Online Adaptive Principal Component Analysis and Its extensions

Jianjun Yuan [1] **Andrew Lamperski** [1]

Abstract

We propose algorithms for online principal component analysis (PCA) and variance minimization for adaptive settings. Previous literature has focused on upper bounding the static adversarial regret, whose comparator is the optimal fixed action in hindsight. However, static regret is not an appropriate metric when the underlying environment is changing. Instead, we adopt the adaptive regret metric from the previous literature and propose online adaptive algorithms for PCA and variance minimization, that have sub-linear adaptive regret guarantees. We demonstrate both theoretically and experimentally that the proposed algorithms can adapt to the changing environments.

1. Introduction

In the general formulation of online learning, at each time step, the decision maker makes decision without knowing its outcome, and suffers a loss based on the decision and the observed outcome. Loss functions are chosen from a fixed class, but the sequence of losses can be generated deterministically, stochastically, or adversarially.

Online learning is a very popular framework with many variants and applications, such as online convex optimization (Zinkevich, 2003; Shalev-Shwartz et al., 2012), online convex optimization for cumulative constraints (Yuan & Lamperski, 2018), online non-convex optimization (Hazan et al., 2017; Gao et al., 2018), online auctions (Blum et al., 2004), online controller design (Yuan & Lamperski, 2017), and online classification and regression (Crammer et al., 2006). Additionally, recent advances in linear dynamical system identification (Hazan et al., 2018) and reinforcement learning (Fazel et al., 2018) have been developed based on the ideas from online learning.

[1]University of Minnesota, Minneapolis, USA. Correspondence to: Jianjun Yuan <yuanx270@umn.edu>, Andrew Lamperski <alampers@umn.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

The standard performance metric for online learning measures the difference between the decision maker's cumulative loss and the cumulative loss of the best fixed decision in hindsight (Cesa-Bianchi & Lugosi, 2006). We call this metric *static regret*, since the comparator is the best fixed optimum in hindsight. However, when the underlying environment is changing, due to the fixed comparator (Herbster & Warmuth, 1998), static regret is no longer appropriate.

Alternatively, to capture the changes of the underlying environment, (Hazan & Seshadhri, 2009) introduced the metric called adaptive regret, which is defined as the maximum static regret over any contiguous time interval.

In this paper, we are mainly concerned with the problem of online Principal Component Analysis (online PCA) for adaptive settings. Previous online PCA algorithms are based on either online gradient descent or matrix exponentiated gradient algorithms (Tsuda et al., 2005; Warmuth & Kuzmin, 2006; 2008; Nie et al., 2016). These works bound the static regret for online PCA algorithms, but do not address adaptive regret. As argued above, static regret is not appropriate under changing environments.

This paper gives an efficient algorithm for online PCA and variance minimization in changing environments. The proposed method mixes the randomized algorithm from (Warmuth & Kuzmin, 2008) with a fixed-share step (Herbster & Warmuth, 1998). This is inspired by the work of (Cesa-Bianchi et al., 2012b;a), which shows that the Hedge algorithm (Freund & Schapire, 1997) together with a fixed-share step provides low regret under a variety of measures, including adaptive regret.

Furthermore, we extend the idea of the additional fixed-share step to the online adaptive variance minimization in two different parameter spaces: the space of unit vectors and the simplex. In the Section 6 on experiments[1], we also test our algorithm's effectiveness. In particular, we show that our proposed algorithm can adapt to the changing environment faster than the previous online PCA algorithm.

While it is possible to apply the algorithm in (Hazan & Seshadhri, 2009) to solve the online adaptive PCA and variance minimization problems with the similar order of the

[1]code available at https://github.com/yuanx270/online-adaptive-PCA

adaptive regret as in this paper, it requires running a pool of algorithms in parallel. Compared to our algorithm, Running this pool algorithms requires complex implementation that increases the running time per step by a factor of $\log T$.

1.1. Notation

Vectors are denoted by bold lower-case symbols. The i-th element of a vector $\mathbf{q}$ is denoted by q_i. The i-th element of a sequence of vectors at time step t, $\mathbf{x_t}$, is denoted by $x_{t,i}$.

For two probability vectors $\mathbf{q}, \mathbf{w} \in \mathbb{R}^n$, we use $d(\mathbf{q}, \mathbf{w})$ to represent the relative entropy between them, which is defined as $\sum_{i=1}^n q_i \ln(\frac{q_i}{w_i})$. The ℓ_1-norm and ℓ_2-norm of the vector $\mathbf{q}$ are denoted as $\|\mathbf{q}\|_1$, $\|\mathbf{q}\|_2$, respectively. $\mathbf{q_{1:T}}$ is the sequence of vectors $\mathbf{q_1}, \ldots, \mathbf{q_T}$, and $m(\mathbf{q_{1:T}})$ is defined to be equal to $\sum_{t=1}^{T-1} D_{TV}(\mathbf{q_{t+1}}, \mathbf{q_t})$, where $D_{TV}(\mathbf{q_t}, \mathbf{q_{t-1}})$ is defined as $\sum_{i: q_{t,i} \geq q_{t-1,i}} (q_{t,i} - q_{t-1,i})$. The expected value operator is denoted by $\mathbb{E}$.

When we refer to a matrix, we use capital letters such as P and Q with $\|Q\|_2$ representing the spectral norm. For the identity matrix, we use I. The quantum relative entropy between two density matrices[2] P and Q is defined as $\Delta(P, Q) = \mathrm{Tr}(P \ln P) - \mathrm{Tr}(P \ln Q)$, where $\ln P$ is the matrix logarithm for symmetric positive definite matrix P (and $\exp(P)$ is the matrix exponential).

2. Problem Formulation

The goal of the PCA (uncentered) algorithm is to find a rank k projection matrix P that minimizes the compression loss: $\sum_{t=1}^{T} \|\mathbf{x_t} - P\mathbf{x_t}\|_2^2$. In this case, $P \in \mathbb{R}^{n \times n}$ must be a symmetric positive semi-definite matrix with only k non-zero eigenvalues which are all equal to 1.

In online PCA, the data points come in a stream. At each time t, the algorithm first chooses a projection matrix P_t with rank k, then the data point $\mathbf{x_t}$ is revealed, and a compression loss of $\|\mathbf{x_t} - P_t\mathbf{x_t}\|_2^2$ is incurred.

The online PCA algorithm (Warmuth & Kuzmin, 2008) aims to minimize the static regret $\mathcal{R}_s$,which is the difference between the total expected compression loss and the loss of the best projection matrix P^* chosen in hindsight:

$$\mathcal{R}_s = \sum_{t=1}^{T} \mathbb{E}[\mathrm{Tr}((I - P_t)\mathbf{x_t}\mathbf{x_t}^T)] - \sum_{t=1}^{T} \mathrm{Tr}((I - P^*)\mathbf{x_t}\mathbf{x_t}^T). \tag{1}$$

The algorithm from (Warmuth & Kuzmin, 2008) is randomized and the expectation is taken over the distribution of P_t matrices. The matrix P^* is the solution to the following optimization problem with $\mathcal{S}$ being the set of rank-k projection matrices:

$$\min_{P \in \mathcal{S}} \sum_{t=1}^{T} \mathrm{Tr}((I - P)\mathbf{x_t}\mathbf{x_t}^T) \tag{2}$$

Algorithms that minimize static regret will converge to P^*, which is the best projection for the entire data set. However, in many scenarios the data generating process changes over time. In this case, a solution that adapts to changes in the data set may be desirable. To model environmental variation, several notions of dynamically varying regret have been proposed (Herbster & Warmuth, 1998; Hazan & Seshadhri, 2009; Cesa-Bianchi et al., 2012b). In this paper, we study adaptive regret $\mathcal{R}_a$ from (Hazan & Seshadhri, 2009), which results in the following online adaptive PCA problem:

$$\begin{aligned} \mathcal{R}_a = \max_{[r,s] \subset [1,T]} \Big\{ & \sum_{t=r}^{s} \mathbb{E}[\mathrm{Tr}((I - P_t)\mathbf{x_t}\mathbf{x_t}^T)] \\ & - \min_{U \in \mathcal{S}} \sum_{t=r}^{s} \mathrm{Tr}((I - U)\mathbf{x_t}\mathbf{x_t}^T) \Big\} \end{aligned} \tag{3}$$

In the next few sections, we will present an algorithm that achieves low adaptive regret.

3. Learning the Adaptive Best Subset of Experts

In (Warmuth & Kuzmin, 2008) it was shown that online PCA can be viewed as an extension of a simpler problem known as the *best subset of experts* problem. In particular, they first propose an online algorithm to solve the best subset of experts problem, and then they show how to modify the algorithm to solve PCA problems. In this section, we show how the addition of a fixed-share step (Herbster & Warmuth, 1998; Cesa-Bianchi et al., 2012b) can lead to an algorithm for an adaptive variant of the best subset of experts problem. Then we will show how to extend the resulting algorithm to PCA problems.

The *adaptive best subset of experts* problem can be described as follows: we have n experts making decisions at each time t. Before revealing the loss vector $\ell_\mathbf{t} \in \mathbb{R}^n$ associated with the experts' decisions at time t, we select a subset of experts of size $n - k$ (represented by vector $\mathbf{v_t}$) to try to minimize the adaptive regret defined as:

$$\mathcal{R}_a^{\text{subexp}} = \max_{[r,s] \subset [1,T]} \Big\{ \sum_{t=r}^{s} \mathbb{E}[\mathbf{v_t}^T \ell_\mathbf{t}] - \min_{\mathbf{u} \in \mathcal{S}_{\text{vec}}} \sum_{t=r}^{s} \mathbf{u}^T \ell_\mathbf{t} \Big\}. \tag{5}$$

Here, the expectation is taken over the probability distribution of $\mathbf{v_t}$. Both $\mathbf{v_t}$ and $\mathbf{u}$ are in $\mathcal{S}_{\text{vec}}$ which denotes the

[2] A density matrix is a symmetric positive semi-definite matrix with trace equal to 1. Thus, the eigenvalues of a density matrix form a probability vector.

Algorithm 1 Adaptive Best Subset of Experts

1: **Input:** $1 \le k < n$ and an initial probability vector $\mathbf{w_1} \in \mathcal{B}^{\text{n}}_{\text{n-k}}$.
2: **for** $t = 1$ **to** T **do**
3: Use Algorithm 2 with input $d = n - k$ to decompose $\mathbf{w_t}$ into $\sum_j p_j \mathbf{r_j}$, which is a convex combination of at most n corners of $\mathbf{r_j}$.
4: Randomly select a corner $\mathbf{r} = \mathbf{r_j}$ with associated probability p_j.
5: Use the k components with zero entries in the drawn corner $\mathbf{r}$ as the selected subset of experts.
6: Receive loss vector $\boldsymbol{\ell}_\mathbf{t}$.
7: Update $\mathbf{w_{t+1}}$ as:

$$v_{t+1,i} = \frac{w_{t,i}\exp(-\eta\ell_{t,i})}{\sum_{j=1}^{n}\exp(-\eta\ell_{t,j})} \tag{4a}$$

$$\hat{w}_{t+1,i} = \frac{\alpha}{n} + (1-\alpha)v_{t+1,i} \tag{4b}$$

$$\mathbf{w_{t+1}} = \text{cap}_{\text{n-k}}(\mathbf{\hat{w}_{t+1}}) \tag{4c}$$

where $\text{cap}_{\text{n-k}}()$ calls Algorithm 3.
8: **end for**

Algorithm 2 Mixture Decomposition (Warmuth & Kuzmin, 2008)

1: **Input:** $1 \le d < n$ and $\mathbf{w} \in \mathcal{B}^{\text{n}}_{\text{d}}$.
2: **repeat**
3: Let $\mathbf{r}$ be a corner for a subset of d non-zero components of $\mathbf{w}$ that includes all components of $\mathbf{w}$ equal to $\frac{|\mathbf{w}|}{d}$.
4: Let s be the smallest of the d chosen components of $\mathbf{r}$ and l be the largest value of the remaining $n-d$ components.
5: update $\mathbf{w}$ as $\mathbf{w} - \min(ds, |\mathbf{w}| - dl)\mathbf{r}$ and **Output** p and $\mathbf{r}$.
6: **until** $\mathbf{w} = 0$

Algorithm 3 Capping Algorithm (Warmuth & Kuzmin, 2008)

1: **Input:** probability vector $\mathbf{w}$ and set size d.
2: Let $\mathbf{w}^{\downarrow}$ index the vector in decreasing order, that is, $\mathbf{w_1}^{\downarrow} = \max(\mathbf{w})$.
3: **if** $\max(\mathbf{w}) \le 1/d$ **then**
4: **return** $\mathbf{w}$.
5: **end if**
6: $i = 1$.
7: **repeat**
8: (* Set first i largest components to $1/d$ and normalize the rest to $(d-i)/d$ *)
9: $\tilde{\mathbf{w}} = \mathbf{w}$, $\tilde{w}^{\downarrow}_j = 1/d$, for $j = 1, \dots, i$.
10: $\tilde{w}^{\downarrow}_j = \frac{d-i}{d}\frac{\tilde{w}^{\downarrow}_j}{\sum_{l=j}^{n}\tilde{w}^{\downarrow}_l}$, for $j = i+1, \dots, n$.
11: $i = i + 1$.
12: **until** $\max(\tilde{\mathbf{w}}) \le 1/d$.

vector set with only $n-k$ non-zero elements equal to 1.

Similar to the static regret case from (Warmuth & Kuzmin, 2008), the problem in Eq.(5) is equivalent to:

$$\mathcal{R}^{\text{subexp}}_a = \max_{[r,s]\subset[1,T]} \Big\{ \sum_{t=r}^{s}(n-k)\mathbf{w_t}^T\boldsymbol{\ell}_\mathbf{t} - \min_{\mathbf{q}\in\mathcal{B}^{\text{n}}_{\text{n-k}}} \sum_{t=r}^{s}(n-k)\mathbf{q}^T\boldsymbol{\ell}_\mathbf{t} \Big\} \tag{6}$$

where $\mathbf{w_t} \in \mathcal{B}^{\text{n}}_{\text{n-k}}$, and $\mathcal{B}^{\text{n}}_{\text{n-k}}$ represents the capped probability simplex defined as $\sum_{i=1}^{n} w_{t,i} = 1$ and $0 \le w_{t,i} \le 1/(n-k), \forall i$.

Such equivalence is due to the Theorem 2 in (Warmuth & Kuzmin, 2008) ensuring that any vector $\mathbf{q} \in \mathcal{B}^{\text{n}}_{\text{n-k}}$ can be decomposed as convex combination of at most n corners of $\mathbf{r_j}$ by using Algorithm 2, where the corner $\mathbf{r_j}$ is defined as having $n-k$ non-zero elements equal to $1/(n-k)$. As a result, the corner can be sampled by the associated probability obtained from the convex combination, which is a valid subset selection vector $\mathbf{v_t}$ with the multiplication of $n-k$.

Connection to the online adaptive PCA. The problem from Eq.(5) can be viewed as restricted version of the online adaptive PCA problem from Eq.(3). In particular, say that $I - P_t = \text{diag}(\mathbf{v_t})$. This corresponds to restricting P_t to be diagonal. If $\boldsymbol{\ell}_\mathbf{t}$ is the diagonal of $\mathbf{x_t}\mathbf{x_t}^T$, then the objectives of Eq.(5) and Eq.(3) are equal.

We now return to the adaptive best subset of experts problem. When $r = 1$ and $s = T$, the problem reduces to the standard static regret minimization problem, which is studied in (Warmuth & Kuzmin, 2008). Their solution applies the basic Hedge Algorithm to obtain a probability distribution for the experts, and modifies the distribution to select a subset of the experts.

To deal with the adaptive regret considered in Eq.(6), we propose the Algorithm 1, which is a simple modification to Algorithm 1 in (Warmuth & Kuzmin, 2008). More specifically, we add Eq.(4b) when updating $\mathbf{w_{t+1}}$ in Step 7, which is called a *fixed-share* step. This is inspired by the analysis in (Cesa-Bianchi et al., 2012b), which shows that the online adaptive best expert problem can be solved by simply adding this fixed-share step to the standard Hedge algorithm.

With the Algorithm 1, the following lemma can be obtained:

Lemma 1. *For all $t \ge 1$, all $\boldsymbol{\ell}_\mathbf{t} \in [0,1]^n$, and for all $\mathbf{q_t} \in \mathcal{B}^n_{n\text{-}k}$, Algorithm 1 satisfies*

$$\mathbf{w_t}^T\boldsymbol{\ell}_\mathbf{t}(1-\exp(-\eta)) - \eta\mathbf{q_t}^T\boldsymbol{\ell}_\mathbf{t} \le \sum_{i=1}^{n} q_{t,i}\ln\Big(\frac{v_{t+1,i}}{\hat{w}_{t,i}}\Big)$$

Proof. With the update in Eq.(4), for any $\mathbf{q_t} \in \mathcal{B}^{\text{n}}_{\text{n-k}}$, we have

$$d(\mathbf{q_t}, \mathbf{w_t}) - d(\mathbf{q_t}, \mathbf{v_{t+1}}) = -\eta \mathbf{q_t}^T \ell_\mathbf{t} - \ln(\sum_{j=1}^{n} w_{t,j} \exp(-\eta \ell_{t,j})) \tag{7}$$

Also, from the proof of Theorem 1 in (Warmuth & Kuzmin, 2008), we have $-\ln(\sum_{j=1}^n w_{t,j} \exp(-\eta \ell_{t,j})) \geq \mathbf{w_t}^T \ell_\mathbf{t}(1 - \exp(-\eta))$. Thus, we will get

$$d(\mathbf{q_t}, \mathbf{w_t}) - d(\mathbf{q_t}, \mathbf{v_{t+1}}) \geq -\eta \mathbf{q_t}^T \ell_\mathbf{t} + \mathbf{w_t}^T \ell_\mathbf{t}(1 - \exp(-\eta)) \tag{8}$$

Moreover, Eq.(4c) is the solution to the following projection problem as shown in (Warmuth & Kuzmin, 2008):

$$\mathbf{w_t} = \operatorname*{argmin}_{\mathbf{w} \in \mathcal{B}^{\text{n}}_{\text{n-k}}} d(\mathbf{w}, \hat{\mathbf{w}}_\mathbf{t}) \tag{9}$$

Since the relative entropy is one kind of Bregman divergence (Bregman, 1967; Censor & Lent, 1981), the Generalized Pythagorean Theorem holds (Herbster & Warmuth, 2001):

$$d(\mathbf{q_t}, \hat{\mathbf{w}}_\mathbf{t}) - d(\mathbf{q_t}, \mathbf{w_t}) \geq d(\mathbf{w_t}, \hat{\mathbf{w}}_\mathbf{t}) \geq 0 \tag{10}$$

where the last inequality is due to the non-negativity of Bregman divergence.

Combining Eq.(8) with Eq.(10) and expanding the left part of $d(\mathbf{q_t}, \hat{\mathbf{w}}_\mathbf{t}) - d(\mathbf{q_t}, \mathbf{v_{t+1}})$, we arrive at Lemma 1. □

Now we are ready to state the following theorem to upper bound the adaptive regret $\mathcal{R}_a^{\text{subexp}}$:

Theorem 1. *If we run the Algorithm 1 to select a subset of $n-k$ experts, then for any sequence of loss vectors $\ell_\mathbf{1}, \ldots, \ell_\mathbf{T} \in [0,1]^n$ with $T \geq 1$, $\min_{\mathbf{q} \in \mathcal{B}^n_{n\text{-}k}} \sum_{t=r}^{s} (n-k) \mathbf{q}^T \ell_\mathbf{t} \leq L$, $\alpha = 1/(T(n-k)+1)$, $D = (n-k)\ln(n(1+(n-k)T)) + 1$, and $\eta = \ln(1 + \sqrt{2D/L})$, we have*

$$\mathcal{R}_a^{subexp} \leq O(\sqrt{2LD} + D)$$

Proof sktech. After showing the inequality from Lemma 1, the main work that remains is to sum the right side from $t = 1$ to T and provide an upper bound. This is achieved by following the proof of the Proposition 2 in (Cesa-Bianchi et al., 2012b). The main idea is to expand the term $\sum_{i=1}^n q_{t,i} \ln(\frac{v_{t+1,i}}{\hat{w}_{t,i}})$ as follows:

$$\sum_{i=1}^n q_{t,i} \ln(\frac{v_{t+1,i}}{\hat{w}_{t,i}}) = \underbrace{\sum_{i=1}^{n} \Big(q_{t,i} \ln \frac{1}{\hat{w}_{t,i}} - q_{t-1,i} \ln \frac{1}{v_{t,i}}\Big)}_{A} + \underbrace{\sum_{i=1}^{n} \Big(q_{t-1,i} \ln \frac{1}{v_{t,i}} - q_{t,i} \ln \frac{1}{v_{t+1,i}}\Big)}_{B} \tag{11}$$

Then we can upper bound the expression of A with the *fixed-share* step, since $\hat{w}_{t,i}$ is lower bounded by $\frac{\alpha}{n}$. We can telescope the expression of B. Then our desired upper bound can be obtained with the help of Lemma 4 from (Freund & Schapire, 1997). □

For space purposes, all the detailed proofs for the omitted/sketched proofs are in the appendix.

4. Online Adaptive PCA

Recall that the online adaptive PCA problem is below:

$$\mathcal{R}_a = \max_{[r,s] \subset [1,T]} \Big\{ \sum_{t=r}^{s} \mathbb{E}[\mathrm{Tr}((I - P_t)\mathbf{x_t}\mathbf{x_t}^T)] - \min_{U \in \mathcal{S}} \sum_{t=r}^{s} \mathrm{Tr}((I - U)\mathbf{x_t}\mathbf{x_t}^T) \Big\} \tag{12}$$

where $\mathcal{S}$ is the rank k projection matrix set.

Again, inspired by (Warmuth & Kuzmin, 2008), we first reformulate the above problem into the following 'capped probability simplex' form:

$$\mathcal{R}_a = \max_{[r,s] \subset [1,T]} \Big\{ \sum_{t=r}^{s} (n-k) \mathrm{Tr}(W_t \mathbf{x_t}\mathbf{x_t}^T) - \min_{Q \in \mathscr{B}^{\text{n}}_{\text{n-k}}} \sum_{t=r}^{s} (n-k) \mathrm{Tr}(Q \mathbf{x_t}\mathbf{x_t}^T) \Big\} \tag{13}$$

where $W_t \in \mathscr{B}^{\text{n}}_{\text{n-k}}$, and $\mathscr{B}^{\text{n}}_{\text{n-k}}$ is the set of all density matrices with eigenvalues bounded by $1/(n-k)$. Note that $\mathscr{B}^{\text{n}}_{\text{n-k}}$ can be expressed as the convex set $\{W : W \succeq 0, \|W\|_2 \leq 1/(n-k), \mathrm{Tr}(W) = 1\}$.

The static regret online PCA is a special case of the above problem with $r = 1$ and $s = T$, and is solved by Algorithm 5 in (Warmuth & Kuzmin, 2008).

Follow the idea in the last section, we propose the Algorithm 4. Compared with the Algorithm 5 in (Warmuth & Kuzmin, 2008), we have added the fixed-share step in the update of W_{t+1} at step 9, which will be shown to be the key in upper bounding the adaptive regret of the online PCA.

In order to analyze Algorithm 4, we need a few supporting results. The first result comes from (Warmuth & Kuzmin, 2006):

Theorem 2. *(Warmuth & Kuzmin, 2006) For any sequence of data points $\mathbf{x_1}, \ldots, \mathbf{x_T}$ with $\mathbf{x_t}\mathbf{x_t}^T \preceq I$ and for any learning rate η, the following bound holds for any matrix $Q_t \in \mathscr{B}^n_{n\text{-}k}$ with the update in Eq.(14a):*

$$\mathrm{Tr}(W_t \mathbf{x_t}\mathbf{x_t}^T) \leq \frac{\Delta(Q_t, W_t) - \Delta(Q_t, V_{t+1}) + \eta \mathrm{Tr}(Q_t \mathbf{x_t}\mathbf{x_t}^T)}{1 - \exp(-\eta)}$$

Based on the above theorem's result, we have the following lemma:

Lemma 2. *For all $t \geq 1$, all $\mathbf{x_t}$ with $\|\mathbf{x_t}\|_2 \leq 1$, and for*

Algorithm 4 Uncentered online adaptive PCA

Input: $1 \le k < n$ and an initial density matrix $W_1 \in \mathcal{B}^n_{n-k}$.
for $t = 1$ **to** T **do**
  Apply eigendecomposition to W_t as $W_t = \bar{D}\operatorname{diag}(\mathbf{w_t})\bar{D}^T$.
  Apply Algorithm 2 with $d = n-k$ to the vector $\mathbf{w_t}$ to decompose it into a convex combination $\sum_j p_j \mathbf{r_j}$ of at most n corners $\mathbf{r_j}$.
  Randomly select a corner $\mathbf{r} = \mathbf{r_j}$ with the associated probability p_j.
  Form a density matrix $R = (n-k)\bar{D}\operatorname{diag}(\mathbf{r})\bar{D}^T$
  Form a rank k projection matrix $P_t = I - R$
  Obtain the data point $\mathbf{x_t}$, which incurs the compression loss $\|\mathbf{x_t} - P_t\mathbf{x_t}\|_2^2$ and expected compression loss $(n-k)\operatorname{Tr}(W_t\mathbf{x_t}\mathbf{x_t}^T)$.
  Update W_{t+1} as:

$$V_{t+1} = \frac{\exp(\ln W_t - \eta\mathbf{x_t}\mathbf{x_t}^T)}{\operatorname{Tr}(\exp(\ln W_t - \eta\mathbf{x_t}\mathbf{x_t}^T))} \tag{14a}$$

$$\hat{w}_{t+1,i} = \frac{\alpha}{n} + (1-\alpha)v_{t+1,i},\, \widehat{W}_{t+1} = U\operatorname{diag}(\hat{\mathbf{w}}_{\mathbf{t+1}})U^T \tag{14b}$$

$$W_{t+1} = \operatorname{cap}_{n-k}(\widehat{W}_{t+1}) \tag{14c}$$

  where we apply eigendecomposition to V_{t+1} as $V_{t+1} = U\operatorname{diag}(\mathbf{v_{t+1}})U^T$, and $\operatorname{cap}_{n-k}(W)$ invokes Algorithm 3 with input being the eigenvalues of W.
end for

all $Q_t \in \mathcal{B}^n_{n-k}$, Algorithm 4 satisfies:

$$\begin{aligned}\operatorname{Tr}(W_t\mathbf{x_t}\mathbf{x_t}^T)(1-\exp(-\eta)) - \eta\operatorname{Tr}(Q_t\mathbf{x_t}\mathbf{x_t}^T) \\ \le -\operatorname{Tr}(Q_t\ln\widehat{W}_t) + \operatorname{Tr}(Q_t\ln V_{t+1})\end{aligned} \tag{15}$$

Proof. First, we need to reformulate the above inequality in Theorem 2, we have:

$$\begin{aligned}&\Delta(Q_t, W_t) - \Delta(Q_t, V_{t+1}) \\ &\quad\ge -\eta\operatorname{Tr}(Q_t\mathbf{x_t}\mathbf{x_t}^T) + \operatorname{Tr}(W_t\mathbf{x_t}\mathbf{x_t}^T)(1-\exp(-\eta))\end{aligned} \tag{16}$$

which is very similar to the Eq.(8).

As is shown in (Warmuth & Kuzmin, 2008), the Eq.(14c) is the solution to the following optimization problem:

$$W_t = \operatorname*{argmin}_{W\in\mathcal{B}^{\text{n}}_{\text{n-k}}} \Delta(W, \widehat{W}_t) \tag{17}$$

As a result, the Generalized Pythagorean Theorem holds (Herbster & Warmuth, 2001) for any $Q_t \in \mathcal{B}^{\text{n}}_{\text{n-k}}$:

$$\Delta(Q_t, \widehat{W}_t) - \Delta(Q_t, W_t) \ge \Delta(W_t, \widehat{W}_t) \ge 0 \tag{18}$$

Combining the above inequality with Eq.(16) and expanding the left part, we have

$$\begin{aligned}\operatorname{Tr}(W_t\mathbf{x_t}\mathbf{x_t}^T)(1-\exp(-\eta)) - \eta\operatorname{Tr}(Q_t\mathbf{x_t}\mathbf{x_t}^T) \\ \le -\operatorname{Tr}(Q_t\ln\widehat{W}_t) + \operatorname{Tr}(Q_t\ln V_{t+1})\end{aligned} \tag{19}$$

which proves the result. □

In the next theorem, we show that with the addition of the fixed-share step in Eq.(14b), we can solve the online adaptive PCA problem in Eq.(12).

Theorem 3. *For any sequence of data points $\mathbf{x_1}, \ldots, \mathbf{x_T}$ with $\|\mathbf{x_t}\|_2 \le 1$, and for $\min_{Q\in\mathcal{B}^n_{n\text{-}k}} \sum_{t=r}^{s}(n-k)\operatorname{Tr}(Q\mathbf{x_t}\mathbf{x_t}^T) \le L$, if we run Algorithm 4 with $\alpha = 1/(T(n-k)+1)$, $D = (n-k)\ln(n(1+(n-k)T))+1$, and $\eta = \ln(1+\sqrt{2D/L})$, for any $T \ge 1$ we have:*

$$\mathcal{R}_a \le O(\sqrt{2LD} + D)$$

Proof sktech. The proof idea is the same as in the proof of Theorem 1. After getting the inequality relationship in Lemma 2 which has a similar form as in Lemma 1, we need to upper bound sum over t of the right side. To achieve this, we first reformulate it as two parts below:

$$-\operatorname{Tr}(Q_t\ln\widehat{W}_t) + \operatorname{Tr}(Q_t\ln V_{t+1}) = \bar{A} + \bar{B} \tag{20}$$

where $\bar{A} = -\operatorname{Tr}(Q_t\ln\widehat{W}_t) + \operatorname{Tr}(Q_{t-1}\ln V_t)$, and $\bar{B} = -\operatorname{Tr}(Q_{t-1}\ln V_t) + \operatorname{Tr}(Q_t\ln V_{t+1})$.

The first part can be upper bounded with the help of the fixed-share step in lower bounding the singular value of $\hat{w}_{t,i}$. After telescoping the second part, we can get the desired upper bound with the help of Lemma 4 from (Freund & Schapire, 1997). □

5. Extension to Online Adaptive Variance Minimization

In this section, we study the closely related problem of online adaptive variance minimization. The problem is defined as follows: At each time t, we first select a vector $\mathbf{y_t} \in \Omega$, and then a covariance matrix $C_t \in \mathbb{R}^{n\times n}$ such that $0 \preceq C_t \preceq I$ is revealed. The goal is to minimize the adaptive regret defined as:

$$\mathcal{R}_a^{\text{var}} = \max_{[r,s]\subset[1,T]}\left\{\sum_{t=r}^{s}\mathbb{E}[\mathbf{y_t}^TC_t\mathbf{y_t}] - \min_{\mathbf{u}\in\Omega}\sum_{t=r}^{s}\mathbf{u}^TC_t\mathbf{u}\right\} \tag{21}$$

where the expectation is taken over the probability distribution of $\mathbf{y_t}$.

This problem has two different situations corresponding to different parameter space Ω of $\mathbf{y_t}$ and $\mathbf{u}$.

Situation 1: When Ω is the set of $\{\mathbf{x}| \, \|\mathbf{x}\|_2 = 1\}$ (e.g., the unit vector space), the solution to $\min_{\mathbf{u}\in\Omega} \sum_{t=r}^{s} \mathbf{u}^T C_t \mathbf{u}$ is the minimum eigenvector of the matrix $\sum_{t=r}^{s} C_t$.

Situation 2: When Ω is the probability simplex (e.g., Ω is equal to $\mathcal{B}_1^{\mathrm{n}}$), it corresponds to the risk minimization in stock portfolios (Markowitz, 1952).

We will start with **Situation 1** since it is highly related to the previous section.

5.1. Online Adaptive Variance Minimization over the Unit vector space

We begin with the observation of the following equivalence (Warmuth & Kuzmin, 2006):

$$\min_{\|\mathbf{u}\|_2=1} \mathbf{u}^T C \mathbf{u} = \min_{U\in\mathcal{B}_1^{\mathrm{n}}} \operatorname{Tr}(UC) \tag{22}$$

where C is any covariance matrix, and $\mathcal{B}_1^{\mathrm{n}}$ is the set of all density matrices.

Thus, the problem in (21) can be reformulated as:

$$\mathcal{R}_a^{\text{var-unit}} = \max_{[r,s]\subset[1,T]} \left\{ \sum_{t=r}^{s} \operatorname{Tr}(Y_t C_t) - \min_{U\in\mathcal{B}_1^{\mathrm{n}}} \sum_{t=r}^{s} \operatorname{Tr}(UC_t) \right\} \tag{23}$$

where $Y_t \in \mathcal{B}_1^{\mathrm{n}}$.

To see the equivalence between $\mathbb{E}[\mathbf{y_t}^T C_t \mathbf{y_t}]$ in Eq.(21) and $\operatorname{Tr}(Y_t C_t)$, we do the eigendecomposition of $Y_t = \sum_{i=1}^{n} \sigma_i \mathbf{y_i}\mathbf{y_i}^T$. Then $\operatorname{Tr}(Y_t C_t)$ is equal to $\sum_{i=1}^{n} \sigma_i \operatorname{Tr}(\mathbf{y_i}\mathbf{y_i}^T C_t) = \sum_{i=1}^{n} \sigma_i \mathbf{y_i}^T C_t \mathbf{y_i}$. Since $Y_t \in \mathcal{B}_1^{\mathrm{n}}$, the vector σ is a simplex vector, and $\sum_{i=1}^{n} \sigma_i \mathbf{y_i}^T C_t \mathbf{y_i}$ is equal to $\mathbb{E}[\mathbf{y_i}^T C_t \mathbf{y_i}]$ with probability distribution defined by the vector σ.

If we examine Eq.(23) and (13) together, we will see that they share some similarities: First, they are almost the same if we set $n-k=1$ in Eq.(13). Also, $\mathbf{x_t}\mathbf{x_t}^T$ in Eq.(13) is a special case of C_t in Eq.(23).

Thus, it is possible to apply Algorithm 4 to solving the problem (23) by setting $n-k=1$. In this case, Algorithms 2 and 3 are not needed. This is summarized in Algorithm 5.

The theorem below is analogous to Theorem 3 in the case that $n-k=1$.

Theorem 4. *For any sequence of covariance matrices $C_1, \ldots, C_T$ with $0 \preceq C_t \preceq I$, and for $\min_{U\in\mathcal{B}_1^n} \sum_{t=r}^{s} \operatorname{Tr}(UC_t) \le L$, if we run Algorithm 5 with $\alpha = 1/(T+1)$, $D = \ln(n(1+T)) + 1$, and $\eta = \ln(1+\sqrt{2D/L})$, for any $T \ge 1$ we have:*

$$\mathcal{R}_a^{\textit{var-unit}} \le O(\sqrt{2LD} + D)$$

Algorithm 5 Online adaptive variance minimization over unit sphere

---

1: **Input:** an initial density matrix $Y_1 \in \mathcal{B}_1^n$.
2: **for** $t = 1$ **to** T **do**
3: Perform eigendecomposition $Y_t = \widehat{D} \operatorname{diag}(\sigma_t) \widehat{D}^T$.
4: Use the vector $\mathbf{y_t} = \widehat{D}[:, j]$ with probability $\sigma_{t,j}$.
5: Receive covariance matrix C_t, which incurs the loss $\mathbf{y_t}^T C_t \mathbf{y_t}$ and expected loss $\operatorname{Tr}(Y_t C_t)$.
6: Update Y_{t+1} as:

$$V_{t+1} = \frac{\exp(\ln Y_t - \eta C_t)}{\operatorname{Tr}(\exp(\ln Y_t - \eta C_t))} \tag{24a}$$

$$\sigma_{t+1,i} = \frac{\alpha}{n} + (1-\alpha) v_{t+1,i}, Y_{t+1} = \widehat{U} \operatorname{diag}(\sigma_{t+1}) \widehat{U}^T \tag{24b}$$

where we apply eigendecomposition to V_{t+1} as $V_{t+1} = \widehat{U} \operatorname{diag}(\mathbf{v_{t+1}}) \widehat{U}^T$.
7: **end for**

Proof sktech. Similar inequality can be obtained as in Lemma 2 by using the result of Theorem 2 in (Warmuth & Kuzmin, 2006). The rest follows the proof of Theorem 3. □

In order to apply the above theorem, we need to either estimate the step size η heuristically or estimate the upper bound L, which may not be easily done.

In the next theorem, we show that we can still upper bound the $\mathcal{R}_a^{\text{var-unit}}$ without knowing L, but the upper bound is a function of time horizon T instead of the upper bound L.

Before we get to the theorem, we need the following lemma which lifts the vector case of Lemma 1 in (Cesa-Bianchi et al., 2012b) to the density matrix case:

Lemma 3. *For any $\eta \ge 0$, $t \ge 1$, any covariance matrix C_t with $0 \preceq C_t \preceq I$, and for any $Q_t \in \mathcal{B}_1^n$, Algorithm 5 satisfies:*

$$\operatorname{Tr}(Y_t C_t) - \operatorname{Tr}(Q_t C_t) \le \frac{1}{\eta}\Big(\operatorname{Tr}(Q_t \ln V_{t+1}) - \operatorname{Tr}(Q_t \ln Y_t)\Big) + \frac{\eta}{2}$$

Now we are ready to present the upper bound on the regret for Algorithm 5.

Theorem 5. *For any sequence of covariance matrices $C_1, \ldots, C_T$ with $0 \preceq C_t \preceq I$, if we run Algorithm 5 with $\alpha = 1/(T+1)$ and $\eta = \frac{\sqrt{\ln(n(1+T))}}{\sqrt{T}}$, for any $T \ge 1$ we have:*

$$\mathcal{R}_a^{\textit{var-unit}} \le O\Big(\sqrt{T \ln\big(n(1+T)\big)}\Big)$$

Proof. In the proof, we will use two cases of Q_t: $Q_t \in \mathcal{B}_1^n$, and $Q_t = 0$.

From Lemma 3, the following inequality is valid for both cases of Q_t:

$$\begin{aligned}\mathrm{Tr}(Y_tC_t) - \mathrm{Tr}(Q_tC_t) \\ \leq \frac{1}{\eta}\Big(\mathrm{Tr}(Q_t \ln V_{t+1}) - \mathrm{Tr}(Q_t \ln Y_t)\Big) + \frac{\eta}{2}\end{aligned} \tag{25}$$

Follow the same analysis as in the proof of Theorem 3, we first do the eigendecomposition to Q_t as $Q_t = \widetilde{D}\,\mathrm{diag}(q_t)\widetilde{D}^T$. Since $\|q_t\|_1$ is either 1 or 0, we will re-write the above inequality as:

$$\begin{aligned}\|q_t\|_1\,\mathrm{Tr}(Y_tC_t) - \mathrm{Tr}(Q_tC_t) \\ \leq \frac{1}{\eta}\Big(\mathrm{Tr}(Q_t \ln V_{t+1}) - \mathrm{Tr}(Q_t \ln Y_t)\Big) + \frac{\eta}{2}\|q_t\|_1\end{aligned} \tag{26}$$

Analyzing the term $\mathrm{Tr}(Q_t \ln V_{t+1}) - \mathrm{Tr}(Q_t \ln Y_t)$ in the above inequality is the same as the analysis of the Eq.(43) in the appendix.

Thus, summing over $t = 1$ to T to the above inequality, and setting $Q_t = Q \in \mathcal{B}_1^n$ for $t = r, \ldots, s$ and 0 elsewhere, we will have

$$\begin{aligned}\sum_{t=r}^{s} \mathrm{Tr}(Y_tC_t) - \min_{U \in \mathcal{B}_1^n} \sum_{t=r}^{s} \mathrm{Tr}(UC_t) \\ \leq \frac{1}{\eta}\Big(\ln\frac{n}{\alpha} + T\ln\frac{1}{1-\alpha}\Big) + \frac{\eta}{2}T,\end{aligned} \tag{27}$$

since it holds for any $Q \in \mathcal{B}_1^n$.

After plugging in the expression of η and α, we will have

$$\begin{aligned}\sum_{t=r}^{s} \mathrm{Tr}(Y_tC_t) - \min_{U \in \mathcal{B}_1^n} \sum_{t=r}^{s} \mathrm{Tr}(UC_t) \\ \leq O\Big(\sqrt{T\ln\big(n(1+T)\big)}\Big)\end{aligned} \tag{28}$$

Since the above inequality holds for any $1 \leq r \leq s \leq T$, we will put a $\max\limits_{[r,s]\subset[1,T]}$ in the left part, which proves the result. $\square$

5.2. Online Adaptive Variance Minimization over the Simplex space

We first re-write the problem in Eq.(21) when Ω is the simplex below:

$$\mathcal{R}_a^{\text{var-sim}} = \max_{[r,s]\subset[1,T]} \Big\{ \sum_{t=r}^{s} \mathbb{E}[\mathbf{y_t}^T C_t \mathbf{y_t}] - \min_{\mathbf{u}\in\mathcal{B}_1^n} \sum_{t=r}^{s} \mathbf{u}^T C_t \mathbf{u} \Big\} \tag{29}$$

where $\mathbf{y_t} \in \mathcal{B}_1^n$, and $\mathcal{B}_1^n$ is the simplex set.

When $r = 1$ and $s = T$, the problem reduces to the static regret problem, which is solved in (Warmuth & Kuzmin, 2006) by the exponentiated gradient algorithm as below:

$$y_{t+1,i} = \frac{y_{t,i}\exp\big(-\eta(C_t\mathbf{y_t})_i\big)}{\sum_i y_{t,i}\exp\big(-\eta(C_t\mathbf{y_t})_i\big)} \tag{30}$$

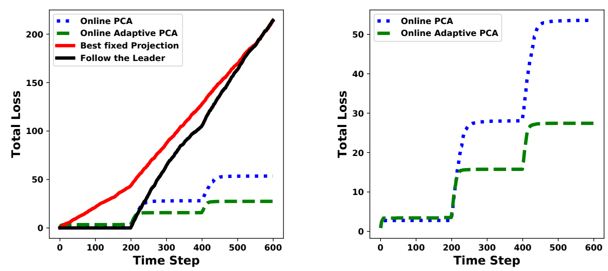
Algorithm 6 Online adaptive variance minimization over simplex

Input: an initial vector $\mathbf{y_1} \in \mathcal{B}_1^n$.
for $t = 1$ **to** T **do**
  Receive covariance matrix C_t.
  Incur the loss $\mathbf{y_t}^T C_t \mathbf{y_t}$.
  Update $\mathbf{y_{t+1}}$ as:

$$v_{t+1,i} = \frac{y_{t,i}\exp\big(-\eta(C_t\mathbf{y_t})_i\big)}{\sum_i y_{t,i}\exp\big(-\eta(C_t\mathbf{y_t})_i\big)}, \tag{31a}$$

$$y_{t+1,i} = \frac{\alpha}{n} + (1-\alpha)v_{t+1,i}. \tag{31b}$$

end for



(a) (b)

Figure 1. Fig.1(a): The cumulative loss of the toy example with data samples coming from three different subspaces. Fig.1(b): The detailed comparison for the two online algorithms.

As is done in the previous sections, we add the fixed-share step after the above update, which is summarized in Algorithm 6.

With the update of y_t in the Algorithm 6, we have the following theorem:

Theorem 6. *For any sequence of covariance matrices C_1, ..., C_T with $0 \preceq C_t \preceq I$, and for $\min_{\mathbf{u}\in\mathcal{B}_1^n} \sum_{t=r}^{s} \mathbf{u}^T C_t \mathbf{u} \leq L$, if we run Algorithm 6 with $\alpha = 1/(T+1)$, $c = \frac{\sqrt{2\ln\big((1+T)n\big)+2}}{\sqrt{L}}$, $b = \frac{c}{2}$, $a = \frac{b}{2b+1}$, and $\eta = 2a$, for any $T \geq 1$ we have:*

$$\mathcal{R}_a^{\textit{var-sim}} \leq 2\sqrt{2L\Big(\ln\big((1+T)n\big)+1\Big)} + 2\ln\big((1+T)n\big)$$

6. Experiments

In this section, we use two examples to illustrate the effectiveness of our proposed online adaptive PCA algorithm. The first example is synthetic, which shows that our proposed algorithm (denoted as Online Adaptive PCA) can adapt to the changing subspace faster than the method of (Warmuth & Kuzmin, 2008). The second example uses the

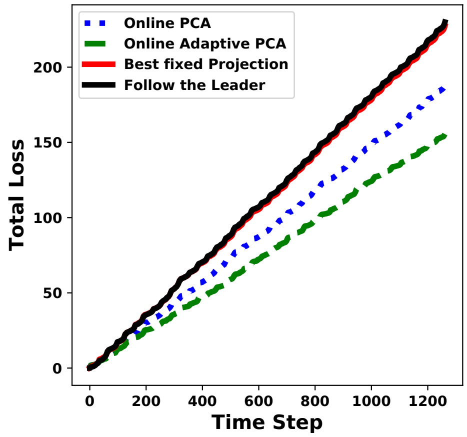


Figure 2. The cumulative loss for the face example with data samples coming from 20 different persons

practical dataset Yale-B to demonstrate that the proposed algorithm can have lower cumulative loss in practice when the data/face samples are coming from different persons.

The other algorithms that are used as comparators are: 1. Follow the Leader algorithm (denoted as Follow the Leader) (Kalai & Vempala, 2005), which only minimizes the loss on the past history; 2. The best fixed solution in hindsight (denoted as Best fixed Projection), which is the solution to the Problem described in Eq.(2); 3. The online static PCA (denoted as Online PCA) (Warmuth & Kuzmin, 2008). Other PCA algorithms are not included, since they are not designed for regret minimization.

6.1. A Toy Example

In this toy example, we create the synthetic data samples coming from changing subspace/environment, which is a similar setup as in (Warmuth & Kuzmin, 2008). The data samples are divided into three equal time intervals, and each interval has 200 data samples. The 200 data samples within same interval is randomly generated by a Gaussian distribution with zero mean and data dimension equal to 20, and the covariance matrix is randomly generated with rank equal to 2. In this way, the data samples are from some unknown 2-dimensional subspace, and any data sample with ℓ_2-norm greater than 1 is normalized to 1. Since the stepsize used in the two online algorithms is determined by the upper bound of the batch solution, we first find the upper bound and plug into the stepsize function, which gives $\eta = 0.19$. We can tune the stepsize heuristically in practice and in this example we just use $\eta = 1$ and $\alpha = 1\text{e}{-5}$.

After all data samples are generated, we apply the previously mentioned algorithms with $k = 2$ and obtain the cumulative loss as a function of time steps, which is shown in Fig.1. From this figure we can see that: 1. Follow the Leader algorithm is not appropriate in the setting where the sequential data is shifting over time. 2. The static regret is not a good metric under this setting, since the best fixed solution in hindsight is suboptimal. 3. Compared with Static PCA, the proposed Adaptive PCA can adapt to the changing environment faster, which results in lower cumulative loss and is more appropriate when the data is shifting over time.

6.2. Face data Compression Example

In this example, we use the Yale-B dataset which is a collection of face images. The data is split into 20 time intervals corresponding to 20 different people. Within each interval, there are 64 face image samples. Like the previous example, we first normalize the data to ensure its ℓ_2-norm not greater than 1. We use $k = 2$, which is the same as the previous example. The stepsize η is also tuned heuristically like the previous example, which is equal to 5 and $\alpha = 1\text{e}{-4}$.

We apply the previously mentioned algorithms and again obtain the cumulative loss as the function of time steps, which is displayed in Fig.2. From this figure we can see that although there is no clear bumps indicating the shift from one subspace to another as the Fig.1 of the toy example, our proposed algorithm still has the lowest cumulative loss, which indicates that upper bounding the adaptive regret is still effective when the compressed faces are coming from different persons.

7. Conclusion

In this paper, we propose an online adaptive PCA algorithm, which augments the previous online static PCA algorithm with a fixed-share step. However, different from the previous online PCA algorithm which is designed to minimize the static regret, the proposed online adaptive PCA algorithm aims to minimize the adaptive regret which is more appropriate when the underlying environment is changing or the sequential data is shifting over time. We demonstrate theoretically and experimentally that our algorithm can adapt to the changing environments. Furthermore, we extend the online adaptive PCA algorithm to online adaptive variance minimization problems.

One may note that the proposed algorithms suffer from the per-iteration computation complexity of $O(n^3)$ due to the eigendecomposition step, although some tricks mentioned in (Arora et al., 2012) could be used to make it comparable with incremental PCA of $O(k^2 n)$. For the future work, one possible direction is to investigate algorithms with slightly worse adaptive regret bound but with better per-iteration computation complexity.

References

Arora, R., Cotter, A., Livescu, K., and Srebro, N. Stochastic optimization for pca and pls. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pp. 861–868. IEEE, 2012.

Blum, A., Kumar, V., Rudra, A., and Wu, F. Online learning in online auctions. *Theoretical Computer Science*, 324(2-3):137–146, 2004.

Bregman, L. M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3): 200–217, 1967.

Censor, Y. and Lent, A. An iterative row-action method for interval convex programming. *Journal of Optimization theory and Applications*, 34(3):321–353, 1981.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Cesa-Bianchi, N., Gaillard, P., Lugosi, G., and Stoltz, G. Mirror descent meets fixed share (and feels no regret). In *Advances in Neural Information Processing Systems*, pp. 980–988, 2012a.

Cesa-Bianchi, N., Gaillard, P., Lugosi, G., and Stoltz, G. A new look at shifting regret. *arXiv preprint arXiv:1202.3323*, 2012b.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

Fazel, M., Ge, R., Kakade, S. M., and Mesbahi, M. Global convergence of policy gradient methods for linearized control problems. *arXiv preprint arXiv:1801.05039*, 2018.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1): 119–139, 1997.

Gao, X., Li, X., and Zhang, S. Online learning with nonconvex losses and non-stationary regret. In *International Conference on Artificial Intelligence and Statistics*, pp. 235–243, 2018.

Hazan, E. and Seshadhri, C. Efficient learning algorithms for changing environments. In *Proceedings of the 26th annual international conference on machine learning*, pp. 393–400. ACM, 2009.

Hazan, E., Singh, K., and Zhang, C. Efficient regret minimization in non-convex games. In *International Conference on Machine Learning*, pp. 1433–1441, 2017.

Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. Spectral filtering for general linear dynamical systems. *arXiv preprint arXiv:1802.03981*, 2018.

Herbster, M. and Warmuth, M. K. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

Herbster, M. and Warmuth, M. K. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1 (Sep):281–309, 2001.

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Markowitz, H. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

Nie, J., Kotlowski, W., and Warmuth, M. K. Online pca with optimal regret. *Journal of Machine Learning Research*, 17(173):1–49, 2016. URL http://jmlr.org/papers/v17/15-320.html.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Tsuda, K., Rätsch, G., and Warmuth, M. K. Matrix exponentiated gradient updates for on-line learning and bregman projection. *Journal of Machine Learning Research*, 6(Jun):995–1018, 2005.

Warmuth, M. K. and Kuzmin, D. Online variance minimization. In *International Conference on Computational Learning Theory*, pp. 514–528. Springer, 2006.

Warmuth, M. K. and Kuzmin, D. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9 (Oct):2287–2320, 2008.

Yuan, J. and Lamperski, A. Online control basis selection by a regularized actor critic algorithm. In *2017 American Control Conference (ACC)*, pp. 4448–4453. IEEE, 2017.

Yuan, J. and Lamperski, A. Online convex optimization for cumulative constraints. In *Advances in Neural Information Processing Systems*, pp. 6137–6146, 2018.

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, 2003.