Computational hardness of fast rates for online sparse PCA: improperness does not help

Elad Hazan* Andrej Risteski†

December 1, 2018

Abstract

One of the most frequent and successful techniques to deal with computational intractability in statistics and machine learning is improper learning by convex relaxation: namely enlarging the class of predictors one is interested in, in order to make the optimization tractable. The common caveat with this approach is that the convex relaxation enlarges the decision set to the extent of slowing the rate (regret) the algorithm significantly.

In this work we pose the question: are there natural online learning problems for which suboptimal regret is achievable using an efficient algorithm, and yet achieving the information-theoretically optimal regret is hard, even if we allow the learner to be improper? We answer this in the affirmative by exhibiting such a problem – an online variant of the sparse PCA problem from statistics.

For this problem, we show that an inefficient algorithm based on the multipicative updates method achieves a round-independent regret, yet achieving regret better than $T^{1/4}$ in polynomial time would violate the planted clique conjecture, even if we allow the learner to be improper.

1 Introduction

Improper learning, especially through convex relaxations is one of the success stories in both statistics and online learning. The idea is that when a family of hypotheses for a problem is difficult to optimize over, one may be able to instead "enlarge" that family, at no cost in terms of sample complexity, yet at the significant benefit of getting an efficient algorithm.

In the statistical community, the most famous examples of this are settings involving sparse and low-rank predictors, which admit natural convex relaxations such as ℓ_1-norm and nuclear norm relaxations. This is effectively the basis of such rich areas of research as *sparse linear regression*, *sparse PCA* and *trace regression*, and in signal processing methods like *matrix completion*, *compressed sensing*, *sparse coding*, etc. In almost all of these cases, convex relaxations are well-understood: typically, with additional assumptions on the problem (e.g. incoherence), the convex relaxation matches the information-theoretic bound on the number of samples necessary. Without such assumptions, however, the statistical performance can significantly deteriorate. For example, LASSO has a poor rate for sparse linear regression, when the dictionary matrix is not incoherent (Zhang et al., 2017).

In the online learning community, examples of successful usage of improper learning include problems like *online max cut, online gambling, online matrix completion* (Hazan et al., 2012; Christiano, 2014; Rakhlin and Sridharan, 2015; Awasthi et al., 2015), *dictionary learning and spectral autoencoders* (Hazan and Ma, 2016), *neural networks* (Livni et al., 2014; Goel and Klivans, 2017; Zhang et al., 2016), and recently control and reinforcement learning (Hazan et al., 2017; 2018). In many of these instances, the obvious regret-optimal algorithm consists of running the *continuous multiplicative weights / experts* algorithm, with an expert for each predictor. The set of predictors, however, is exponentially large, so this results in a non-efficient algorithm. Improper learning remedies this by instead running follow-the-regularized-leader, (usually) over a convex hull of the original predictors with an appropriate regularizer.

*Princeton University. Computer Science Department.

†Massachusets Institute of Technology. Applied Mathematics Department and IDSS.

While the above instances are encouraging, there is no overarching theory for *when* we can hope go use an improper approach to get an efficient algorithm, while not sacrificing the regret. Thus, we ask the following natural question:

Is there a simple, natural setup, in which a computationally inefficient algorithm achieves a much lower regret than any computationally efficient algorithm, **even** *allowing it to be improper?*

We answer this question in the affirmative, in a very natural setup: *online sparse PCA*. In this setup, at any round $t = 1, 2, .., T$, the learner predicts with a matrix X^t, and receives a cost of $(X^t_{i,j} - Y^t_{i,j})^2$, for indices (i, j) and matrix Y^t chosen by the adversary.

We measure the regret with respect to the best matrix X which is of the form $uu^\top$, for a vector u which is k-sparse (in other words, rank-1 and sparse.) – while **allowing** the algorithm to output a matrix X in the convex hull of these predictors. This problem is an online version of sparse PCA (Johnstone, 2001), which is typically studied in the batch, statistical seting.

We show that there is a natural computationally inefficient algorithm which achieves a dimension independent regret $O(k \log d)$. In contrast, no **efficient** algorithm can achieve regret better than $O(kT^{1/4})$, assuming that the *planted clique problem* has no polynomial-time algorithm – even if we allow the algorithm to be improper.

2 Overview of results

The problem setup we focus on is *online sparse PCA*, which can be described as the following online game:

Online sparse PCA: At each round t, the learner is asked to provide a matrix X^t, s.t. $\sum_{i,j=1}^d |X^t_{i,j}| \leq k^2$ and the coordinates of X^t satisfy $-1 \leq X^t_{i,j} \leq 1, \forall (i, j) \in [d] \times [d]$.

At any round t, the loss that the algorithm suffers is $f^t(x) = (X^t_{i^t,j^t} - Y^t_{i^t,j^t})^2$ for a matrix Y, s.t. $-1 \leq Y_{i,j} \leq 1, \forall (i, j) \in [d] \times [d]$ and indices (i, j) chosen by the adversary. [1]

The regret will be measured with respect to the optimal sparse, rank-1 predictor

$$\text{Regret} = \sum_{t=1}^{T} f^t(X^t) - \min_{X^*: X^* = uu^\top, u \in \{-1,0,1\}^d, \|u\|_0 = k} \sum_{t=1}^{T} f^t(X^*)$$

Note that this indeed fits the paradigm of the question we asked in the introduction: our algorithm is **improper**, in the sense that we allow the learner to play (a *very* loose) convex combination of the predictors in the set $\{X^* : X^* = uu^\top,\ u \in \{-1, 0, 1\}^d, \|u\|_0 = k\}$. In fact, the usual semidefinite relaxation for sparse PCA in the statistical setup, introduced in (d'Aspremont et al., 2005), would consist of matrices X, s.t.

$$\{X : X \succeq 0;\ \ -1 \leq X_{i,j} \leq 1, \forall (i, j) \in [d] \times [d];\ \ \text{Tr}(X) = k\}$$

Note the positive semidefiniteness constraint implies that $X_{i,i} + X_{j,j} \geq X_{i,j} + X_{j,i}, \forall i, j$ and $X_{i,i} + X_{j,j} \geq -X_{i,j} - X_{j,i}, \forall i, j$, which implies that $\sum_{i,j=1}^d |X^t_{i,j}| \leq k^2$. Of course, we wish to allow the learner to be **as improper** as possible, so the relaxation being looser only makes the result stronger.

[1] We can allow less "local" cost functions – algorithmic results still hold so long as the costs are bounded.

Algorithm 1 Online sparse PCA

1: Prediction set: $\mathcal{K} = \{X \in \mathbb{R}^{d\times d}, \sum_{i,j=1}^{d} |X_{i,j}| \leq k^2, -1 \leq X_{i,j} \leq 1\}$.
2: **for** $t = 1, \ldots, T$ **do**
3: Learner predicts $X^t \in \mathcal{K}$.
4: Adversary generates loss function $f^t : \mathcal{K} \to \mathbb{R}$ given by

$$f^t(X) = (X^t_{i^t,j^t} - Y^t_{i^t,j^t})^2$$

5: Learner suffers $f^t(X^t)$, gets to see $\nabla f^t(X)$.
6: **end for**

The computational hardness assumption we will base our impossibility of fast rate result on is the so-called *planted clique* conjecture, namely:

Assumption 1. *Any algorithm that can distinguish between a graph G sampled from the Erdos-Renyi random-graph ensemble $\mathcal{G}(d, 1/2)$ and the planted clique ensemble $\mathcal{G}(d, 1/2, k)$ for $k = \omega(d^{1/2-4/5\epsilon})$ for a constant $\epsilon > 0$, and succeeds with probability $> 1/2$, cannot run in time poly(d).*

We note that while it's better to base hardness results on safer complexity conjectures like P $\neq$ NP, it's not at all uncommon for improper learning results to rely on average-case hardness results like planted clique. In fact, in certain setups, it can be shown that improper learning hardness **cannot** be based on P $\neq$ NP or related worst-case hardness assumptions. (Applebaum et al., 2008)

The main result we will prove is the following one:

Theorem 1. .

1. *(*Fast rate with inefficient algorithm*) The experts algorithm , with an expert for each $\{X^* : X^* = uu^\top \in \{-1, 0, 1\}^d, \|u\|_0 = k\}$, achieves (round-independent) regret $O(k \log d)$.*
2. *(*Fast rate is impossible efficiently*) Assuming (1), any polynomial-time algorithm for the online sparse PCA game has regret at least $\Omega(kT^{\frac{1-\epsilon}{4}})$.*

3 Prior work

3.1 Sparse PCA

Sparse PCA in the standard, statistical (spiked covariance) setup, originally introduced by (Johnstone, 2001) is well-studied, especially in recent years, and fairly well understood. In the standard formulation, one receives d-dimensional vectors $x_1, x_2, \ldots, x_T$, and wants to distinguish between the following two scenarios:

1. H_0 (null hypothesis): x_i is sampled from $N(0, I)$.
2. H_1 (signal hypothesis): x_i is set to $\sqrt{\lambda} g_i u + \xi$, where g_i is sampled from $N(0, 1)$ and ξ is sampled from $N(0, I)$, and u is a **sparse** vector, s.t. $u \in \{-1, 0, 1\}^d$ and $\|u\|_0 = k$. Note in this case, the covariance matrix of the x_i's is $\lambda uu^\top + I$ – i.e. roughly has a planted sparse "eigenvector".

It is known that there exists an inefficient algorithm that can distinguish between H_0 and H_1 with constant probability (and in fact, approximately recover u), provided that $T \gtrsim \frac{k \log d}{\lambda^2}$ (Berthet and Rigollet, 2013). In contrast, however, the best polynomial-time algorithm that we know of (based on a convex relaxation, see e.g. (Amini and Wainwright, 2008)) can distinguish between H_0 and H_1 with constant probability (and in fact, approximately recover u), only if $T \gtrsim \frac{k^2 \log d}{\lambda^2}$ (Berthet and Rigollet, 2013). In fact, (Berthet and Rigollet, 2013) showed that if we generalize

the noise model slightly, and allow ξ to follow a sub-Gaussian distribution with covariance bounded by $N(0, I)$, one cannot achieve distinguishing probability better than $1/2$ if $T \lesssim \frac{k^{2-\epsilon} \log d}{\lambda^2}$ if the planted clique conjecture is true for cliques of size $k^{1/2} - \epsilon$.

Let us place Theorem 1 in the context of these known results: the fast rate from our main theorem is $O(k \log d)$, which means that the "per round" regret is $O(\frac{k \log d}{T})$. Thus, after $T \gtrsim k \log d$, the per-round regret drops below 1 – which exactly agrees with the "fast rate" based on an inefficient algorithm in the statistical setup.

On the other hand, it's very easy to see that running online gradient descent in our setup would give a regret of order $O(\sqrt{k^2 T})$, which would result in a "per-round" regret of $O(\frac{k}{\sqrt{T}})$. Thus, in order for the "per-round" regret to drop below 1, we'd need $T \gtrsim k^2$ – exactly matching the "slow rate" in the statistical setup. Note that we can't quite prove hardness for regret $O(kT^{1/2-\epsilon})$: it's not clear to us if this is an artifact of our reduction/proof technique, or if in our online/improper setup, there is a way to achieve regret $O(kT^{1/4})$. We leave this for future work.

We briefly mention that using second-order methods like Online Newton Step (Hazan et al., 2007), it's possible to get regret that scales like $O(d^2 \log T)$ – which improves the dependency on T, but scales poorly with dimension. In fact, in our hardness reduction, the number of rounds T should be thought of as d^2 – one per entry in the matrix, so the above regret bound is vacuous for our construction.

We reiterate that crucially, these lower bounds hold even if we allow **improper** learners. As such, it's quite unclear how to port results from the statistical setup to ours directly.

3.2 Improper learning hardness

As far as hardness results for improper learning, there is a line of work on hardness of improper PAC-learning of various hypothesis classes, initiated by (Kearns and Valiant, 1994), continued by (Klivans and Sherstov, 2009; Kharitonov, 1993) and more recently (Daniely et al., 2014; Daniely and Shalev-Shwartz, 2016) in the context of learning DNF's, intersection of half-spaces, finite automata. Note that these problems are substantially harder from a complexity-theory point of view: by the usual online-to-batch conversion, getting a regret of even $o(T)$ in the online version of the problem is hard. Similarly as in our case though, they are also either based on cryptographic hardness, or random instances of hard problems (e.g. Feige's hypothesis in (Daniely et al., 2014; Daniely and Shalev-Shwartz, 2016)). We also mention that improper PAC-learning hardness **cannot** be based on P $\neq$ NP or related worst-case hardness assumptions, as was shown in a seminal paper by (Applebaum et al., 2008).

Setups where a *fast* rate is intractable to achieve, but a *slow* rate can be efficiently achieved seem almost non-existent. In fact, the only related setup we are aware of is a result due to (Awasthi et al., 2015), who consider a particular online learning setup proposed by (Christiano, 2014), which deals with generalizations of the online max-cut problem: namely an online prediction problem, where one receives a pair of nodes in a graph, predicts labels for them, and is compared to the best fixed labeling in retrospect. The authors prove that achieving the information-theoretically optimal regret is computationally hard, via a reduction from planted clique, same as here. While the hardness is based on the same problem, the reduction is entirely different, as is the analysis; furthermore, importantly, the hardness there is to get the optimal dependency in the regret on the *alphabet size* of the labels, whereas in our case it is the optimal dependency on the *number of rounds* T.

4 Part I: Fast rate with an inefficient algorithm

We proceed with the first part of Theorem 1, the fast rate using an inefficient algorithm. The proof is straightforward here, and it relies on the acheivability of round-independent regret of the experts algorithm when the loss is exp-concave.

Theorem 2 (Inefficient fast rate; restated)**.** *The experts algorithm, with an expert for each* $u \in \{-1, 0, 1\}^d, \|u\|_0 = k$*, achieves regret* $O(k \log d)$.

Proof. By Theorem 3.2 in (Cesa-Bianchi and Lugosi, 2006), it suffices to prove that f^t is $O(1)$-exp-concave over the set $\mathcal{K}$. By the definition of exp-concavity, it suffices to show that

$$\nabla^2(f^t) \gtrsim \nabla(f^t)\nabla(f^t)^\top \tag{4.1}$$

Notice that the Hessian of f^t is $\nabla^2(f^t) = 2e_{i,j}e_{i,j}^\top$, and the gradient is $\nabla(f^t) = 2(X_{i_t,j_t} - Y^t_{i_t,j_t})e_{i,j}$, where $e_{i,j}$ is the matrix with 1 at the (i,j)-th entry, and 0 everywhere else. So,

$$\nabla^2(f^t) \succeq \frac{1}{2(X_{i_t,j_t} - Y_{i_t,j_t})^2}\nabla(f^t)\nabla(f^t)^\top$$

and since $-1 \le X_{i_t,j_y}, Y_{i_t,j_t} \le 1$, this proves (4.1), which is what we need.

□

5 Part II: Impossibility of fast rates with efficient algorithms

We proceed to the proof of the lower bound. The main idea is to use the low-regret algorithm as a distinguisher for the planted vs random distribution. More concretely, we will design losses based on the edges of the input graph G, coming from either the planted or random distribution, and if the cumulative loss is sufficiently small, we will proclaim the graph to be coming from the planted distribution.

The intuition will be that in the planted case, there is a low-cumulative loss solution, and if the regret is small enough, the low-regret algorithm will achieve a small total loss. On the other hand, in the random case, due to the randomness in the graph, we will be able to claim that *any* algorithm: efficient or not, must incur a large cost.

Theorem 3 (Inefficient fast rate; restated)**.** *Assuming* (1)*, any polynomial-time algorithm for the online PCA game has regret at least* $\Omega(kT^{\frac{1-\epsilon}{4}})$.

Proof. The losses for the learner will be supplied as follows.

Let $A_{i,j} \in \{-1,1\}^{d\times d}$ be the adjacency matrix of the input graph, using -1 for an absent edge, and $+1$ for a present edge, which recall is sampled either from $\mathcal{G}(d,1/2)$ or $\mathcal{G}(d,1/2,k)$.

We will simulate the learner for T rounds of the online learning game, for $T = d^{2-\epsilon}$ rounds, as follows:

- Let's sort all pairs $(i,j) : i < j$ in lexicographic order, and apply a uniformly random permutation to the pairs; let's denote by (i_t, j_t) the t-th pair after the permutation is applied.
- In each round t, the loss function is selected as

$$\begin{aligned} f^t(x) &= (A_{i^t,j^t} - X^t_{i^t,j^t})^2 \\ &= (1 - A_{i^t,j^t}X^t_{i^t,j^t})^2 \end{aligned}$$

- At the end, denoting the cumulative loss as C, we output "planted", if

$$C \le T - \frac{1}{2}T\frac{\binom{k}{2}}{\binom{d}{2}}$$

  and "random" otherwise.

We claim the following two claims hold:

Lemma 4. *In the case the graph G is sampled from $\mathcal{G}(d,1/2,k)$, there exists a solution $X^* = uu^\top, u \in \{-1,0,1\}^d, \|u\|_0 = k$, s.t. with high probability (over the choice of the pairs supplied to the algorithm), the cost incurred by it is at most*

$$T - T(1-o(1))\frac{\binom{k}{2}}{\binom{d}{2}}$$

Lemma 5. *In the case the graph G was sampled from $\mathcal{G}(d,1/2)$, the total cost incurred by* any *algorithm with probability* $99/100$ *is at least*

$$T - o\left(T\frac{\binom{k}{2}}{\binom{d}{2}}\right)$$

Before proving these, let's see they imply the statement of the theorem. Suppose that the regret is $O(kT^{\frac{1}{4}(1-\epsilon)})$. In case the graph G is planted, from Lemma 4 we'd get that the total cost is at most

$$T - T(1-o(1))\frac{\binom{k}{2}}{\binom{d}{2}} + O(kT^{\frac{1}{4}(1-\epsilon)}) = T - T(1-o(1))\frac{\binom{k}{2}}{\binom{d}{2}} + o\left(T\frac{k^2}{d^2}\right)$$

where the last equality holds since $k = \omega(d^{1/2-4/5\epsilon})$ and $T = d^{2-\epsilon}$. On the other hand, in the random case, by Lemma 5, the total cost is at least $T - o\left(T\frac{\binom{k}{2}}{\binom{d}{2}}\right)$ with high probability. We prove the two lemmas. We proceed to Lemma 4 next:

Proof of Lemma 4. Let $X^* = uu^\top$, where u is the indicator of the planted clique. Let C^t be the random variable denoting the cost at round t, and note that $C^t = 1 - A_{i^t,j^t}X^*_{i^t,j^t}$ hence

$$\sum_{t=1}^{T} C^t = T - N_T$$

where N_T is a random variable denoting the number of edges queried in the planted clique. We will prove that with high probability, $N_T \geq (1-o(1))T\frac{\binom{k}{2}}{\binom{d}{2}}$. Notice that this immediately implies the proof of the lemma.

For an edge e in the planted clique, let Z_e be an indicator variable, which is 1 if the edge e is *not* queried during the T rounds we run the reduction for. Notice that $\mathbb{E}[Z_e] = \frac{\binom{d}{2}-T}{\binom{d}{2}} = 1 - \frac{T}{\binom{d}{2}}$. Moreover, the variables Z_e are negatively associated , hence we can apply Chernoff, i.e.

$$\Pr\left[\sum_e Z_e \leq \binom{k}{2}\left(1-\frac{T}{\binom{d}{2}}\right) + \sqrt{\binom{k}{2}\left(1-\frac{T}{\binom{d}{2}}\right)\log n}\right] \geq 1 - e^{-\log^2 n}$$

This immediately implies that

$$N_T = \binom{k}{2} - \sum_e Z_e \geq \binom{k}{2}\frac{T}{\binom{d}{2}} - \sqrt{\binom{k}{2}\log n} \geq (1-o(1))T\frac{\binom{k}{2}}{\binom{d}{2}}$$

where the last inequality holds since $T = \omega(\frac{d^2}{k}\sqrt{\log n})$,

Hence, with high probability,

$$\sum_{t=1}^{T} C^t \leq T - (1-o(1))T\frac{\binom{k}{2}}{\binom{d}{2}}$$

□

Next, we move to Lemma 5. The proof is similar to the one of Lemma 4, but somewhat more complicated: the gap in the loss between this case and the random case will be due to the fact that the entries $A_{i,j}$ can be intuitively thought of as being chosen *after* the online learner has played a matrix, at the rounds in which the entry is queried for the first time.

Indeed, when the pair (i^t, j^t) is queried, it hasn't been seen before, so the value of A_{i^t,j^t} is a Rademacher variable independent of anything that the algorithm has seen so far, so it has probability $1/2$ of guessing it. This enables us to prove that the objective $(1 - A_{i^t,j^t}X^t_{i^t,j^t})^2$ has expectation at least T, which is certainly much larger than it is in the planted case. However, to prove the result we need, we need to analyze the variance as well. We use a second-moment method and apply Chebyshev's inequality. Here, we will use two crucial facts:
(1) The covariance between the random variables $A_{i^t,j^t}X^t_{i^t,j^t}$ and $A_{i^{t'},j^{t'}}X^t_{i^{t'},j^{t'}}$ is 0: roughly, the reason for this is again, that the random variables A_{i^t,j^t} and $A_{i^{t'},j^{t'}}$ are zero-mean and mutually independent.
(2) The variance of the random variables $A_{i^t,j^t}X^t_{i^t,j^t}$ is bounded by $1/k^2$ since indeed, the edges (i_t, j_t) are randomly chosen (nearly) uniformly at random.
By the law of total covariance, we can conclude the statement we need.

Proof of Lemma 5. For any learning algorithm, let us write the total cumulative loss as a random variable in the entries of the adjacency matrix A, the choice of pairs (i^t, j^t), and any possible randomness of the algorithm used at round t, which we denote by r^t. Furthermore, for notational convenience, let us denote by $r^{<t}$ the vector of random variable $\{r^k : 1 \leq k < t\}$ and by $A^{<t}$ the vector of random variables $\{(i^t, j^t), A_{i^t,j^t} : 1 \leq k < t\}$. Finally, let's denote by Y^t the matrix with entries $Y^t_{i,j} = 2X^t_{i,j} - (X^t_{i,j})^2$.

In the notation we introduced, the cumulative cost of an algorithm can be written as

$$T - \sum_{t=1}^{T} A_{i^t,j^t} Y^t_{i_t,j_t}$$

We will proceed by analyzing the expectation and variance of $\sum_{t=1}^{T} A_{i^t,j^t} Y^t_{i_t,j_t}$. Namely, we will prove that:

$$\mathbb{E}\left[\sum_{t=1}^{T} A_{i^t,j^t} Y^t_{i_t,j_t}\right] = 0 \tag{5.1}$$

$$\text{Var}\left[\sum_{t=1}^{T} A_{i^t,j^t} Y^t_{i_t,j_t}\right] \leq 9(1+o(1))T\frac{\binom{k}{2}}{\binom{d}{2}} \tag{5.2}$$

Before proving these claims, let's see that they imply the statement of the Lemma. Indeed, by Chebyshev's inequality, we have

$$\Pr\left[\sum_{t=1}^{T} C^t \leq T - 30\sqrt{(1+o(1))T\frac{\binom{k}{2}}{\binom{d}{2}}}\right] \leq \frac{1}{100}$$

Since $T = \omega(\frac{d^2}{k^2})$, the claim follows.

For notational convenience, let' denote by $\tilde{C}_t = A_{i^t,j^t} Y^t_{i_t,j_t}$.

Let us proceed to proving (5.1) first. We claim that $\mathbb{E}[\tilde{C}_t] = 0$. Indeed, by the law of total expectation,

$$\mathbb{E}[A_{i^t,j^t} Y^t_{i_t,j_t}] = \mathbb{E}[Y^t_{i_t,j_t}]\mathbb{E}[A_{i^t,j^t}|Y^t_{i_t,j_t}] \tag{5.3}$$

Since the pair (i^t, j^t) was not queried in any previous round, and the values of the matrix A are independent, $A_{i^t,j^t}|Y^t_{i_t,j_t}$ is -1 with probability $1/2$, and 1 with probability $1/2$. By linearity of expectation, (5.1) follows.

We proceed to (5.2). Expanding the variance of the sum, we wish to bound

$$\sum_{t=1}^{T} \text{Var}(\tilde{C}_t) + \sum_{t \neq t'} \text{Cov}(\tilde{C}_t, \tilde{C}_{t'})$$

Let's bound the variances terms first, which are relatively easy. We have

$$\begin{aligned}
\text{Var}(\tilde{C}_t) &= \text{Var}\left(A_{i_t,j_t} Y^t_{i_t,j_t}\right) \\
&\leq \mathbb{E}\left[(A_{i_t,j_t} Y^t_{i_t,j_t})^2\right] \\
&\leq \mathbb{E}[(Y^t_{i_t,j_t})^2] \\
&\leq \mathbb{E}[(2|X^t_{i_t,j_t}| + |X^t_{i_t,j_t}|^2)^2] \\
&\leq \mathbb{E}[(3|X^t_{i_t,j_t}|)^2] \\
&\leq 9\mathbb{E}[|X^t_{i_t,j_t}|]
\end{aligned}$$

Let us bound $\mathbb{E}[|X^t_{i_t,j_t}|]$ by the condition on the predictors X^t. Namely, we have:

$$\begin{aligned}\mathbb{E}[|X^t_{i_t,j_t}|] &= \mathbb{E}_{A^{<t},r^{\le t}}\left[\mathbb{E}_{(i,j):i<j}\left[|X^t_{i_t,j_t}| \;\middle|\; A^{<t}, r^{\le t}\right]\right] \\ &\overset{①}{=} \mathbb{E}_{A^{<t},r^{\le t}}\left[\frac{1}{\binom{d}{2}-t}\sum_{(i,j)\notin A^{<t}}|X^t_{i_t,j_t}| \;\middle|\; A^{<t}, r^{\le t}\right] \\ &\le (1+o(1))\frac{\binom{k}{2}}{\binom{d}{2}} \qquad (5.4)\end{aligned}$$

(5.5)

where: ① follows since conditioned on $A^{<t}, r^{\le t}$, (i_t, j_t) is uniform over the unseen edges (those not in $A^{<t}$), ② follows since $Y^t_{i_t,j_t} \ge 0$, and the remaining inequalities are just algebraic manipulation. Hence, we have

$$\sum_{t=1}^{T}\mathrm{Var}(\tilde{C}_t) \le 9(1+o(1))T\frac{\binom{k}{2}}{\binom{d}{2}} \qquad (5.6)$$

We next consider the covariance terms. We have

$$\mathrm{Cov}(\tilde{C}_t, \tilde{C}_{t'}) = \mathbb{E}[A_{i_t,j_t}A_{i_{t'},j_{t'}}Y^t_{i^t,j^t}Y^{t'}_{i^{t'},j^{t'}}] - \mathbb{E}[A_{i_t,j_t}Y^t_{i^t,j^t}]\mathbb{E}[A_{i_{t'},j_{t'}}Y^{t'}_{i^{t'},j^{t'}}]$$

We claim that $\mathbb{E}[A_{i_t,j_t}A_{i_{t'},j_{t'}}Y^t_{i^t,j^t}Y^{t'}_{i^{t'},j^{t'}}] = 0, \forall t, t'$ and $\mathbb{E}[A_{i_t,j_t}Y^t_{i^t,j^t}] = 0, \forall t$. The latter terms were already in fact analyzed in (5.3), so we proceed to the former.

$$\mathbb{E}[A_{i_t,j_t}A_{i_{t'},j_{t'}}Y^t_{i^t,j^t}Y^{t'}_{i^{t'},j^{t'}}] = \mathbb{E}\left[Y^t_{i^t,j^t}\mathbb{E}\left[A_{i_t,j_t}\mathbb{E}\left[Y^{t'}_{i^{t'},j^{t'}}|A_{i_t,j_t}, Y^t_{i^t,j^t}\mathbb{E}\left[A_{i_{t'},j_{t'}}|Y^{t'}_{i^{t'},j^{t'}}\right]|A_{i_t,j_t}\right]|Y^t_{i^t,j^t}\right]\right]$$

Consider the innermost expectation over $A_{i_{t'},j_{t'}}$: analogously as in the proof of (5.3), since the pair $(i^{t'}, j^{t'})$ was not queried in any previous round, and the values of the matrix A are independent, even after conditioning on $A_{i_t,j_t}, Y^t_{i^t,j^t}, Y^{t'}_{i^{t'},j^{t'}}$, the variable $A_{i_{t'},j_{t'}}$ is 1 with probability $1/2$ and -1 with probability $1/2$, so

$$\mathbb{E}[A_{i_{t'},j_{t'}}|A_{i_{t'},j_{t'}}Y^t_{i^t,j^t}Y^{t'}_{i^{t'},j^{t'}}] = 0$$

Together with (5.6), the claim (5.2) follows, which finishes the lemma.

□

□

6 Conclusion

We proposed a very natural online variant of the sparse PCA problem, which exhibits hardness to achieve optimal regret – *even* if the algorithm is allowed to be improper. Given the plethora of problems in both statistics and machine learning more generally, for which improper relaxations are a natural choice – it's a very interesting question to study to what extent this phenomenon persists.

For instance, one of the most tantalizing problems in statistics is understanding the computational complexity of *improper* sparse linear regression. Namely, the information-theoretically optimal rate (achievable using a non-polynomial-time algorithm) is substantially better than what computationally efficient algorithms (e.g. Lasso) can achieve (for a survey, see Section 2.4 in (Rigollet and Hütter, 2017)). While computational lower bounds have been proven for *proper* predictors (see, e.g. (Zhang et al., 2014)), nothing is known for *improper* predictors. Studying these questions in an *online* setting may be easier, and potentially illuminating and relevant in practice.

References

Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 2454–2458. IEEE, 2008.

Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 211–220. IEEE, 2008.

Pranjal Awasthi, Moses Charikar, Kevin A Lai, and Andrej Risteski. Label optimal regret bounds for online local learning. In *Conference on Learning Theory*, pages 150–166, 2015.

Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse pca. *arXiv preprint arXiv:1304.0828*, 2013.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Yudong Chen and Jiaming Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv preprint arXiv:1402.1267*, 2014.

Paul Christiano. Online local learning via semidefinite programming. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 468–474. ACM, 2014.

Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnfs. In *Conference on Learning Theory*, pages 815–830, 2016.

Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 441–448. ACM, 2014.

Alexandre d'Aspremont, Laurent E Ghaoui, Michael I Jordan, and Gert R Lanckriet. A direct formulation for sparse pca using semidefinite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

Surbhi Goel and Adam Klivans. Eigenvalue decay implies polynomial-time learnability for neural networks. In *Advances in Neural Information Processing Systems*, pages 2189–2199, 2017.

Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational lower bounds for community detection on random graphs. In *Conference on Learning Theory*, pages 899–928, 2015.

Elad Hazan and Tengyu Ma. A non-generative framework and convex relaxations for unsupervised learning. In *Advances in Neural Information Processing Systems*, pages 3306–3314, 2016.

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

Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. In *Conference on Learning Theory*, pages 38–1, 2012.

Elad Hazan, Karan Singh, and Cyril Zhang. Learning linear dynamical systems via spectral filtering. In *Advances in Neural Information Processing Systems*, pages 6702–6712, 2017.

Elad Hazan, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. Spectral filtering for general linear dynamical systems. *arXiv preprint arXiv:1802.03981*, 2018.

Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *Annals of statistics*, pages 295–327, 2001.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.

Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 372–381. ACM, 1993.

Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pages 855–863, 2014.

Alexander Rakhlin and Karthik Sridharan. Hierarchies of relaxations for online prediction problems with evolving constraints. In *Conference on Learning Theory*, pages 1457–1479, 2015.

Phillippe Rigollet and Jan-Christian Hütter. High dimensional statistics. 2017.

Yuchen Zhang, Martin J Wainwright, and Michael I Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In *Conference on Learning Theory*, pages 921–948, 2014.

Yuchen Zhang, Jason D Lee, and Michael I Jordan. l1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning*, pages 993–1001, 2016.

Yuchen Zhang, Martin J Wainwright, Michael I Jordan, et al. Optimal prediction for sparse linear models? lower bounds for coordinate-separable m-estimators. *Electronic Journal of Statistics*, 11(1):752–799, 2017.