
Online Convex Optimization in the Random Order Model

Dan Garber¹ Gal Korceia² Kfir Y. Levy²

Abstract

Online Convex Optimization (OCO) is a powerful framework for sequential prediction, portraying the natural uncertainty inherent in data-streams as though the data were generated by an almost omniscient adversary. However, this view, which is often too pessimistic for real-world data, comes with a price. The complexity of solving many important online tasks in this adversarial framework becomes much worse than that of their offline and even stochastic counterparts. In this work we consider a natural random-order version of the OCO model, in which the adversary can choose the set of loss functions, but does not get to choose the order in which they are supplied to the learner; Instead, they are observed in uniformly random order. Focusing on two important families of online tasks, one in which the cumulative loss function is strongly convex (though individual loss functions may not even be convex), and the other being online k -PCA, we show that under standard well-conditioned-data assumptions, standard online gradient descent (OGD) methods become much more efficient in the random-order model. In particular, for the first group of tasks OGD guarantees poly-logarithmic regret. In the case of online k -PCA, OGD guarantees sublinear regret using only a rank- k SVD on each iteration and memory linear in the size of the solution.

1. Introduction

Online Convex Optimization (OCO) (Hazan, 2016; Shalev-Shwartz, 2012) has emerged in the past two decades as a

¹Department of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel ²Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa, Israel. Correspondence to: Dan Garbar <dangar@technion.ac.il>, Gal Korceia <galko@campus.technion.ac.il>, Kfir Y. Levy <kfirylevy@technion.ac.il>.

powerful and popular paradigm for modeling sequential prediction problems in face of uncertainty. Its main strength is that the notion of uncertainty considered is very strong and involves worst-case scenarios, i.e., the sequence of loss functions is considered to be selected by an *adversary* who knows the prediction algorithm (perhaps up to potential randomness used for prediction). At the same time, this paradigm lands itself to tractable optimization, and theoretically efficient algorithms that provably minimize the regret are known to exist.

Of course, when considering applications of this paradigm to practical prediction tasks which involve very high dimensional settings and high-throughput rates, classical polynomial-time efficiency may be prohibitive, and there is a need for practically efficient algorithms, usually such that require at most linear (in the dimension) memory and linear runtime per prediction round.

Unfortunately, for several important online tasks such as online linear regression and online PCA, state-of-the-art methods require quadratic memory and at least quadratic runtime per round, which greatly limits their applicability to high dimensional data, see for instance (Garber, 2019a;b) and references therein. Perhaps surprisingly, in both of these cases, this is in clear contrast to their *offline* and even *stochastic* counterparts, which admit significantly more efficient algorithms, under standard well-conditionedness assumptions (e.g., moderate condition number of data matrix in linear regression, or non-negligible eigengap in covariance matrix for PCA). In particular, in both cases these efficient algorithms are simply gradient descent / stochastic gradient descent methods. It is thus natural to ask, if *Online Gradient Descent* (Zinkevich, 2003a), which requires only linear memory and linear runtime (putting aside to gradient computation time) could be shown to be efficient for online variants of such fundamental tasks.

Towards obtaining highly-efficient algorithms for online sequential prediction, in this work we consider a weaker variant of OCO, in which, while we still allow an adversary to choose the set of loss functions (with complete knowledge of the prediction algorithm), we do not allow him to control the order in which they are supplied to the learner. Instead, here we assume that the losses are observed in a random order, according to a uniformly chosen permutation.

tation. We refer to this model as *Random-Order Online Convex Optimization* (ROOCO). In particular, we note that ROOCO forms a natural middle ground between the easier and highly popular setting of prediction with a stochastic adversary, who chooses a distribution over loss functions from a certain family of distributions, and on every round a loss function is sampled i.i.d. from this distribution, and the more difficult standard OCO setting with an oblivious adversary, in which the adversary also controls the order of the loss functions.

It is quite clear that the proposed ROOCO model has several major drawbacks in comparison to the standard OCO model, even when focusing on practical scenarios: it clearly does not capture sequences with temporal structure, such as those we might expect to be inherit in applications such as online portfolio selection (Cover, 2011; Agarwal et al., 2006) or time-series forecasting (Anava et al., 2013). Nevertheless, we argue that many scenarios of interest do not depend inherently on time. For instance in prediction tasks in which data is generated by many independent sources (such as a web application that serves many individual and unrelated clients or a router in a highly distributed computer network), or when it is reasonable to assume that data is streamed without particular order. Such scenarios may indeed be faithfully modeled by ROOCO.

For further motivation, in Figure 1 we report results on a simple online linear regression task with the MNIST dataset (LeCun et al., 1998). We show that in the fully adversarial setting of OCO, we can easily construct pathological, yet not practically reasonable, adversarial orders over the data that make OGD incur cumulative loss significantly higher than under the much more plausible random-order scenario.

Our main algorithmic contribution is in showing that for two types of fundamental applications: the first includes prediction with sequences for which the cumulative loss is strongly convex (though each individual loss function need not even be convex), and hence generalizes as a special case the important task of online linear regression in the well-conditioned case, and the second being online k -PCA in the well-conditioned case (i.e., under spectral gap in the cumulative covariance matrix), the standard online gradient descent method minimizes the regret with high-probability while requiring only linear memory and linear runtime per round (assuming the gradient vector is given). In particular, in the first case of sequences with cumulative strong convexity, we show that OGD attains poly-logarithmic regret with high probability. In the second case of k -PCA, we show that with high probability, OGD only requires to maintain a rank- k projection matrix, which is linear in the size of the solution. Also, each iteration requires only a rank- k SVD which requires nearly linear time to compute (in the size of the solution). This significantly improves over

the current state-of-the-art complexities for the fully adversarial OCO model which require quadratic memory and at least quadratic runtime, for both online linear regression and k -PCA. For sequences of convex losses, even when the cumulative loss is strongly convex, the only method known to obtain logarithmic regret in the fully-adversarial OCO setting, is the Online Newton Step method (ONS), which also require the losses to be exp-concave (Hazan et al., 2007). ONS requires in general quadratic memory to store the sum of outer products of gradients, and at least quadratic runtime to compute to so-called newton direction. For k -PCA all existing methods for the fully adversarial setting require to store in memory the cumulative covariance which requires quadratic memory and/or to compute SVD of potentially high-rank matrices (Warmuth & Kuzmin, 2006a;b; Garber et al., 2015; Allen-Zhu & Li, 2017; Carmon et al., 2019).

On the technical side, we leverage recent developments in matrix concentration bounds for sampling without replacement (Mackey et al., 2014b) as our main non-standard technical tool. For the k -PCA problem we rely on the ideas introduced in (Garber, 2019a) for $k = 1$ and later also used in (Arora & Marinov, 2019) with arbitrary k , to show that with a “warm-start” initialization (which is not difficult to obtain in our setting), the iterates of OGD on the convex relaxation, are of rank k with high probability.

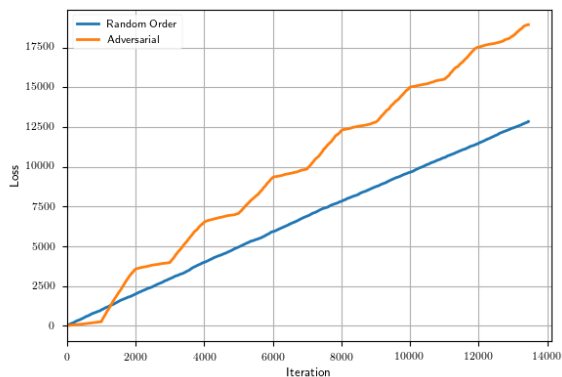


Figure 1. Linear regression with MNIST. This experiment demonstrates how the random order model gives rise to better performance of OGD. We only consider images labeled “3” and “5” (a total of 13454 images). In the adversarial setting, we partition our data into alternating consecutive homogeneous blocks of size 1000, where each block either consists of only “3”-images or only “5”-images. It can be seen that the cumulative loss for the adversarial order case is much worse compared to the random order case.

2. Preliminaries

Given any positive integer m we use the standard notations $[m] := \{1, 2, \dots, m\}$ and $[[m]] := \{0, 1, \dots, m\}$. Matrices are denoted by bold capital letters, vectors are bold lower-case and scalars or entries are not bold. \mathbf{e}_i will denote the i th standard basis vector in \mathbb{R}^d , equal to 1 in component i and 0 everywhere else. The dimension of \mathbf{e}_i will always be clear from context. For vectors in \mathbb{R}^d we let $\|\cdot\|$ denote the Euclidean norm. The spectral norm of a matrix is denoted by $\|\mathbf{X}\| = \sigma_{\max}(\mathbf{X})$ where $\sigma_{\max}(\mathbf{X})$ is the maximum singular value of the matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$. The Euclidean inner product between two matrices is $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{Tr}(\mathbf{X}^\top \mathbf{Y})$, and the corresponding Euclidean norm, called the Frobenius norm is denoted $\|\mathbf{X}\|_F$. That is, $\|\mathbf{X}\|_F = \langle \mathbf{X}, \mathbf{X} \rangle^{1/2}$. The symbols $\lambda_{\max}(\mathbf{X})$ and $\lambda_{\min}(\mathbf{X})$ refer to the algebraic maximum and minimum eigenvalues of a matrix $\mathbf{X} \in \mathbb{R}^{d \times d}$.

Definition 2.1. *Given a convex and compact set $\mathcal{K} \subset \mathbb{R}^d$ and a real-valued function f , differentiable over \mathcal{K} , we say f is G -Lipschitz over \mathcal{K} if for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, we have $\|\nabla f(\mathbf{x})\| \leq G$.*

In particular, if f is convex, differentiable and G -Lipschitz over a convex and compact \mathcal{K} , we have that for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, $f(\mathbf{x}) - f(\mathbf{y}) \leq (\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{x}) \leq G\|\mathbf{x} - \mathbf{y}\|$.

Definition 2.2. *Given a convex and compact set $\mathcal{K} \subset \mathbb{R}^d$ and a real-valued function f which is differentiable over \mathcal{K} , we say f is α -strongly convex over \mathcal{K} if for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, $f(\mathbf{x}) \leq f(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^\top \nabla f(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{x} - \mathbf{y}\|^2$.*

Moreover, if f is also twice differentiable over \mathcal{K} , then, for any $\mathbf{x} \in \mathcal{K}$, $\nabla^2 f(\mathbf{x}) \succeq \alpha \mathbf{I}$.

Definition 2.3. *Given a convex and compact set $\mathcal{K} \subset \mathbb{R}^d$ and a real-valued function f which is differentiable over \mathcal{K} , we say f is b -smooth if and only if its gradient is b -Lipschitz, for some $b > 0$ over \mathcal{K} , i.e., for all $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq b \cdot \|\mathbf{x} - \mathbf{y}\|$. If the function f is twice differentiable over \mathcal{K} , the above condition is equivalent to the following condition on the Hessian, $\nabla^2 f(\mathbf{x}) \preceq b \mathbf{I}$.*

3. Random Order Model

Online Convex Optimization. Online Convex Optimization is posed as a repeated game between a learner and an adversary. First, the adversary chooses T convex loss functions f_1, \dots, f_T from a fixed convex and compact domain $\mathcal{K} \subset \mathbb{R}^d$ to the reals¹. On each iteration $t \in [T]$, the learner makes a prediction $\mathbf{w}_t \in \mathcal{K}$ using some online algorithm \mathcal{A} , and suffers a loss of $f_t(\mathbf{w}_t)$. In addition, we assume that the Euclidean diameter of the set \mathcal{K} is bounded by D , and that for any $\mathbf{w} \in \mathcal{K}$ and all $t \in [T]$, it holds that $\|\nabla f_t(\mathbf{w})\| \leq G$,

¹In this work we focus only on *oblivious* adversaries which choose the entire sequence of losses beforehand, but with knowledge of the algorithm used by the learner.

for some $G > 0$. The goal of the learner is to minimize the regret, which is given by

$$\text{regret}_T(\mathcal{A}) = \sum_{t=1}^T f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{K}} \sum_{t=1}^T f_t(\mathbf{w}).$$

Perhaps the simplest algorithm for OCO is the Online Gradient Descent (OGD) algorithm, which is an adaptation of the classical projected (sub)gradient descent for offline convex optimization. On each iteration, the algorithm picks the next prediction as follows:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{K}}(\mathbf{w}_t - \eta_t \nabla f_t(\mathbf{w}_t)), \quad (1)$$

where for any $\mathbf{w}_0 \in \mathbb{R}^d$ we denote

$$\Pi_{\mathcal{K}}(\mathbf{w}_0) = \arg \min_{\mathbf{w} \in \mathcal{K}} \|\mathbf{w} - \mathbf{w}_0\|.$$

Throughout the paper, we consider a variant of this standard setting: online convex optimization in a *random order model*, in which the adversary can choose the set of loss functions, but cannot control the order in which these loss functions are presented to the learner. Instead, these losses are observed in a uniformly random order. This is formally stated in the following assumption.

Assumption 3.1 (random order assumption). *We say that a sequence of T functions satisfies the random order assumption if each of the functions is chosen by an adversary, but their order is determined by a random permutation over T elements.*

3.1. Cumulative Strongly Convex Sequences

Our first suite of results concerns a family of online tasks, which generalizes online linear and logistic regression in the random order model (Assumption 3.1). As discussed in the introduction, our aim is to obtain poly-logarithmic regret for a sequence of functions whose average is strongly convex, while each of the functions might not even be convex by itself. In this subsection, the learner's prediction is generally assumed to be a vector in \mathbb{R}^d , and is denoted \mathbf{w}_t . Note that the Hessian of each loss function is a $d \times d$ real matrix.

Throughout the discussion, we always assume that the following holds for the loss functions. Given a convex and compact set $\mathcal{K} \subset \mathbb{R}^d$, each of the functions f_1, \dots, f_T is twice differentiable, b -smooth and with gradients bounded in ℓ_2 -norm over the domain \mathcal{K} . Moreover D is the ℓ_2 -diameter of \mathcal{K} . Next we present the main assumption that refers to the Hessians of the loss functions and define "well-conditioned data".

Assumption 3.2 (cumulative α -strong convexity). *Given $\alpha \in \mathbb{R}_+$ and a convex and compact domain $\mathcal{K} \subset \mathbb{R}^d$, we say that a sequence of T twice differentiable functions*

f_1, \dots, f_T over the domain \mathcal{K} satisfies the cumulative α -strong convexity assumption, if

$$\forall \mathbf{w} \in \mathcal{K} : \frac{1}{T} \sum_{t=1}^T \nabla^2 f_t(\mathbf{w}) \succeq \alpha \mathbf{I}.$$

We consider three classes of functions to which our results apply.

Definition 3.3 (quadratic functions). *Let $\mathbf{A}_1, \dots, \mathbf{A}_T$ be a sequence of $d \times d$ real symmetric matrices. We define the loss function $f_t(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\top \mathbf{A}_t \mathbf{w}$ for every $t \in [T]$. The sequence f_1, \dots, f_T satisfies the cumulative α -strong convexity assumption (Assumption 3.2) if $\frac{1}{T} \sum_{t=1}^T \mathbf{A}_t \succeq \alpha \mathbf{I}$.*

Definition 3.4 (strongly convex function composed with linear transformation). *Let $\mathbf{A}_1, \dots, \mathbf{A}_T$ be a sequence of $d \times d$ real matrices, and let g be a β -strongly convex function. We define the loss function $f_t(\mathbf{w}) = g_t(\mathbf{A}_t \mathbf{w})$ for every $t \in [T]$. The sequence f_1, \dots, f_T satisfies the cumulative α -strong convexity assumption (Assumption 3.2) if $\frac{1}{T} \sum_{t=1}^T \mathbf{A}_t^\top \mathbf{A}_t \succeq \frac{\alpha}{\beta} \mathbf{I}$.*

Definition 3.5 (general cumulative strongly convex sequences). *Let f_1, \dots, f_T be a sequence of functions from \mathcal{K} to \mathbb{R} which are twice-differential over \mathcal{K} (but not necessarily convex). The sequence f_1, \dots, f_T satisfies the cumulative α -strong convexity assumption (Assumption 3.2) if the averaged function $\frac{1}{T} \sum_{t=1}^T f_t(\cdot)$ is α -strongly convex over \mathcal{K} .*

Main Results Next we present our main results. In Table 1 we summarize and compare our results to relevant results in two related models. The first is the stochastic model, where loss functions are drawn i.i.d. from a fixed distribution. The second is the adversarial model, where the order of the samples arrives at an arbitrary (possibly adversarial) order. Recall that individual losses are allowed to be non-convex, and we only assume the cumulative loss to be convex.

The following theorem states our main result of this subsection, suggesting that under the random order model (Assumption 3.1), and for a sequence of loss functions that satisfies the cumulative strong convexity assumption (Assumption 3.2), OGD guarantees poly-logarithmic regret.

Theorem 3.6. *Let $\{f_t\}_{t \in [T]}$ be a sequence of loss functions that satisfies the random order model and cumulative α -strong convexity assumptions (Assumptions 3.1 and 3.2 accordingly). Then, for any $\delta \in (0, 1)$, Online Gradient Descent with step sizes $\eta_t = \frac{1}{\alpha t}$ for any $t \in [T]$, gives with probability at least $1 - \delta$ the following for quadratic loss functions (see Definition 3.3)*

$$\text{regret}_T = \tilde{O} \left(\frac{b^2(G + Db)G}{\alpha^3} \log \left(\frac{T}{\delta} \right) \log T \right).$$

For strongly convex functions of a linear transformation (see Definition 3.4), OGD with the same step sizes gives with probability at least $1 - \delta$

$$\text{regret}_T = \tilde{O} \left(\frac{b(G + Db)G}{\alpha^2} \log \left(\frac{T}{\delta} \right) \log T \right).$$

Finally, for general loss functions with cumulative strong convexity (see Definition 3.5), OGD with the same step sizes gives with probability at least $1 - \delta$

$$\text{regret}_T = \tilde{O} \left(\frac{b^2(G + Db)G}{\alpha^3} \left(\log \frac{T}{\delta} + d \right) \log T \right),$$

where in all results we use $\tilde{O}(\cdot)$ to suppress logarithmic dependencies on the dimension and other parameters, but not on T .

In Section 4.1 we provide a proof sketch for Theorem 3.6. The full proof is deferred to the appendix.

The next theorems show that the random order assumption (Assumption 3.1) is crucial for the improved regret bounds in Theorem 3.6. Specifically, there exists a sequences of loss functions that satisfy the cumulative strong convexity assumption (Assumption 3.2), and for which adversarial order makes OGD suffer linear regret in T with at least constant probability; this is proved in Theorem 3.7. Furthermore, in Theorem 3.8, we show that even the additional assumption that each of the loss functions is convex by itself (in combination with cumulative strong convexity) cannot yield a polylogarithmic regret without the random order assumption (Assumption 3.1); specifically, there exists such a sequence where adversarial order results in $\Omega(\sqrt{T})$ regret.

Theorem 3.7. *Let $\mathcal{K} = [0, 1]$. There exists a sequence $\{f_t\}_{t \in [T]}$ of quadratic loss functions from \mathcal{K} to \mathbb{R} , which are $\Theta(1)$ -smooth and satisfy $\Theta(1)$ -cumulative strong convexity (Assumption 3.2), for which OGD with arbitrary positive step-sizes $\{\eta_t\}_{t \in [T]}$ and uniformly random initialization, incurs linear regret with probability at least 0.4.*

Theorem 3.8. *There exist a choice of feasible set \mathcal{K} and a sequence of convex loss functions f_1, \dots, f_T which satisfy $\Theta(1)$ -cumulative strong convexity (Assumption 3.2), such that any algorithm suffers $\Omega(\sqrt{T})$ regret.*

3.2. Online Principal Component Analysis

We recall the problem of online Principal Component Analysis (PCA) (Warmuth & Kuzmin, 2006b; Nie et al., 2013). Fix $1 \leq k < d$, where d is the dimension. In the online setting, for each batch of m data points that arrives on round $t \in [T]$ $\mathbf{X}_t \in \mathbb{R}^{d \times m}$ (i.e., data-points as columns), with covariance matrix $\mathbf{M}_t = \mathbf{X}_t \mathbf{X}_t^\top$, the online algorithm is required to predict a projection matrix $\mathbf{P}_t \in \mathbb{R}^{d \times k}$ onto a k -dimensional subspace of \mathbb{R}^d which belongs to the set

Table 1. Regret guarantees for Online Gradient Descent under various models and assumptions. The \times symbol indicates that a relevant assumption is not needed. All results for the stochastic model are from (Hazan et al., 2007). The \tilde{O} notation hides logarithmic dependencies in d and other parameters, but not in T . SCLT stands for strongly convex functions of linear transformations (see Definition 3.4). For our result with general loss functions we assume T is sub-exponential in d .

MODEL		STRONGLY CONVEX?	CONVEX?	REGRET
STOCHASTIC		\times	IN EXPECTATION	$\Theta(\sqrt{T})$
		IN EXPECTATION	IN EXPECTATION	$\Theta(\frac{G^2}{\alpha} \log T)$
RANDOM ORDER	QUADRATIC (DEF 3.3)	CUMULATIVE	CUMULATIVE	$O\left(\frac{b^2(G+Db)G}{\alpha^3} \log^2 T\right)$ (THEOREM 3.6)
	SCLT (DEF 3.4)	CUMULATIVE	INDIVIDUAL	$\tilde{O}\left(\frac{b(G+Db)G}{\alpha^2} \log^2 T\right)$ (THEOREM 3.6)
	GENERAL (DEF 3.5)	CUMULATIVE	CUMULATIVE	$\tilde{O}\left(\frac{b^2 d(G+Db)G}{\alpha^3} \log^2 T\right)$ (THEOREM 3.6)
ADVERSARIAL		\times / CUMULATIVE	\times / CUMULATIVE	$\Theta(T)$ (THEOREM 3.7)
		\times / CUMULATIVE	INDIVIDUAL	$\Theta(\sqrt{T})$ (THEOREM 3.8)
		\times	INDIVIDUAL	$\Theta(\sqrt{T})$
		INDIVIDUAL	INDIVIDUAL	$\Theta(\frac{G^2}{\alpha} \log T)$

$\mathcal{I}_{d,k} \equiv \{\mathbf{P} : \mathbf{P}^\top \mathbf{P} = \mathbf{I}_k\}$ before observing \mathbf{X}_t . The loss on each round t is then defined as

$$f_t(\mathbf{P}_t) := \frac{1}{2} \|\mathbf{X}_t - \mathbf{P}_t \mathbf{P}_t^\top \mathbf{X}_t\|_F^2.$$

The regret for the online PCA is given by

$$\begin{aligned} \text{regret}_T &= \sum_{t \in [T]} \|\mathbf{X}_t - \mathbf{P}_t \mathbf{P}_t^\top \mathbf{X}_t\|_F^2 \\ &- \min_{\mathbf{P} \in \mathcal{I}_{d,k}} \sum_{t \in [T]} \|\mathbf{X}_t - \mathbf{P} \mathbf{P}^\top \mathbf{X}_t\|_F^2 \\ &= \max_{\mathbf{P} \in \mathcal{I}_{d,k}} \sum_{t \in [T]} \text{Tr}(\mathbf{P} \mathbf{P}^\top \mathbf{M}_t) - \sum_{t \in [T]} \text{Tr}(\mathbf{P}_t \mathbf{P}_t^\top \mathbf{M}_t). \end{aligned} \quad (2)$$

Since Problem (2) is non-convex, we consider a well known convex relaxation that was proposed in (Warmuth & Kuzmin, 2006b) which relaxes it by taking the convex hull

$$\mathcal{S}_{d,k} = \text{convex-hull}\{\mathbf{P} \mathbf{P}^\top \mid \mathbf{P} \in \mathcal{I}_{d,k}\}.$$

Thus, we introduce the symmetric matrix $\mathbf{W}_t \in \mathbb{R}^{d \times d}$, which belongs to the set $\mathcal{S}_{d,k}$. Belonging to $\mathcal{S}_{d,k}$ amounts to satisfying the constraints $\mathbf{W}_t \in \{\mathbf{0} \preceq \mathbf{W} \preceq \mathbf{I}, \text{Tr}(\mathbf{W}) = k\}$ (Warmuth & Kuzmin, 2006b). Using this, we can reformulate Problem (2) and get the following optimization problem with linear payoff functions, allowing us to use known algorithms such as Online Gradient Ascent (OGA).

$$\text{regret}_T = \max_{\mathbf{W} \in \mathcal{S}_{d,k}} \sum_{t \in [T]} \text{Tr}(\mathbf{W} \mathbf{M}_t) - \sum_{t \in [T]} \text{Tr}(\mathbf{W}_t \mathbf{M}_t). \quad (3)$$

In this paper, we assume that the data $\{\mathbf{X}_t\}_{t \in [T]}$ satisfy the random order model (Assumption 3.1), i.e., the matrices $\mathbf{X}_1, \dots, \mathbf{X}_T$ are chosen by an oblivious adversary but

Algorithm 1 batch Online Principal Component Analysis based on Online Gradient Ascent

input: T, L, \mathbf{W}_0 , step sizes $\{\eta_\tau\}_{\tau \geq 0}$
for $\tau = 0, \dots, T/L - 1$ **do**
 play \mathbf{W}_τ and observe the block-averaged covariance matrix $\mathbf{M}_\tau = \frac{1}{L} \sum_{t=\tau L+1}^{(\tau+1)L} \mathbf{X}_t \mathbf{X}_t^\top$
 set $\mathbf{W}_{\tau+1} = \Pi_{\mathcal{S}_{d,k}}(\mathbf{W}_\tau + \eta_\tau \mathbf{M}_\tau)$
end for

are supplied to the learner in a uniformly-chosen random order. Our algorithm, Algorithm 1, is a variant of OGA that predicts over blocks of L consecutive covariance matrices. We denote the average over a length- L block τ of such covariance matrices by $\mathbf{M}_\tau = \frac{1}{L} \sum_{t=\tau L+1}^{(\tau+1)L} \mathbf{M}_t = \frac{1}{L} \sum_{t=\tau L+1}^{(\tau+1)L} \mathbf{X}_t \mathbf{X}_t^\top$, and the average of all covariance matrices as $\mathbf{M} = \frac{1}{T} \sum_{t=1}^T \mathbf{M}_t = \frac{1}{T} \sum_{t=1}^T \mathbf{X}_t \mathbf{X}_t^\top$, where for ease of presentation, throughout the rest of this paper we assume, only slightly losing generality, that T/L is an integer. This allows us to define the following linear payoff function over blocks:

$$F_\tau(\mathbf{W}) := \text{Tr}(\mathbf{W} \mathbf{M}_\tau).$$

Thus, the corresponding regret minimization task, formulated in Problem (3), becomes

$$\text{regret}_T = L \left(\max_{\mathbf{W} \in \mathcal{S}_{d,k}} \sum_{\tau} \text{Tr}(\mathbf{W} \mathbf{M}_\tau) - \sum_{\tau} \text{Tr}(\mathbf{W}_\tau \mathbf{M}_\tau) \right). \quad (4)$$

Algorithm 1 involves the projected-gradient mapping over the set $\mathcal{S}_{d,k}$ and thus requires a full-rank SVD computation in each iteration, at least in worst-case.

Next, we show that under an eigengap assumption this computation can be replaced by a more efficient rank- k SVD computation. First (in Lemma 3.9), we explore how the projection onto $\mathcal{S}_{d,k}$ is constructed. We demonstrate that an eigengap assumption suffices for the projection to have rank at most k .

Lemma 3.9. *Let \mathbf{Y} be a symmetric matrix in $\mathbb{R}^{d \times d}$ and write its eigen-decomposition as $\mathbf{Y} = \sum_{i=1}^d \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$. Suppose that \mathbf{Y} admits an eigengap $\lambda_k(\mathbf{Y}) - \lambda_{k+1}(\mathbf{Y}) \geq 1$. Then, it holds that the Euclidean projection of \mathbf{Y} onto the set $\mathcal{S}_{d,k}$ is of the form:*

$$\Pi_{\mathcal{S}_{d,k}}[\mathbf{Y}] = \sum_{i=1}^k \mathbf{v}_i \mathbf{v}_i^\top.$$

From this lemma we make the following observation, relating between the SVD of a point to project, and the resulting projection.

Observation 3.10 (Low-rank projection, low-rank SVD). *Let \mathbf{Y} be a symmetric $\mathbb{R}^{d \times d}$ matrix. If $\text{rank}(\Pi_{\mathcal{S}_{d,k}}[\mathbf{Y}]) \leq k$, then only the top k components in the SVD of \mathbf{Y} are required to compute the projection. Hence, only a rank- k SVD of \mathbf{Y} is required.*

With the above observation in hand, we present the main results of this section. As reviewed above, it is known that the online PCA problem, when cast as online linear optimization over $\mathcal{S}_{d,k}$, is learnable via a standard variant of Online Gradient Ascent, which achieves an $O(\sqrt{T})$ regret bound, but requires a full SVD computation on each iteration. Next we show that given a sequence of covariance matrices which admits an eigengap with parameter $s > 0$, when partitioning the covariance matrices into length- L blocks with a suitable choice of L , and initializing OGA with a proper "warm-start" matrix (which is straightforward to obtain in our random-order setting), the full SVD computation can be avoided and replaced by only a rank- k SVD computation per iteration (see Theorem 3.11).

Throughout the rest of this paper we assume that the following holds for the covariance matrices. All matrices \mathbf{M}_t for $t \in [T]$ satisfy $\|\mathbf{M}_t\| \leq b$ for some $b > 0$ and also $\|\mathbf{M}_t\|_F \leq C$ for some $C > 0$. Recall also our assumption that T/L is an integer, where L is the size of the blocks used to group the iterations.

Theorem 3.11 (convergence of Algorithm 1). *Let $\delta \in (0, 1)$ and suppose the data $\{\mathbf{M}_t = \mathbf{X}_t \mathbf{X}_t^\top\}_{t \in [T]}$ satisfy the random order assumption (Assumption 3.1). Additionally, suppose the average $\mathbf{M} = \frac{1}{T} \sum_{t=1}^T \mathbf{M}_t$ admits an eigengap $s(\mathbf{M}) := \lambda_k(\mathbf{M}) - \lambda_{k+1}(\mathbf{M}) \geq \rho > 0$. Let $\mathbf{W}^* \in \mathcal{S}_{d,k}$ be the optimal solution of Problem (4). Consider partitioning the matrices $\{\mathbf{M}_t\}_{t \in [T]}$ into length- L blocks with block size $L = \Theta((b^2/\zeta^2) \log(d/\delta))^2$, and running OGA*

for T/L iterations with step sizes $\{\eta_\tau\}_{\tau \in [T/L-1]}$ with the first iterate $\mathbf{W}_0 \in \mathcal{S}_{d,k}$ satisfying $\|\mathbf{W}_0 - \mathbf{W}^\|_F \leq \gamma \zeta$, for $\zeta = \rho/(2 + 6b\gamma)$ and $\gamma = \Theta(\sqrt{k}/\rho)$, and step-sizes $\eta_\tau = \Theta(\sqrt{k}/(C\sqrt{T/L}))$*

For T large enough it holds with probability at least $1 - \delta$ that

$$\text{regret}_T = O((b/\zeta)C\sqrt{kT \log(d/\delta)})$$

and

$$\forall \tau \in [T/L - 1] : \text{rank}(\mathbf{W}_\tau) = k.$$

Note that Theorem 3.11 relies crucially on an *eigengap assumption*, asserting that the k largest eigenvalues of the average of all covariance matrices are strictly larger than all other eigenvalues. This assumption of a non-negligible eigengap in the covariance matrix is very natural for PCA, often observed in practice, and essential for the proof. In Section 4.2 we provide a proof sketch for Theorem 3.11. The full proof is deferred to the appendix.

3.2.1. COMPUTING A "WARM-START" MATRIX

We now discuss the possibility of satisfying the "warm-start" requirement in Theorem 3.11. Recall that \mathbf{W}^* , the optimizer of Problem (4), is the projection of the whole data sequence \mathbf{M} onto the set $\mathcal{S}_{d,k}$, which corresponds to the top k eigenvectors of \mathbf{M} . To compute a warm-start, we can take the covariance matrix over a block \mathbf{M}' of size at least $L_0 := O\left(\frac{b^2}{\zeta_0^2} \log \frac{dT}{\delta}\right)$ from the data, where ζ is a parameter to be determined, and similarly compute the projection of \mathbf{M}' onto the set $\mathcal{S}_{d,k}$, which we denote \mathbf{W}_0 . That is, \mathbf{W}_0 corresponds to the top k eigenvectors of \mathbf{M}' .

Using Davis-Kahan $\sin \theta$ Theorem (see for instance Theorem 2 in (Yu et al., 2014)), we have that

$$\begin{aligned} \|\mathbf{W}_0 - \mathbf{W}^*\|_F &\leq \frac{\sqrt{k} \|\mathbf{M} - \mathbf{M}'\|}{\lambda_k(\mathbf{M}) - \lambda_{k+1}(\mathbf{M})} \\ &\stackrel{(a)}{\leq} \frac{\sqrt{k}}{s(\mathbf{M})} \zeta \leq \frac{\sqrt{k}}{\rho} \zeta_0, \end{aligned}$$

where (a) follows from the choice of L_0 together with an appropriate concentration bound (see Lemma A.11 in the appendix). In particular, for $\zeta_0 = \frac{\rho \gamma \zeta}{\sqrt{k}}$, \mathbf{W}_0 satisfies the requirements of Theorem 3.11 with probability at least $1 - O(\delta)$.

4. Proofs: Main Ideas and Techniques

4.1. Proof Sketch for cumulative strongly convex losses

Here we provide a proof sketch for Theorem 3.6 that discusses the case of cumulative strongly convex losses under the random order assumption (Assumption 3.1).

²For ease of presentation we assume T/L is an integer.

Proof Sketch of Theorem 3.6. The first idea in our analysis is to show that the regret of the original loss sequence can be related to a loss of a batched loss sequence. Thus, letting L be block size whose value will be set later on, we can define a sequence of batched losses, $\forall \tau \in [\lceil T/L \rceil - 1]$,

$$F_\tau(\mathbf{w}) := \frac{1}{L} \sum_{t=\tau L}^{(\tau+1)L} f_t(\mathbf{w}).$$

Also, for any $\tau \in [\lceil T/L \rceil - 1]$ we denote by $\tilde{\mathbf{w}}_\tau$ the value of \mathbf{w}_t at the beginning of each block τ .

Using the above definition together with the OGD update rule, we can show that,

$$\sum_{t=1}^T f_t(\mathbf{w}_t) \leq L \sum_{\tau=0}^{\lceil T/L \rceil} F_\tau(\tilde{\mathbf{w}}_\tau) + \frac{G^2 L}{\alpha} (1 + \log T),$$

and note that the second term in the above expression scales with the block size L .

The second part of our analysis shows that for an appropriately large block size $L \geq L_0$, each batched loss function $F_\tau(\cdot)$ is $(\alpha/2)$ -strongly convex with high probability. To do so we use concentration inequalities for the Hessian of each batched loss $\nabla^2 F_\tau(\cdot)$. Note that since the losses arrive according to a random order permutation, this requires us to build upon specialized Matrix concentration inequalities (Mackey et al., 2014a) that apply to random matrix permutations in the case of quadratic or general functions and a version of Matrix Chernoff (Tropp, 2011) for the case of strongly convex functions of linear transformations. For quadratic loss functions (but not necessarily convex), the Hessian is constant and it is enough to take $L_0 = \tilde{O}\left(\frac{b^2}{\alpha^2} \log(T/\delta)\right)$. For strongly convex functions of linear transformations, $L_0 = \tilde{O}\left(\frac{b}{\alpha} \log(T/\delta)\right)$ suffices, since the losses are convex and hence the Hessians are positive semidefinite, which allows to apply tighter Matrix-Chernoff bounds. For general losses, for which the Hessians cannot be treated as constant over the feasible set \mathcal{K} , we need to take $L_0 = \tilde{O}\left(\frac{db^2}{\alpha^2} \log(T/\delta)\right)$ which follows from using a discretization argument of the set \mathcal{K} and using the union-bound.

The last part of the proof shows that we can use this strong-convexity of the blocks and translate it into a regret bound of $O\left(\frac{(G+Db)GL}{\alpha} \log T\right)$ for the original sequence of losses and decision points. Using the appropriate $L := L_0$ concludes the proof.

It is highly important to emphasize that the partitioning of the losses into blocks of length L is only for the purpose of the regret analysis, while our application of OGD is directly on the original sequence of losses. \square

4.2. Proof Sketch for online PCA

Here we provide a proof sketch for Theorem 3.11 that discusses the case of Online PCA. In the sketch we focus on showing that with high probability,

$$\forall \tau \in [T/L - 1] : \text{rank}(\mathbf{W}_\tau) = k.$$

Given the above holds, the regret analysis for showing $O(\sqrt{T})$ regret is quite standard (see e.g. (Hazan, 2016)).

Proof Sketch of Theorem 3.11. The first part of the proof shows that with probability at least $1 - \delta$ we can bound the spectral norm distance between the average covariance matrix and every batch of size $L \geq L_0$, i.e., $\forall \tau \in [\lceil T/L \rceil - 1]$,

$$\left\| \frac{1}{T} \sum_{t=1}^T \mathbf{M}_t - \frac{1}{L} \sum_{t=\tau L+1}^{(\tau+1)L} \mathbf{M}_t \right\| = \|\mathbf{M} - \mathbf{M}_\tau\| \leq \zeta,$$

where $L_0 = O\left(\frac{b^2}{\zeta^2} \log \frac{dT}{\delta}\right)$.

Conditioning on the event that the above inequality holds, we are then able to show the following,

$$\|\mathbf{W}^* \mathbf{M} \mathbf{W}^* - \mathbf{W}_\tau \mathbf{M} \mathbf{W}_\tau\| = O(b\gamma\zeta). \quad (5)$$

The proof of the above inequality relies on analyzing the OGA update rule combined with exploiting the eigengap property of the data Matrix \mathbf{M} . The intuition is that OGA, when initialized with a ‘‘warm-start’’ matrix, produces with high probability iterates that remain close enough to the optimum.

The next part of the proof works as follows: We assume by induction that for some block τ we have $\text{rank}(\mathbf{W}_\tau) = k$, and use it to show that $\text{rank}(\mathbf{W}_{\tau+1}) = k$ holds as well. This obviously holds for $\tau = 0$ by the assumption of the theorem.

In order to show that the induction step holds we examine the unique structure of the Euclidean projection onto the set $\mathcal{S}_{d,k}$, as captured in Lemma 3.9. We show that $\text{rank}(\mathbf{W}_\tau) = k$ together with Eq. (5) implies that

$$\lambda_k(\mathbf{W}_\tau + \eta_\tau \mathbf{M}_\tau) \geq 1 + \lambda_{k+1}(\mathbf{W}_\tau + \eta_\tau \mathbf{M}_\tau). \quad (6)$$

Combining Eq. (6) together with Lemma 3.9, implies that $\text{rank}(\mathbf{W}_{\tau+1}) = k$, and the low-rank property holds also for the next iterate. \square

5. Experiments

We conduct two sets of experiments demonstrating our theoretical findings for the online PCA problem. (Recall also our experiment from the introduction, which considers the online linear regression problem.)

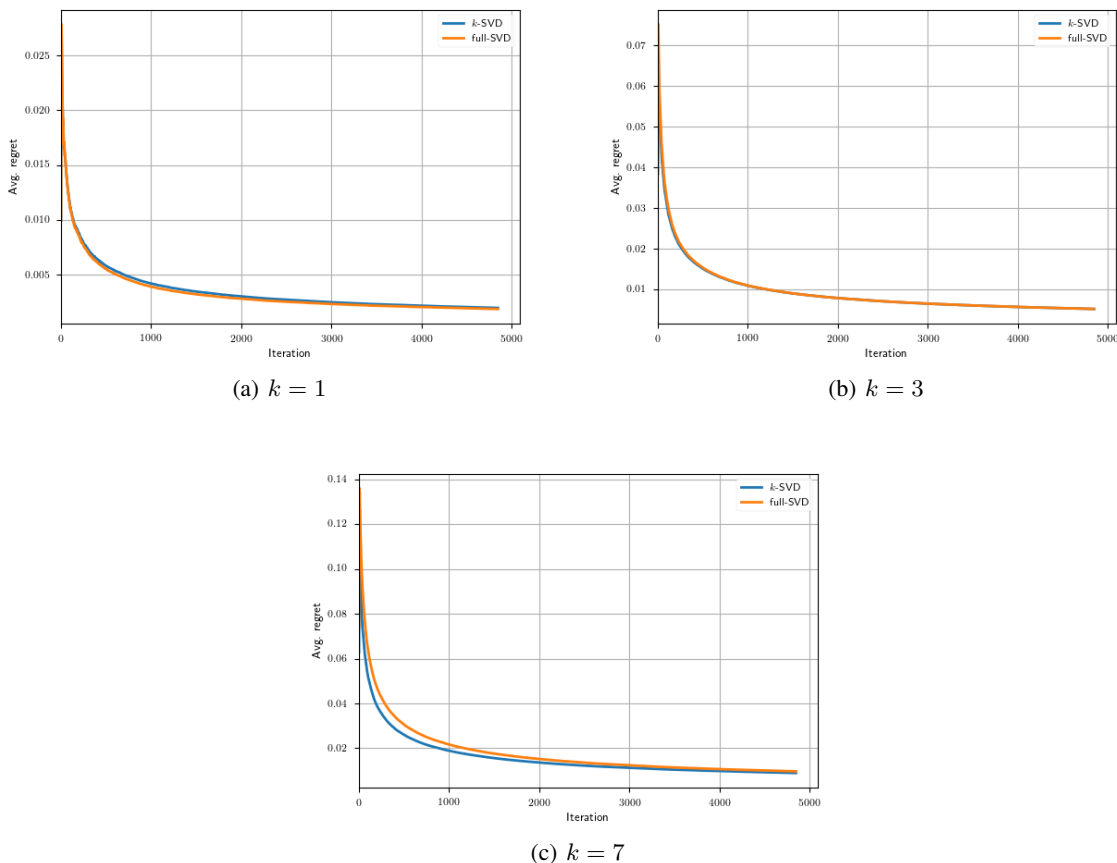


Figure 2. Average regret of full-SVD and k -SVD on synthetic data for $k = \{1, 3, 7\}$. The case of $k = 2$ behaves similarly (see Appendix).

5.1. k -SVD vs. Full-SVD

We compare the average regret of two algorithms: the first is Algorithm 1, where $\mathbf{W}_{\tau+1}$ is computed via rank- k SVD. The second is the same algorithm, but where $\mathbf{W}_{\tau+1}$ is computed via full SVD. We consider the following datasets, where the samples (and therefore also loss functions) arrive in a random order.

Synthetic Data A synthetic dataset generated as follows. The data is sampled from a multi-variate Normal distribution with zero mean and diagonal covariance matrix Σ . For each value of k , we have $\Sigma_{i,i} = 1$ for $1 \leq i \leq k$ and $\Sigma_{i,i} = gap \times 2^{-i \times 0.1}$ for $k+1 \leq i \leq d$. In our experiments $gap = 0.1$, $k = \{1, 2, 3, 7\}$, $d = 1000$, and the window size is $L = 10$. We use 3% of the data to compute the initialization \mathbf{W}_0 , and step sizes $\eta_t = \frac{1}{\sqrt{t}}$. The results on the synthetic data set are the average of 10 experiments and can be found in Figure 2 and in Table 2. It can be seen that even though the dimensionality is high, using low rank-SVD for the projections performs comparably to using full SVD.

Table 2. Results for synthetic dataset. "bad projections" percentage, i.e. percentage of number of iterations in which the projection in Algorithm 1 has rank greater than k , for different values of k and with block size $L = 10$.

SYN DATASET	$k = 1$	$k = 2$	$k = 3$	$k = 7$
BAD PROJ[%]	0.0082	0.012	0.066	0.37

MNIST we use the training set of the MNIST handwritten digit recognition dataset, which contains 60,000 28×28 images, which we split into 58200 images for testing, and 1800 images (3%) are used to compute the initialization \mathbf{W}_0 and step sizes $\eta_t = \frac{1}{10^3 \sqrt{t}}$. We pre-normalized the data by mean centering the feature vectors and scaling each feature by its standard deviation and average the results of 20 experiments. We set $L = 20$ and $k = \{1, 3, 7, 15\}$. The results on the synthetic data set can be found in Figure 3 and in Table 3. Similarly to the experiment on synthetic data, it can be seen that even though the dimensionality is high, low rank SVD for the projections performs comparably to using

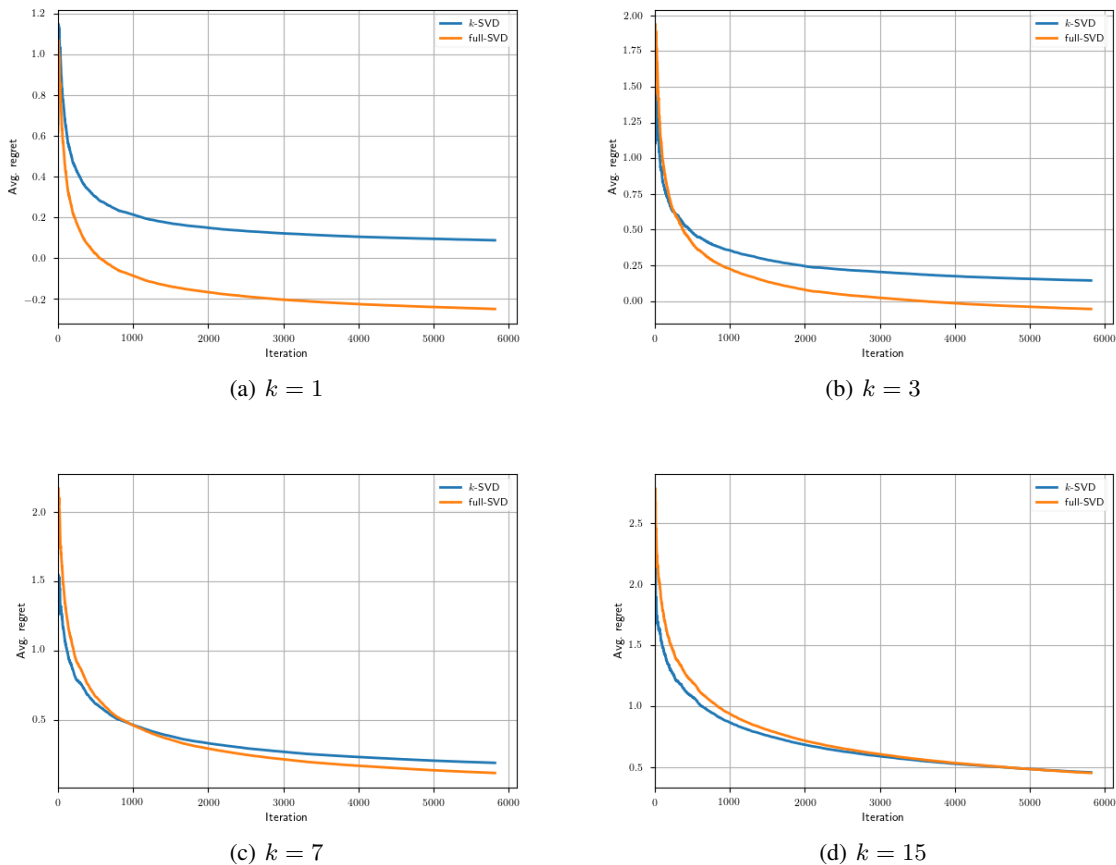


Figure 3. Average regret of full-SVD and k -SVD on MNIST data for $k = \{1, 3, 7, 15\}$.

full SVD.

Table 3. Results for MNIST dataset. "bad projections" percentage, i.e. percentage of number of iterations in which the projection in Algorithm 1 has rank greater than k , for different values of k and with block size $L = 20$.

MNIST DATASET	$k = 1$	$k = 3$	$k = 7$	$k = 15$
BAD PROJ[%]	0	0.23	0.21	0.14

Acknowledgments

Dan Garber is supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1108/18).

References

Agarwal, A., Hazan, E., Kale, S., and Schapire, R. E. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd international conference on Machine learning*, pp. 9–16, 2006.

Allen-Zhu, Z. and Li, Y. Follow the compressed leader: Faster online learning of eigenvectors and faster mmwu. In *International Conference on Machine Learning*, pp. 116–125, 2017.

Anava, O., Hazan, E., Mannor, S., and Shamir, O. Online learning for time series prediction. In *Conference on learning theory*, pp. 172–184, 2013.

Arora, R. and Marinov, T. V. Efficient convex relaxations for streaming PCA. In Wallach, H. M., Larochelle, H.,

- Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 10496–10505, 2019.
- Arora, R., Cotter, A., and Srebro, N. Stochastic optimization of PCA with capped MSG. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pp. 1815–1823, Red Hook, NY, USA, 2013.
- Carmon, Y., Duchi, J. C., Sidford, A., and Tian, K. A rank-1 sketch for matrix multiplicative weights. In Beygelzimer, A. and Hsu, D. (eds.), *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pp. 589–623. PMLR, 2019.
- Cover, T. M. Universal portfolios. In *The Kelly Capital Growth Investment Criterion: Theory and Practice*, pp. 181–209. World Scientific, 2011.
- Garber, D. On the regret minimization of nonconvex online gradient ascent for online pca. In *Conference on Learning Theory*, pp. 1349–1373, 2019a.
- Garber, D. Logarithmic regret for online gradient descent beyond strong convexity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 295–303, 2019b.
- Garber, D., Hazan, E., and Ma, T. Online learning of eigenvectors. In *ICML*, pp. 560–568, 2015.
- Hazan, E. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016.
- Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Mackey, L., Jordan, M. I., Chen, R. Y., Farrell, B., and Tropp, J. A. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014a.
- Mackey, L., Jordan, M. I., Chen, R. Y., Farrell, B., Tropp, J. A., et al. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014b.
- Nie, J., Kotlowski, W., and Warmuth, M. K. Online PCA with optimal regrets. In *24th International Conference on Algorithmic Learning Theory, ALT*, 2013.
- Shalev-Shwartz, S. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.
- Tropp, J. A. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 03(01n02):115–126, 2011.
- Wang, W. and Lu, C. Projection onto the capped simplex. *ArXiv*, abs/1503.01002, 2015.
- Warmuth, M. K. and Kuzmin, D. Online variance minimization. In *19th Annual Conference on Learning Theory, COLT*, 2006a.
- Warmuth, M. K. and Kuzmin, D. Randomized PCA algorithms with regret bounds that are logarithmic in the dimension. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, NIPS*, 2006b.
- Yu, Y., Wang, T., and Samworth, R. J. A Useful Variant of the Davis–Kahan Theorem for Statisticians. *Biometrika*, 102(2):315–323, 2014.
- Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 928–936, 2003a.
- Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003b.