to subject a patient to a battery of medical tests, for medical as well as cost reasons. Another example from the same paper is a search engine, where a ranking of web pages must be generated for each incoming user query and the limited amount of time allowed to answer a query imposes restrictions on the number of attributes that can be evaluated in the process. In both of these problems, predictions need to be made sequentially as patients or search queries arrive online, learning a good model in the process.

In this paper, we model the problem of prediction with limited access to features in the most natural and basic manner as an online sparse linear regression problem. In this problem, an online learner makes real-valued predictions for the labels of examples arriving sequentially over a number of rounds. Each example has d features that can be potentially accessed by the learner. However, in each round, the learner is restricted to choosing an arbitrary subset of features of size at most k, a budget parameter. The learner then acquires the values of the subset of features, and then makes its prediction, at which point the true label of the example is revealed to the learner. The learner suffers

a loss for making an incorrect prediction (for simplicity, we use square loss in this paper). The goal of the learner is to make predictions with total loss comparable to the loss of the best sparse linear regressor with a bounded norm, where the term *sparse* refers to the fact that the linear regressor has nonzero weights on at most k features. To measure the performance of the online learner, we use the standard notion of *regret*, which is the difference between the total loss of the online learner and the total loss of the best sparse linear regressor.

While regret is the primary performance metric, we are also interested in *efficiency* of the online learner. Ideally, we desire an online learning algorithm that minimizes regret while making predictions *efficiently*, i.e., in polynomial time (as a function of d and T). In this paper, we prove that this goal is impossible unless there is a randomized polynomial-time algorithm for deciding satisfiability of 3CNF formulas, the canonical NP-hard problem. This computational hardness result resolves open problems from (Kale, 2014) and (Zolghadr et al., 2013). In fact, the computational hardness persists even if the online learner is given the additional flexibility of choosing $k' = D\log(d)k$ features for any constant $D > 0$. In light of this result, in this paper we also give an *inefficient* algorithm for the problem which queries $k' \geq k+2$ features in each round, that runs in $O(\binom{d}{k}k')$ time per round, and that obtains regret bounded by $O(\frac{d^2}{(k'-k)^2}\sqrt{k\log(d)T})$.

2. Related Work and Known Results

A related setting is attribute-efficient learning (Cesa-Bianchi et al., 2011; Hazan and Koren, 2012; Kukliansky and 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. While the aforementioned papers give efficient, near-optimal algorithms for this problem, these algorithms do not work in the online sparse regression setting in which we are interested because here we are required to make predictions using only a limited number of features.

In (Kale, 2014), a simple algorithm has been suggested, which is based on running a bandit algorithm in which the actions correspond to selecting one of $\binom{d}{k}$ subsets of coordinates of size k at regular intervals, and within each interval, running an online regression algorithm (such as the Online Newton-Step algorithm of Hazan et al. (2007)) over the k coordinates chosen by the bandit algorithm. This algorithm, with the right choice of interval lengths, has a regret bound of $O(k^2d^{k/3}T^{2/3}\log(T/d))$. The algorithm has exponential dependence on k both in running time and the regret. Also, Kale (2014) sketches a different algorithm with performance guarantees similar to the algorithm presented in this paper; our work builds upon that sketch and gives tighter regret bounds.

Zolghadr et al. (2013) consider a very closely related setting (called *online probing*) in which features and labels may be obtained by the learner at some cost (which may be different for different features), and this cost is factored into the loss of the learner. In the special case of their setting corresponding to the problem considered here, they given an algorithm, LQDEXP3, which relies on discretizing all k-sparse weight vectors and running an exponential-weights experts algorithm on the resulting set with stochastic loss estimators, obtaining a $O(\sqrt{dT})$ regret bound. However the running time of their algorithm is prohibitive: $O((dT)^{O(k)})$ time per iteration. In the same paper, they pose the open problem of finding a computationally efficient no-regret algorithm for the problem. The hardness result in this paper resolves this open problem.

In the batch, binary-classification setting, a related model of learning called *Restricted Focus-of-Attention* has also been studied (Ben-David and Dichterman, 1993; Birkendorf et al., 1998).

On the computational hardness side, it is known that it is NP-hard to compute the optimal sparse linear regressor (Foster et al., 2015; Natarajan, 1995). The hardness result in this paper is in fact inspired by the work of Foster et al. (2015), who proved that it is computationally hard to find even an approximately optimal sparse linear regressor for an ordinary least squares regression problem given a batch of labeled data. While these results imply that it is hard to *properly*[1] solve the offline problem, in the online setting we allow *improper* learning, and hence these prior results don't yield hardness results for the online problem considered in this paper.

3. Notation and Setup

We use the notation $[d] = \{1, 2, \ldots, d\}$ to refer to the coordinates. All vectors in this paper are in $\mathbb{R}^d$, and all matrices in $\mathbb{R}^{d \times d}$. For a subset S of $[d]$, and a vector $\mathbf{x}$, we use the notation $\mathbf{x}(S)$ to denote the projection of $\mathbf{x}$ on the coordinates indexed by S. We also use the notation $\mathbf{I}_S$ to denote the diagonal matrix which has ones in the coordinates indexed by S and zeros elsewhere: this is the identity matrix on the subspace of $\mathbb{R}^d$ induced by the coordinates in S, as well as the projection matrix for this subspace. We use the notation $\|\cdot\|$ to denote the ℓ_2 norm in $\mathbb{R}^d$ and $\|\cdot\|_0$ to denote the zero "norm," i.e., the number of nonzero coordinates.

We consider a prediction problem in which the examples are vectors in $\mathbb{R}^d$ with ℓ_2 norm bounded by 1, and labels are in the range $[-1, 1]$. We use square loss to measure the accuracy of a prediction: i.e., for a labeled example $(\mathbf{x}, y) \in \mathbb{R}^d \times [-1, 1]$, the loss of a prediction $\hat{y}$ is $(\hat{y} - y)^2$. The learner's task is to make predictions online as examples arrive one by one based on observing only k out of d features of the learner's choosing on any example (the learner is allowed to choose different subsets of features to observe in each round). The learner's goal is to minimize regret relative to the best k-sparse linear regressor whose ℓ_2 norm is bounded by 1.

Formally, for $t = 1, 2, \ldots, T$, the learner:

1. selects a subset $S_t \subseteq [d]$ of size at most k,
2. observes $\mathbf{x}_t(S_t)$, i.e., the values of the features of $\mathbf{x}_t$ restricted to the subset S_t,
3. makes a prediction $\hat{y}_t \in [-1, 1]$,
4. observes the true label y_t, and suffers loss $(\hat{y}_t - y_t)^2$.

Define regret of the learner as

$$\text{Regret} := \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 - \min_{\mathbf{w}:\, \|\mathbf{w}\|_0 \le k,\, \|\mathbf{w}\| \le 1} \sum_{t=1}^{T} (\mathbf{w} \cdot \mathbf{x}_t - y_t)^2.$$

In case $\hat{y}_t$ is chosen using randomization, we consider expected regret instead.

Given the NP-hardness of computing the optimal k-sparse linear regressor (Foster et al., 2015; Natarajan, 1995), we also consider a variant of the problem which gives the learner more flexibility than the comparator: the learner is allowed to choose $k' \ge k$ coordinates to query in each round.

1. *Proper* learning means finding the optimal sparse linear regressor, whereas *improper* learning means finding an arbitrary predictor with performance comparable to that of the optimal sparse linear regressor.

The definition of the regret remains the same. We call this the (k, k', d)-*online sparse regression problem*.

We are interested in the following two goals[2]:

1. (No Regret) Make predictions $\hat{y}_t$ so that regret is bounded by $\mathrm{poly}(d, k)T^{1-\delta}$ for some $\delta > 0$.
2. (Efficiency) Make these predictions efficiently, i.e., in $\mathrm{poly}(d, k, T)$ time per iteration.

In this paper, we show it is possible to get an *inefficient* no-regret algorithm for the online sparse regression problem. Complementing this result, we also show that an *efficient* no-regret algorithm cannot exist, assuming the standard hardness assumption that $\mathsf{NP} \not\subseteq \mathsf{BPP}$.

4. Upper bound

In this section we give an *inefficient* algorithm for the (k, k', d)-online sparse regression problem which obtains an expected regret of $O(\frac{d^2}{(k'-k)^2}\sqrt{k\log(d)T})$. The algorithm needs k' to be at least $k + 2$. It is inefficient because it maintains statistics for every subset of $[d]$ of size k, of which there are $\binom{d}{k}$.

At a high level, the algorithm runs an experts algorithm (specifically, Hedge) treating all such subsets as experts. Each expert internally runs stochastic gradient descent only on the coordinates specified by the corresponding subset, ensuring low regret to any bounded-norm parameter vector that is nonzero only on those coordinates. The Hedge algorithm ensures low regret to the best subset of coordinates, and thus the overall algorithm achieves low regret with respect to any k-sparse parameter vector. The necessity of using $k' \geq k + 2$ features in the algorithm is that the algorithm uses the additional $k' - k$ features to generate unbiased estimators for $\mathbf{x}_t\mathbf{x}_t^\top$ and $y_t\mathbf{x}_t$ in each round, which are needed to generate stochastic gradients for all the experts. These estimators have large variance unless $k' - k$ is large.

The pseudocode is given in Algorithm 1. In the algorithm, in round t, the algorithm generates a distribution D_t over the subsets of $[d]$ of size k; for any such subset S, we use the notation $D_t(S)$ to denote the probability of choosing the set S in this distribution. We also define the function Π on $\mathbb{R}^d$ to be the projection onto the unit ball, i.e., for $\mathbf{w} \in \mathbb{R}^d$, $\Pi(\mathbf{w}) = \mathbf{w}$ if $\|\mathbf{w}\| \leq 1$, and $\Pi(\mathbf{w}) = \frac{1}{\|\mathbf{w}\|}\mathbf{w}$ otherwise.

Theorem 1 *There is an algorithm for the online sparse regression problem with any given parameters (k, k', d) such that $k' \geq k + 2$ running in $O(\binom{d}{k} \cdot k')$ time per iteration with $O(\frac{d^2}{(k'-k)^2}\sqrt{k\log(d)T})$ expected regret.*

Proof The algorithm is given in Algorithm 1. Since the algorithm maintains a parameter vector in $\mathbb{R}^k$ for each subset of $[d]$ of size k, the running time is dominated by the time to sample from D_t and update it, and the time to update the parameter vectors. The updates can be implemented in $O(k')$ time, so overall each round can be implemented in $O(\binom{d}{k} \cdot k')$ time.

We now analyze the regret of the algorithm. Let $\mathbb{E}_t[\cdot]$ denote the expectation conditioned on all the randomness prior to round t. Then, it is easy to check, using the fact that $k' - k \geq 2$, that the construction of $\mathbf{X}_t$ and $\mathbf{z}_t$ in Step 8 of the algorithm has the following property:

$$\mathbb{E}_t[\mathbf{X}_t] = \mathbf{x}_t\mathbf{x}_t^\top \text{ and } \mathbb{E}_t[\mathbf{z}_t] = y_t\mathbf{x}_t. \tag{1}$$

2. In this paper, we use the $\mathrm{poly}(\cdot)$ notation to denote a polynomially-bounded function of its arguments.

Algorithm 1 ALGORITHM FOR ONLINE SPARSE REGRESSION

1: Define the parameters $p = \frac{k'-k}{d}$, $q = \frac{(k'-k)(k'-k-1)}{d(d-1)}$, $\eta_{\text{HEDGE}} = q\sqrt{\frac{\ln(d)}{T}}$, and $\eta_{\text{SGD}} = q\sqrt{\frac{1}{T}}$.
2: Let D_1 be the uniform distribution over all subsets of $[d]$ of size k.
3: For every subset S of $[d]$ of size k, let $\mathbf{w}_{S,1} = \mathbf{0}$, the all-zero vector in $\mathbb{R}^d$.
4: **for** $t = 1, 2, \ldots, T$ **do**
5: Sample a subset $\hat{S}_t$ of $[d]$ of size k from D_t, and a subset R_t of $[d]$ of size $k' - k$ drawn uniformly at random, independently of $\hat{S}_t$.
6: Acquire $\mathbf{x}_t(S_t)$ for $S_t := \hat{S}_t \cup R_t$.
7: Make the prediction $\hat{y}_t = \mathbf{w}_{\hat{S}_t,t} \cdot \mathbf{x}_t$ and obtain the true label y_t.
8: Compute the matrix $\mathbf{X}_t \in \mathbb{R}^{d\times d}$ and the vector $\mathbf{z}_t \in \mathbb{R}^d$ defined as follows:

$$\mathbf{X}_t(i,j) = \begin{cases} \frac{\mathbf{x}_t(i)^2}{p} & \text{if } i = j \text{ and } i \in R_t \\ \frac{\mathbf{x}_t(i)\mathbf{x}_t(j)}{q} & \text{if } i \neq j \text{ and } i,j \in R_t \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \mathbf{z}_t(i) = \begin{cases} \frac{y_t\mathbf{x}_t(i)}{p} & \text{if } i \in R_t \\ 0 & \text{otherwise,} \end{cases}$$

9: Update the distribution over the subsets: for all subsets S of $[d]$ of size k, let

$$D_{t+1}(S) \;=\; D_t(S)\exp(-\eta_{\text{HEDGE}}(\mathbf{w}_{S,t}^\top \mathbf{X}_t \mathbf{w}_{S,t} - 2\mathbf{z}_t^\top \mathbf{w}_{S,t} + y_t^2))/Z_t,$$

where Z_t is the normalization factor to make D_{t+1} a distribution.
10: For each subset S of $[d]$ of size k, let

$$\mathbf{w}_{S,t+1} \;=\; \Pi(\mathbf{w}_{S,t} - 2\eta_{\text{SGD}}\mathbf{I}_S(\mathbf{X}_t\mathbf{w}_{S,t} - \mathbf{z}_t)).$$

11: **end for**

Next, notice that in Step 9, the algorithm runs the standard Hedge-algorithm update (see, for example, Section 2.1 in (Arora et al., 2012)) on $\binom{d}{k}$ experts, one for each subset of $[d]$ of size k, where, in round t, the cost of the expert corresponding to subset S is defined to be[3]

$$\mathbf{w}_{S,t}^\top \mathbf{X}_t \mathbf{w}_{S,t} - 2\mathbf{z}_t^\top \mathbf{w}_{S,t} + y_t^2. \tag{2}$$

It is easy to check, using the facts that $\|\mathbf{x}_t\| \le 1$, $\|\mathbf{w}_{S,t}\| \le 1$ and $p \ge q$, that the cost (2) is bounded deterministically in absolute value by $O(\frac{1}{q}) = O(\frac{d^2}{(k'-k)^2})$. Let $\mathbb{E}_{D_t}[\cdot]$ denote the expectation over the random choice of $\hat{S}_t$ from the distribution D_t conditioned on all other randomness up to and including round t. Since there are $\binom{d}{k} \le d^k$ experts in the Hedge algorithm here, the standard regret bound for Hedge (Arora et al., 2012, Theorem 2.3) with the specified value of η_{HEDGE} implies that for any subset S of $[d]$ of size k, using $\ln\binom{d}{k} \le k\ln d$, we have

$$\sum_{t=1}^T \mathbb{E}_{D_t}[\mathbf{w}_{\hat{S}_t,t}\mathbf{X}_t\mathbf{w}_{\hat{S}_t,t} - 2\mathbf{z}_t^\top\mathbf{w}_{\hat{S}_t,t} + y_t^2] \;\le\; \sum_{t=1}^T(\mathbf{w}_{S,t}^\top\mathbf{X}_t\mathbf{w}_{S,t} - 2\mathbf{z}_t^\top\mathbf{w}_{S,t} + y_t^2) + O(\tfrac{d^2}{(k'-k)^2}\sqrt{k\ln(d)T}). \tag{3}$$

3. Recall that the costs in Hedge may be chosen adaptively.

Next, we note, using (1) and the fact that conditioned on the randomness prior to round t, $\mathbf{w}_{S,t}$ is completely determined, that (for any S)

$$\mathbb{E}_t[\mathbf{w}_{S,t}^\top \mathbf{X}_t \mathbf{w}_{S,t} - 2\mathbf{z}_t^\top \mathbf{w}_{S,t} + y_t^2] \;=\; \mathbf{w}_{S,t}^\top \mathbf{x}_t \mathbf{x}_t^\top \mathbf{w}_{S,t} - 2y_t \mathbf{x}_t^\top \mathbf{w}_{S,t} + y_t^2 \;=\; (\mathbf{w}_{S,t} \cdot \mathbf{x}_t - y_t)^2. \quad (4)$$

Taking expectations on both sides of (3) over all the randomness in the algorithm, and using (4), we get that for any subset S of $[d]$ of size k, we have

$$\sum_{t=1}^{T} \mathbb{E}[(\mathbf{w}_{\hat{S}_t,t} \cdot \mathbf{x}_t - y_t)^2] \;\leq\; \sum_{t=1}^{T} \mathbb{E}[(\mathbf{w}_{S,t} \cdot \mathbf{x}_t - y_t)^2] + O(\tfrac{d^2}{(k'-k)^2}\sqrt{k \log(d) T}). \quad (5)$$

The left-hand side of (5) equals $\sum_{t=1}^{T} \mathbb{E}[(\hat{y}_t - y_t)^2]$. We now analyze the right-hand side.

For any given subset S of $[d]$ of size k, we claim that in Step 10 of the algorithm, the parameter vector $\mathbf{w}_{S,t}$ is updated using stochastic gradient descent with the loss function $\ell_t(\mathbf{w}) := (\mathbf{x}_t^\top \mathbf{I}_S \mathbf{w} - y_t)^2$ over the set $\{\mathbf{w} \mid ||\mathbf{w}||_2 \leq 1\}$, only on the coordinates in S, while the coordinates not in S are fixed to 0. To prove this claim, first, we note that the premultiplication by $\mathbf{I}_S$ in the update in Step 10 ensures that in the parameter vector $\mathbf{w}_{S,t+1}$, all coordinates that are not in S are set to 0, assuming that coordinates of $\mathbf{w}_{S,t}$ not in S were 0.

Next, at time t, consider the loss function $\ell_t(\mathbf{w}) = (\mathbf{x}_t^\top \mathbf{I}_S \mathbf{w} - y_t)^2$. The gradient of this loss function at $\mathbf{w}_{S,t}$ is

$$\nabla \ell_t(\mathbf{w}_{S,t}) \;=\; 2(\mathbf{x}_t^\top \mathbf{I}_S \mathbf{w}_{S,t} - y_t)\mathbf{I}_S \mathbf{x}_t \;=\; 2\mathbf{I}_S(\mathbf{x}_t \mathbf{x}_t^\top \mathbf{w}_{S,t} - y_t \mathbf{x}_t),$$

where we use the fact that $\mathbf{I}_S \mathbf{w}_{S,t} = \mathbf{w}_{S,t}$ since $\mathbf{w}_{S,t}$ has zeros in coordinates not in S. Now, by (1), we have

$$\mathbb{E}_t[2\mathbf{I}_S(\mathbf{X}_t \mathbf{w}_{S,t} - \mathbf{z}_t)] \;=\; 2\mathbf{I}_S(\mathbf{x}_t \mathbf{x}_t^\top \mathbf{w}_{S,t} - y_t \mathbf{x}_t),$$

and thus, Step 10 of the algorithm is a stochastic gradient descent update as claimed. Furthermore, a calculation similar to the one for bounding the loss of the experts in the Hedge algorithm shows that the norm of the stochastic gradient is bounded deterministically by $O(\frac{1}{q})$, which is $O(\frac{d^2}{(k'-k)^2})$.

Using a standard regret bound for stochastic gradient descent (see, for example, Lemma 3.1 in (Flaxman et al., 2005)) with the specified value of η_{SGD}, we conclude that for any fixed vector $\mathbf{w}$ of ℓ_2 norm at most 1, we have,

$$\sum_{t=1}^{T} \mathbb{E}[(\mathbf{x}_t^\top \mathbf{I}_S \mathbf{w}_{S,t} - y_t)^2] \;\leq\; \sum_{t=1}^{T} (\mathbf{x}_t^\top \mathbf{I}_S \mathbf{w} - y_t)^2 + O(\tfrac{d^2}{(k'-k)^2}\sqrt{T}).$$

Since $\mathbf{I}_S \mathbf{w}_{S,t} = \mathbf{w}_{S,t}$, the left hand side of the above inequality equals $\sum_{t=1}^{T} \mathbb{E}[(\mathbf{w}_{S,t} \cdot \mathbf{x}_t - y_t)^2]$.

Finally, let $\mathbf{w}$ be an arbitrary k-sparse vector of ℓ_2 norm at most 1. Let $S = \{i \mid w_i \neq 0\}$. Note that $|S| \leq k$, and $\mathbf{I}_S(\mathbf{w}) = \mathbf{w}$. Applying the above bound for this set S, we get

$$\sum_{t=1}^{T} \mathbb{E}[(\mathbf{w}_{S,t} \cdot \mathbf{x}_t - y_t)^2] \;\leq\; \sum_{t=1}^{T} (\mathbf{w} \cdot \mathbf{x}_t - y_t)^2 + O(\tfrac{d^2}{(k'-k)^2}\sqrt{T}). \quad (6)$$

Combining the inequality (6) with inequality (5), we conclude that

$$\sum_{t=1}^{T} \mathbb{E}[(\hat{y}_t - y_t)^2] \;\leq\; \sum_{t=1}^{T} (\mathbf{w} \cdot \mathbf{x}_t - y_t)^2 + O(\tfrac{d^2}{(k'-k)^2}\sqrt{k \log(d) T}).$$

This gives us the required regret bound. ■

5. Computational lower bound

In this section we show that there cannot exist an efficient no-regret algorithm for the online sparse regression problem unless NP $\subseteq$ BPP. This hardness result follows from the hardness of approximating the Set Cover problem. We give a reduction showing that if there were an efficient no-regret algorithm $\mathsf{Alg}_{\mathsf{OSR}}$ for the online sparse regression problem, then given an instance of the Set Cover problem, we could distinguish, in randomized polynomial time, between two cases: either there is a small set cover for the instance, or any set cover of the instance is large. This task is known to be NP-hard for specific parameter values. Specifically, our reduction has the following properties:

1. If the instance has a small set cover, then in the induced online sparse regression problem there is a k-sparse parameter vector (of ℓ_2 norm at most 1) giving 0 loss, and thus the algorithm $\mathsf{Alg}_{\mathsf{OSR}}$ must have small total loss (equal to the regret) as well.

2. If there is no small set cover for the instance, then the prediction made by $\mathsf{Alg}_{\mathsf{OSR}}$ in any round has at least a constant loss in expectation, which implies that its total (expected) loss must be large, in fact, linear in T.

By measuring the total loss of the algorithm, we can distinguish between the two instances of the Set Cover problem mentioned above with probability at least $3/4$, thus yielding a BPP algorithm for an NP-hard problem.

The starting point for our reduction is the work of Dinur and Steurer (2014), who give a polynomial-time reduction of deciding satisfiability of 3CNF formulas to distinguishing instances of Set Cover with certain useful combinatorial properties. We denote the satisfiability problem of 3CNF formulas by 3SAT.

Reduction 1 *For any given constant $D > 0$, there is a constant $c_D \in (0,1)$ and a $poly(n^D)$-time algorithm that takes a 3CNF formula ϕ of size n as input and constructs a Set Cover instance over a ground set of size $m = \mathrm{poly}(n^D)$ with $d = \mathrm{poly}(n)$ sets, with the following properties:*

1. *if $\phi \in$ 3SAT, then there is a collection of $k = O(d^{c_D})$ sets, which covers each element* exactly once*, and*

2. *if $\phi \notin$ 3SAT, then no collection of $k' = \lfloor D \ln(d) k \rfloor$ sets covers all elements, i.e., at least one element is left uncovered.*

The Set Cover instance generated from ϕ can be encoded as a binary matrix $\mathbf{M}_\phi \in \{0,1\}^{m \times d}$ with the rows corresponding to the elements of the ground set, and the columns correspond to the sets, such that each column is the indicator vector of the corresponding set.

Using this reduction, we now show how an efficient, no-regret algorithm for online sparse regression can be used to give a BPP algorithm for 3SAT.

Theorem 2 *Let $D > 0$ be any given constant. Suppose there is an algorithm, $\mathsf{Alg}_{\mathsf{OSR}}$, for the (k, k', d)-online sparse regression problem with $k = O(d^{c_D})$, where c_D is the constant from Reduction 1, and $k' = \lfloor D \ln(d) k \rfloor$, that runs in $\mathrm{poly}(d, T)$ time per iteration and has expected regret bounded by $\mathrm{poly}(d) T^{1-\delta}$ for some constant $\delta > 0$. Then NP $\subseteq$ BPP.*

Algorithm 2 ALGORITHM $\mathsf{Alg}_{\mathsf{3SAT}}$ FOR DECIDING SATISFIABILITY OF 3CNF FORMULAS

Require: A constant $D > 0$, and an algorithm $\mathsf{Alg}_{\mathsf{OSR}}$ for the (k, k', d)-online sparse regression problem with $k = O(d^{c_D})$, where c_D is the constant from Reduction 1, and $k' = \lfloor D \ln(d) k \rfloor$, that runs in $\mathrm{poly}(d, T)$ time per iteration with regret bounded by $p(d) \cdot T^{1-\delta}$ with probability at least $3/4$.

Require: Online algorithm $\mathsf{Alg}_{\mathsf{OSR}}$ and a 3CNF formula ϕ.

Compute the matrix $\mathbf{M}_\phi$ and the associated parameters k, k', d, m from Reduction 1.
Start running $\mathsf{Alg}_{\mathsf{OSR}}$ with the parameters k, k', d.
for $t = 1, 2, \ldots, T := \lceil \max\{(2p(d)mdk)^{1/\delta}, 256m^2d^2k^2\} \rceil$ **do**
    Sample a row of $\mathbf{M}_\phi$ uniformly at random; call it $\hat{\mathbf{x}}_t$.
    Sample $\sigma_t \in \{-1, 1\}$ uniformly at random independently of $\hat{\mathbf{x}}_t$.
    Set $\mathbf{x}_t = \frac{\sigma_t}{\sqrt{d}} \hat{\mathbf{x}}_t$ and $y_t = \frac{\sigma_t}{\sqrt{dk}}$.
    Obtain a set of coordinates S_t of size at most k' by invoking $\mathsf{Alg}_{\mathsf{OSR}}$, and provide it the coordinates $\mathbf{x}_t(S_t)$.
    Obtain the prediction $\hat{y}_t$ from $\mathsf{Alg}_{\mathsf{OSR}}$, and provide it the true label y_t.
end for
if $\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \leq \frac{T}{2mdk}$ **then**
    Output "satisfiable".
else
    Output "unsatisfiable".
end if

Proof Since the expected regret of $\mathsf{Alg}_{\mathsf{OSR}}$ is bounded by $p(d)T^{1-\delta}$ (where $p(d)$ is a polynomial function of d), by Markov's inequality we conclude that with probability at least $3/4$, the regret of $\mathsf{Alg}_{\mathsf{OSR}}$ is bounded by $p(d) \cdot T^{1-\delta}$. Figure 2 gives a randomized algorithm, $\mathsf{Alg}_{\mathsf{3SAT}}$, for deciding satisfiability of a given 3CNF formula ϕ using the algorithm $\mathsf{Alg}_{\mathsf{OSR}}$. Note that the feature vectors (i.e., the $\mathbf{x}_t$ vectors) generated by $\mathsf{Alg}_{\mathsf{3SAT}}$ are bounded in ℓ_2 norm by 1, as required. It is clear that $\mathsf{Alg}_{\mathsf{3SAT}}$ is a polynomial-time algorithm since T is a polynomial function of n (since $m, k, d, p(d)$ are polynomial functions of n), and $\mathsf{Alg}_{\mathsf{OSR}}$ runs in $\mathrm{poly}(d, T)$ time per iteration.

We now claim that this algorithm correctly decides satisfiability of ϕ with probability at least $3/4$, and is hence a BPP algorithm for 3SAT.

To prove this, suppose $\phi \in$ 3SAT. Then, there are k sets in the Set Cover which cover all elements with each element being covered exactly once. Consider the k-sparse parameter vector $\mathbf{w}$ which has $\frac{1}{\sqrt{k}}$ in the positions corresponding to these k sets and 0 elsewhere. Note that $\|\mathbf{w}\| \leq 1$, as required. Note that since this collection of k sets covers each element exactly once, we have $\mathbf{M}_\phi \mathbf{w} = \frac{1}{\sqrt{k}} \mathbf{1}$, where $\mathbf{1}$ is the all-1's vector. In particular, since $\hat{\mathbf{x}}_t$ is a row of $\mathbf{M}_\phi$, we have

$$\mathbf{w} \cdot \mathbf{x}_t \;=\; \mathbf{w} \cdot \left(\frac{\sigma_t}{\sqrt{d}} \hat{\mathbf{x}}_t \right) \;=\; \frac{\sigma_t}{\sqrt{dk}} \;=\; y_t.$$

Thus, $(\mathbf{w} \cdot \mathbf{x}_t - y_t)^2 = 0$ for all rounds t. Since algorithm $\mathsf{Alg}_{\mathsf{OSR}}$ has regret at most $p(d) \cdot T^{1-\delta}$ with probability at least $3/4$, its total loss $\sum_{t=1}^{T} (\hat{y}_t - y_t)^2$ is bounded by $p(d) \cdot T^{1-\delta} \leq \frac{T}{2mdk}$ (since $T \geq (2p(d)mdk)^{1/\delta}$), with probability at least $3/4$. Thus, in this case algorithm $\mathsf{Alg}_{\mathsf{3SAT}}$ correctly outputs "satisfiable" with probability at least $3/4$.

Next, suppose $\phi \notin$ 3SAT. Fix any round t and let S_t be the set of coordinates of size at most k' selected by algorithm $\mathsf{Alg}_{\mathsf{OSR}}$ to query. This set S_t corresponds to k' sets in the Set Cover instance. Since $\phi \notin$ 3SAT, there is at least one element in the ground set that is not covered by any set among these k' sets. This implies that there is at least one row of $\mathbf{M}_\phi$ that is 0 in all the coordinates in S_t. Since $\hat{\mathbf{x}}_t$ is a uniformly random row of $\mathbf{M}_\phi$ chosen independently of S_t, we have

$$\Pr[\mathbf{x}_t(S_t) = \mathbf{0}] \;=\; \Pr[\hat{\mathbf{x}}_t(S_t) = \mathbf{0}] \;\geq\; \frac{1}{m}.$$

Here, $\mathbf{0}$ denotes the all-zero vector of size k'.

Now, we claim that $\mathbb{E}[y_t \hat{y}_t \mid \mathbf{x}_t(S_t) = \mathbf{0}] = 0$. This is because the condition that $\mathbf{x}_t(S_t) = 0$ is equivalent to the condition that $\hat{\mathbf{x}}_t(S_t) = 0$. Since y_t is chosen from $\{-\frac{1}{\sqrt{dk}}, \frac{1}{\sqrt{dk}}\}$ uniformly at random independently of $\hat{\mathbf{x}}_t$ and $\hat{y}_t$, the claim follows. The expected loss of the online algorithm in round t can now be bounded as follows:

$$\begin{aligned}\mathbb{E}[(\hat{y}_t - y_t)^2 \;\big|\; \mathbf{x}_t(S_t) = \mathbf{0}] \;&=\; \mathbb{E}\left[\hat{y}_t^2 + \frac{1}{dk} - 2y_t\hat{y}_t \;\middle|\; \mathbf{x}_t(S_t) = \mathbf{0}\right] \\ &=\; \mathbb{E}\left[\hat{y}_t^2 + \frac{1}{dk} \;\middle|\; \mathbf{x}_t(S_t) = \mathbf{0}\right] \;\geq\; \frac{1}{dk},\end{aligned}$$

and hence

$$\mathbb{E}[(y_t - \hat{y}_t)^2] \;\geq\; \mathbb{E}[(y_t - \hat{y}_t)^2 \mid \mathbf{x}_t(S_t) = \mathbf{0}] \cdot \Pr[\mathbf{x}_t(S_t) = \mathbf{0}] \;\geq\; \frac{1}{dk} \cdot \frac{1}{m} \;=\; \frac{1}{mdk}.$$

Let $\mathbb{E}_t[\cdot]$ denote expectation of a random variable conditioned on all randomness prior to round t. Since the choices of $\mathbf{x}_t$ and y_t are independent of previous choices in each round, the same argument also implies that $\mathbb{E}_t[(y_t - \hat{y}_t)^2] \;\geq\; \frac{1}{mdk}$. Applying Azuma's inequality (see Theorem 7.2.1 in (Alon and Spencer, 1992)) to the martingale difference sequence $\mathbb{E}_t[(y_t - \hat{y}_t)^2] - (y_t - \hat{y}_t)^2$ for $t = 1, 2, \ldots, T$, since each term is bounded in absolute value by 4, we get

$$\Pr\left[\sum_{t=1}^{T} \mathbb{E}_t[(y_t - \hat{y}_t)^2] - (y_t - \hat{y}_t)^2 \geq 8\sqrt{T}\right] \;\leq\; \exp\left(-\tfrac{64T}{2 \cdot 16T}\right) \;\leq\; \frac{1}{4}.$$

Thus, with probability at least $3/4$, the total loss $\sum_{t=1}^{T}(\hat{y}_t - y_t)^2$ is greater than $\sum_{t=1}^{T} \mathbb{E}_t[(y_t - \hat{y}_t)^2] - 8\sqrt{T} \;\geq\; \frac{1}{mdk}T - 8\sqrt{T} \;\geq\; \frac{T}{2mdk}$ (since $T \geq 256m^2d^2k^2$). Thus in this case the algorithm correctly outputs "unsatisfiable" with probability at least $3/4$. ■

5.1. Parameter settings for hard instances

Theorem 2 implies that for any given constant $D > 0$, there is a constant c_D such that the parameter settings $k = O(d^{c_D})$, and $k' = \lfloor D \ln(d) k \rfloor$ yield hard instances for the online sparse regression problem. The reduction of Dinur and Steurer (2014) can be "tweaked"[4] so that the c_D is arbitrarily close to 1 for any constant D.

4. This is accomplished by simply replacing the Label Cover instance they construct with polynomially many disjoint copies of the same instance.

We can now extend the hardness results to the parameter settings $k = O(d^\epsilon)$ for any $\epsilon \in (0, 1)$ and $k' = \lfloor D \ln(d) k \rfloor$ either by tweaking the reduction of Dinur and Steurer (2014) so it yields $c_D = \epsilon$ if ϵ is close enough to 1, or if ϵ is small, by adding $O(d^{1/\epsilon})$ all-zero columns to the matrix $\mathbf{M}_\phi$. The two combinatorial properties of $\mathbf{M}_\phi$ in Reduction 1 are clearly still satisfied, and the proof of Theorem 2 goes through.

5.2. Allowing flexibility in choosing features to observe

While the hardness result described above applies in the setting where the online learner is restricted to choose k' features to observe in each round, our hardness proof easily extends to more flexible situations such as that in which the online learner is only required to keep the average number of features observed over the rounds bounded by k'.

To see this, suppose that k' is such that it is NP-hard to distinguish between cases where either the instance has a set cover of size k or no collection of $2k'$ sets covers all the elements. Then consider an algorithm for the online sparse regression problem which chooses at most k' features on average over the rounds. By Markov's inequality, on at least half the rounds, the algorithm chooses at most $2k'$ features, and in these rounds, the arguments in the proof of Theorem 2 imply that the algorithm incurs at least constant loss in expectation, leading to linear (in T) total loss in expectation. This suffices to show hardness, since any sublinear regret algorithm for the problem can then be used to distinguish between the two cases, exactly as in the proof of Theorem 2.

6. Conclusions

In this paper, we prove that minimizing regret in the online sparse regression problem is computationally hard even if the learner is allowed access to many more features than the comparator, a sparse linear regressor. We complement this result by giving an inefficient no-regret algorithm.

The main open question remaining from this work is what extra assumptions can one make on the examples arriving online to make the problem tractable. Note that the sequence of examples constructed in the lower bound proof is i.i.d., so clearly stronger assumptions than that are necessary to obtain any efficient algorithms.

References

Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, 1992. ISBN 0-471-53588-5.

Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. In *COLT*, pages 287–296, 1993.

Andreas Birkendorf, Eli Dichterman, Jeffrey C. Jackson, Norbert Klasner, and Hans-Ulrich Simon. On restricted-focus-of-attention learnability of boolean functions. *Machine Learning*, 30(1):89–123, 1998.

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

Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014.

Abraham Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA*, pages 385–394, 2005.

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

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

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

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

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

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Computing*, 25(2):227–234, 1995.

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