On the equivalence of Oja's algorithm and GROUSE

Laura Balzano
girasole@umich.edu
University of Michigan

Abstract

The analysis of streaming PCA has gained significant traction through the analysis of an early simple variant: Oja's algorithm, which implements online projected gradient descent for the trace objective. Several other streaming PCA algorithms have been developed, each with their own performance guarantees or empirical studies, and the question arises whether there is a relationship between the algorithms. We show that the Grassmannian Rank-One Subspace Estimation (GROUSE) algorithm is indeed equivalent to Oja's algorithm in the sense that, at each iteration, given a step size for one of the algorithms, we may construct a step size for the other algorithm that results in an identical update. This allows us to apply all results on one algorithm to the other. In particular, we have (1) better global convergence guarantees of GROUSE to the global minimizer of the PCA objective with full data; and (2) local convergence guarantees for Oja's algorithm with incomplete or compressed data.

1 INTRODUCTION

While the field of optimization is very established, with well-known algorithms for solving general problems, researchers are constantly "discovering" new algorithmic approaches. There is probably no one problem where this is more true than in Streaming PCA, where researchers many times over have developed new methodologies to solve the problem. An algorithm developed in 1982 by Oja (1982) applies projected gradient descent with rank-one gradient updates. This algorithm is probably the most studied (Chen et al., 1998; Yi et al., 2005; Jain et al., 2016; Allen-Zhu and Li, 2017; Henriksen and Ward, 2019; Huang et al., 2021), and it has been proved to converge under somewhat general conditions. It also resembles many other methods in the literature, as can be seen in the following survey papers (Comon and Golub, 1990; Balzano et al., 2018).

The Grassmannian Rank-One Subspace Estimation (GROUSE) (Balzano et al., 2010) algorithm is a stochastic manifold optimization approach to streaming PCA, which would not in general be equivalent to a projected gradient algorithm like Oja's algorithm. The contribution of this paper is to show that these algorithms are indeed equivalent, in the sense that, fixing an initialization and step size regimen for one, there exists a step size regimen for the other that, with the same initialization, will give identical output at every iteration of the algorithm. Additionally, our analysis highlights a minor but key difference between the two algorithms in their treatment of orthonormality during the gradient calculation. These differences suggest yet another projected gradient algorithm, which was discussed by Tang (2019) and which we show is also equivalent to GROUSE and Oja's algorithm. These results then allow us to use theory from one algorithm applied to the other. In this paper we write down two results that arise because of the equivalence: local convergence guarantees of a variant of Oja's algorithm for incompletely observed vectors, and global convergence guarantees for GROUSE.

2 PROBLEM FORMULATION

The Principal Component Analysis (PCA) problem[1] in batch is posed in the following two ways. Suppose we have n data vectors $\{\mathbf{x}_t\}_{t=1}^n \subset \mathbb{R}^d$ and we model them as random, zero mean, with a shared covariance matrix $\mathbf{\Sigma}$. To learn the top eigenvectors of that covariance, so that we can project the data onto its highest

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[1] Here we have posed the problem of learning the dominant k-dimensional subspace for fixed k. PCA, more generally, also identifies the relative importance of each principal component via the eigenvalues.

variance subspace, we will solve for the subspace $\mathbf{U}$ that maximizes $\mathrm{Tr}\left(\mathbf{U}^T\boldsymbol{\Sigma}\mathbf{U}\right)$, which in finite sample is approximated by

$$\underset{\mathbf{U}\in\mathbb{R}^{d\times k},\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\text{maximize}} \ \mathrm{Tr}\left(\mathbf{U}^T\hat{\boldsymbol{\Sigma}}\mathbf{U}\right) \ , \tag{1}$$

where $\hat{\boldsymbol{\Sigma}} = \sum_{t=1}^n \mathbf{x}_t\mathbf{x}_t^T$ is the sample covariance matrix. We also reach the same objective when we suppose the data are deterministic, and we want to find a k-dimensional subspace on which we can project the data and preserve as much of the norm as possible. Then we may let $\mathbf{X} = \begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n\end{bmatrix}$ and solve:

$$\underset{\mathbf{U}\in\mathbb{R}^{d\times k},\mathbf{U}^T\mathbf{U}=\mathbf{I}}{\text{minimize}} \ \left\|\mathbf{U}\mathbf{U}^T\mathbf{X}-\mathbf{X}\right\|_F^2 \tag{2}$$

The two objective functions in (1) and (2) are equivalent[2]. Both of these problems can be written as a sum of functions, each of which depends on only one data point $\mathbf{x}_t$:

$$\sum_{t=1}^n F_t^{(\mathrm{Trace})}(\mathbf{U}) \coloneqq \sum_{t=1}^n \mathrm{Tr}(\mathbf{U}^T\mathbf{x}_t\mathbf{x}_t^T\mathbf{U}) \ , \tag{3}$$

and in the matrix approximation setting using the decomposition of the Frobenius norm:

$$\sum_{t=1}^n F_t^{(\mathrm{Frob})}(\mathbf{U}) \coloneqq \sum_{t=1}^n \left\|\mathbf{U}\mathbf{U}^T\mathbf{x}_t-\mathbf{x}_t\right\|_2^2 \ . \tag{4}$$

It is therefore natural to think about optimizing these objectives in the streaming setting using stochastic or incremental gradient descent (Bertsekas, 2011). In fact we know from a great deal of work that for this objective, though it is non-convex, all local minima are global minima (See our discussion in Section 2.1; we believe the earliest such result is by Yang (1995)), and so gradient descent has a chance to converge to a global minimizer. This is the approach taken by Oja's algorithm and the GROUSE algorithm.

The Euclidean gradient for the trace objective is read easily from Eq (3) as

$$\nabla_{\mathbf{U}} F_t^{(\mathrm{Trace})} = \mathbf{x}_t\mathbf{x}_t^T\mathbf{U} \ ,$$

which we can also write as $\mathbf{x}_t\mathbf{w}_t^T$ where since $\mathbf{U}$ is assumed to have orthonormal columns, $\mathbf{U}^T\mathbf{x}_t = \mathbf{w}_t$ are the weights of the projection of $\mathbf{x}_t$ onto the span of $\mathbf{U}$, *i.e.*,

$$\mathbf{w}_t = \arg\min_{\mathbf{w}} \|\mathbf{x}_t - \mathbf{U}\mathbf{w}\|_2^2 \ .$$

[2]This equivalence relies on the constraint that the columns of $\mathbf{U}$ are orthonormal. If $\mathbf{U}$ is unconstrained, then the objective in (1) is unbounded, and the two are not equivalent. It's an open question as to what other constraints might guarantee that the problems are equivalent.

The Euclidean gradient for the Frobenius norm objective is the same if we impose $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, but if we take the gradient in Euclidean space before imposing this constraint we get a different outcome:

$$\nabla_{\mathbf{U}} F_t^{(\mathrm{Frob})} = (\mathbf{U}\mathbf{U}^T\mathbf{x}_t - \mathbf{x}_t)\mathbf{x}_t^T\mathbf{U} = -2\mathbf{r}_t\mathbf{w}_t^T \ ,$$

where $\mathbf{r}_t$ is the project residual, *i.e.*, $\mathbf{r}_t = \mathbf{x}_t - \mathbf{U}\mathbf{w}_t$. The relationship of the two gradients can be seen as follows – the second is a restriction of the first to the tangent space of the Grassmannian $\mathcal{G}(d,k)$, the space of all subspaces of dimension k in $\mathbb{R}^d$ (Edelman et al., 1998, Sec 2.5.3). See more discussion in Appendix A.

2.1 Related Work

Subspace tracking, i.e., the problem of incrementally updating the principal subspace of streaming data, has been a problem of interest in a wide range of application for several decades. New algorithms are regularly invented. Oja's (Oja, 1982; Oja and Karhunen, 1985) and Krasulina's (Krasulina, 1969; Karasalo, 1986) algorithms are the oldest to the best of our knowledge, and there was a flurry of activity in the 1980-90s (Yang and Kaveh, 1988; Smith, 1993; Yang, 1995; Mathew et al., 1995; Gustafsson, 1998; Hua et al., 1999; Real et al., 1999; Bischof and Shroff, 1992; Moonen et al., 1992; Stewart, 1992), with continued attention through the modern era where subspace tracking is applied in the context of massive data, missing, or corrupted / adversarially perturbed data (Attallah and Abed-Meraim, 2001; Chatterjee, 2005; Badeau et al., 2005; Chan et al., 2005; Brand, 2006; Warmuth and Kuzmin, 2008; Doukopoulos and Moustakides, 2008; Strobach, 2009; Balzano et al., 2010; Chi et al., 2013; Arora et al., 2013; Hardt and Price, 2014; De Sa et al., 2015; Nie et al., 2016; Jain et al., 2016; Ghashami et al., 2016; Zhan et al., 2016; Chan et al., 2018; Yang et al., 2018; Javed et al., 2018; Tripuraneni et al., 2018; Kotłowski and Neu, 2019; Garber, 2019; Tang, 2019). Surveys are found in (Comon and Golub, 1990; Balzano et al., 2018). Techniques from optimization such as variance reduction have also been applied to improve algorithms and convergence guarantees (Shamir, 2016; Xu et al., 2018; Arora and Marinov, 2019). Streaming PCA or subspace estimation can be formulated with updates based on a single vector at a time or a block of vectors; in this work we focus on the single vector case.

Oja's algorithm has long been of theoretical interest, with results spanning decades proving convergence of versions of the algorithm (Chen et al., 1998; Balsubramani et al., 2013; Jain et al., 2016; Allen-Zhu and Li, 2017; Li et al., 2016; Henriksen and Ward, 2019; Amid and Warmuth, 2020; Gemp et al., 2020; Lunde et al., 2021; Huang et al., 2021; Liang, 2021). Generally for

Algorithm 1 Oja's algorithm

1: Given $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$;
2: Given step size scheme $\eta_t > 0$;
3: Set $t := 0$;
4: **repeat**
5: Define $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{x}_t - \mathbf{U}_t\mathbf{w}\|_2^2 = \mathbf{U}_t^T\mathbf{x}_t$;
6: Update:

$$\widehat{\mathbf{U}}_{t+1} = \mathbf{U}_t + \eta_t \mathbf{x}_t \mathbf{w}_t^T \tag{5}$$

$$\mathbf{U}_{t+1} = \Pi(\widehat{\mathbf{U}}_{t+1}) \tag{6}$$

7: $t := t + 1$;
8: **until** termination

Algorithm 2 GROUSE (Balzano et al., 2010) (with fully observed data)

1: Given $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$;
2: Given step size scheme $\theta_t > 0$;
3: Set $t := 0$;
4: **repeat**
5: Define $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{x}_t - \mathbf{U}_t\mathbf{w}\|_2^2 = \mathbf{U}_t^T\mathbf{x}_t$;
6: Define $\mathbf{p}_t := \mathbf{U}_t\mathbf{w}_t = \mathbf{U}_t\mathbf{U}_t^T\mathbf{x}_t$ and $\mathbf{r}_t = \mathbf{x}_t - \mathbf{p}_t = (\mathbf{I} - \mathbf{U}_t\mathbf{U}_t^T)\mathbf{x}_t$.
7: Update:

$$\mathbf{U}_{t+1} = \mathbf{U}_t + (\cos(\theta_t\|\mathbf{r}_t\|\|\mathbf{p}_t\|) - 1)\frac{\mathbf{p}_t}{\|\mathbf{p}_t\|}\frac{\mathbf{w}_t^T}{\|\mathbf{w}_t\|} + \sin(\theta_t\|\mathbf{r}_t\|\|\mathbf{p}_t\|)\frac{\mathbf{r}_t}{\|\mathbf{r}_t\|}\frac{\mathbf{w}_t^T}{\|\mathbf{w}_t\|} \tag{7}$$

8: $t := t + 1$;
9: **until** termination

this analysis, it is assumed that the stream of data arises i.i.d., zero mean with covariance $\mathbf{\Sigma}$, with the goal of estimating the principal components of $\mathbf{\Sigma}$. One reason that Oja's algorithm is so amenable to analysis is that its update can be written using only the independent data stream $\mathbf{x}_t$. We will discuss this in more detail in Section 4.

With such a wide variety of subspace tracking algorithms in the literature, it has been of great interest to understand the relationships among them. The work by Wang et al. (2018) carefully studied Oja's algorithm as well as two other recent variants, GROUSE (Balzano et al., 2010) and PETRELS (Chi et al., 2012), with incomplete observations. Given input data drawn from a stochastic process, each of these algorithms has another stochastic process as its output. By making several assumptions on the data-generating process, including that the data are drawn from a low-rank subspace with i.i.d. coefficients and i.i.d. additive noise, Wang et al. (2018) identify the deterministic function that exactly characterizes the stochastic processes in the high-dimensional limit as dimension $d \to \infty$. They make the observation in (Wang et al., 2018, Thm 1) that this deterministic function is identical for Oja's algorithm and GROUSE. In contrast, but also supporting this observation, our work shows that the outputs of these algorithms are identical at every step. Our result holds for finite dimension regardless of whether the data are from a low-rank model or from any i.i.d. random process, and regardless of whether the algorithms even converge. In that sense ours is a much more general equivalence result than the result of Wang et al. (2018).

Next we discuss work examining the landscape of the PCA problem. The following important result was proven by Yang (1995): Assuming the data are randomly drawn with zero mean and covariance $\mathbf{\Sigma}$, $\mathbf{U}$ is a stationary point of $\mathbb{E}[F^{(\text{Frob})}]$ if and only if it is a matrix with orthonormal columns, and with a column space spanned by k eigenvectors of the sample covariance matrix $\mathbf{\Sigma}$ (equiv. for $\mathbb{E}[F^{\text{Trace}}]$ assuming the orthonormal constraint). Moreover, if $\lambda_k(\mathbf{\Sigma}) > \lambda_{k+1}(\mathbf{\Sigma})$, i.e., there is a strict eigengap, then all stationary points are strict saddle points (with at least one direction of negative curvature) except the global optimum, where $\mathbf{U}$ contains the top-k eigenvectors of $\mathbf{\Sigma}$ up to an orthonormal transformation. Several recent works have repeated these results and extended them to more modern settings: where we have finite samples, data are observed with additive noise, or data are observed through underdetermined linear measurements ("matrix completion" or "matrix sensing") (Li et al., 2019; Ge et al., 2017; Zhu et al., 2021). Landscape results are known also for robust subspace recovery, despite being both nonconvex and nonsmooth; locally around the global optima, the landscape is favorable for gradient methods (Maunu et al., 2019), and it is possible to initialize to that local region using PCA.

Recent work on algorithmic equivalence (Zhao et al., 2021) has highlighted relationships between independently derived algorithms. Many of these relationships have long been understood in the optimization community, but were often only pointed out as ancillary to the main contribution. Their approach does not directly apply to the algorithms we consider, though it inspires more careful general consideration as future work. For equivalence of PCA algorithms, Tripuraneni et al. (2018) has argued that generic Riemannian SGD is equivalent to Oja's algorithm up to a correc-

Algorithm 3 PGF – projected gradient descent on the Frobenius norm objective

1: Given $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$;
2: Given step size scheme $\gamma_t > 0$;
3: Set $t := 0$;
4: **repeat**
5: Define $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{x}_t - \mathbf{U}_t\mathbf{w}\|_2^2 = \mathbf{U}_t^T\mathbf{x}_t$;
6: Define $\mathbf{r}_t = \mathbf{x}_t - \mathbf{U}_t\mathbf{w}_t$.
7: Update:

$$\widehat{\mathbf{U}}_{t+1} = \mathbf{U}_t + \gamma_t \mathbf{r}_t \mathbf{w}_t^T \tag{8}$$

$$\mathbf{U}_{t+1} = \Pi(\widehat{\mathbf{U}}_{t+1}) \tag{9}$$

8: $t := t + 1$;
9: **until** termination

tion, and the GROUSE algorithm has been proven to be equivalent to a form of the truncated Incremental SVD (Bunch and Nielsen, 1978; Balzano and Wright, 2013). In both cases this is without the same flexibility as in our work to port results in both directions. Still, the potential of identifying more equivalencies, and especially of connecting gradient methods to linear algebraic methods in this area, is very intriguing.

3 ALGORITHMS AND RESULT

The three algorithms we analyze are Oja's algorithm (Oja, 1982) given in Algorithm 1, the GROUSE algorithm (Balzano et al., 2010) given in Algorithm 2, and projected gradient descent on the Frobenius norm objective given in Algorithm 3, which we abbreviate PGF. The notation $\Pi(\mathbf{U})$ represents any function that outputs an orthonormalization of the columns of $\mathbf{U}$, e.g., Gram-Schmidt.

Before discussing the algorithmic equivalence we comment on the computational complexity. Step 5 is shared by all three algorithms and requires $O(dk)$ operations when $\mathbf{U}$ has orthonormal columns. The update steps also require $O(dk)$ except for the orthonormalization, which requires $O(dk^2)$. Given this, it seems that GROUSE Algorithm 2 is best computationally, since it doesn't require orthonormalization. However, there are two caveats to this argument: First, often with Oja's algorithm (Algorithm 1) or PGF (Algorithm 3), one doesn't orthonormalize at every step, but only periodically to keep the estimate from becoming ill-conditioned. Second, if we are dealing with missing data and looking only at a subset of rows of $\mathbf{U}$ (or generally if we don't guarantee $\mathbf{U}$ is orthonormal at each step), the least squares computation in Step 5 now requires $O(dk^2)$ operations (or $O(mk^2)$ operations where m is the dimension of the observation of $\mathbf{x}_t$). Therefore, in these practical scenarios, the algorithms' computational complexities are very similar.

We will now discuss the equivalence of all three algorithms. As seen in Appendix A, these algorithms have only minor differences in their gradient. This then manifests in the algorithm updates, from which we can identify a clear geometric reason why the algorithms are equivalent. Since the gradient update for these incremental algorithms is rank-one, only one direction of the current subspace iterate $\mathbf{U}_t$ will change with an update step. In all three algorithms, that direction is updated to be a linear combination of the projection $\mathbf{p}_t = \mathbf{U}_t\mathbf{w}_t$ of $\mathbf{x}_t$ onto the current subspace and the projection residual $\mathbf{r}_t = \mathbf{x}_t - \mathbf{p}_t$. The step-size is what dictates the linear combination, and we can tweak the step sizes of each algorithm so that the updates match exactly.

We can make this precise as follows. Suppose that all $\mathbf{U}_t$ output by the algorithms at each iteration have orthonormal columns. Let $\mathbf{Z}$ be a $k \times k$ orthogonal matrix, depending on $\mathbf{U}_t$ and $\mathbf{x}_t$:

$$\mathbf{Z} = \begin{bmatrix} \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|} & \mathbf{z}_2 & \cdots & \mathbf{z}_k \end{bmatrix}, \tag{10}$$

where $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{x}_t - \mathbf{U}_t\mathbf{w}\|_2^2 = \mathbf{U}_t^T\mathbf{x}_t$ and $\mathbf{z}_2, \ldots, \mathbf{z}_k \in \mathbb{R}^k$ are orthogonal to each other and to $\mathbf{w}_t$. This matrix is such that

$$\mathbf{U}_t\mathbf{Z} = \begin{bmatrix} \frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k \end{bmatrix},$$

where $\mathbf{v}_i = \mathbf{U}_t\mathbf{z}_i$ for $i = 2, \ldots, k$. Note that since $\mathbf{U}_t$ has orthonormal columns and $\mathbf{Z}$ is a square orthogonal matrix, $\text{span}(\mathbf{U}_t) = \text{span}(\mathbf{U}_t\mathbf{Z})$ and the columns of $\mathbf{U}_t\mathbf{Z}$ are orthonormal.

All three algorithms have an update of the form $\mathbf{U}_t + \mathbf{a}_t\mathbf{w}_t^T$. If we multiply this general form on the right by $\mathbf{Z}$ we get

$$\begin{aligned} \mathbf{U}_{t+1}\mathbf{Z} &= \mathbf{U}_t\mathbf{Z} + \mathbf{a}_t\mathbf{w}_t^T\mathbf{Z} \\ &= \begin{bmatrix} \frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k \end{bmatrix} + \begin{bmatrix} \|\mathbf{w}_t\|\mathbf{a}_t & 0 & \cdots & 0 \end{bmatrix}. \end{aligned}$$

So we see in this change of coordinates, only one column of $\mathbf{U}_t$ is being updated, and it will be a linear combination of $\mathbf{U}_t\mathbf{w}_t/\|\mathbf{w}_t\| = \mathbf{p}_t/\|\mathbf{w}_t\|$ and $\mathbf{a}_t$, which in all three algorithms is a linear combination of $\mathbf{p}_t$ and $\mathbf{r}_t$.

The geometric picture for all three updates is shown in Figure 1. Since $\mathbf{x}_t$ is orthogonal to all directions in $\mathbf{U}_t$ other than $\mathbf{p}_t = \mathbf{U}_t\mathbf{w}_t$ by definition of projection, we see that the update for all three algorithms takes place entirely in a plane spanned by $\mathbf{p}_t$ and $\mathbf{r}_t$. In other

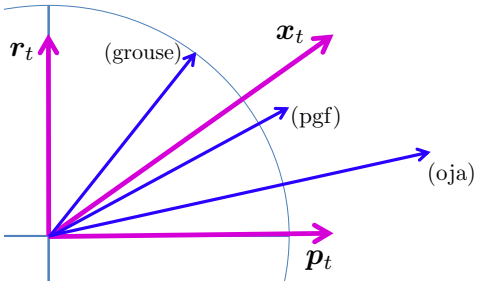

Figure 1: A cartoon of the three algorithms' updates, before projection onto the Grassmannian in the case of Oja's algorithm and PGF, with non-equivalent step sizes. In pink, the data $\mathbf{x}_t$, the projection onto the current subspace iterate $\mathbf{p}_t = \mathbf{U}_t\mathbf{w}_t$, and the projection residual $\mathbf{r}_t = \mathbf{x}_t - \mathbf{p}_t$ all lie in the same 2d plane. The light blue curve illustrates the unit sphere in that plane. All three algorithm updates lie in this plane. The difference between algorithm updates lies only in the choice of linear combination of these vectors, dictated by the choice of step size.

words, the update, i.e. the new direction that replaces $\mathbf{U}_t\mathbf{w}_t/\|\mathbf{w}_t\|$ in the subspace estimate, is a norm-1 vector in this plane. The update for Oja's algorithm is:

$$(1+\eta_t\|\mathbf{w}_t\|^2)\frac{\mathbf{p}_t}{\|\mathbf{w}_t\|} + \eta_t\|\mathbf{w}_t\|\mathbf{r}_t \ . \tag{11}$$

The update for the GROUSE algorithm is:

$$\cos(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)\frac{\mathbf{p}_t}{\|\mathbf{w}_t\|} + \sin(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)\frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} \ . \tag{12}$$

And the update for PGF is:

$$\frac{\mathbf{p}_t}{\|\mathbf{w}_t\|} + \gamma_t\|\mathbf{w}_t\|\mathbf{r}_t \ . \tag{13}$$

The careful derivation for each of these updates is found in the proof of the theorem.

Theorem 1. *Fix a step size scheme for Oja's algorithm, η_t, and an initialization $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$. Let $\mathbf{w}_t \in \mathbb{R}^{k\times 1}$ be the weights of the projection of data $\mathbf{x}_t$ onto the subspace $\mathbf{U}_t$, and let $\mathbf{r}_t = \mathbf{x}_t - \mathbf{U}_t\mathbf{w}_t$. Then if one performs the GROUSE algorithm with step size*

$$\theta_t = \frac{1}{\|\mathbf{r}_t\|\|\mathbf{w}_t\|}\arctan\left(\frac{\eta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|}{1+\eta_t\|\mathbf{w}_t\|^2}\right) \ , \tag{14}$$

or projected gradient descent on the Frobenius norm objective with step size

$$\gamma_t = \frac{\eta_t}{1+\eta_t\|\mathbf{w}_t\|^2} \ , \tag{15}$$

the iterates of the three algorithms will be identical for all t in the sense that

$$\text{span}(\mathbf{U}_t^{\text{oja}}) = \text{span}(\mathbf{U}_t^{\text{grouse}}) = \text{span}(\mathbf{U}_t^{\text{pgf}}) \ . \tag{16}$$

Proof. We assume the conclusion is true for t and prove the spans are the same for $t+1$, since all algorithms are initialized with the same $\mathbf{U}_0$. Let $\mathbf{Z}$ be as in Eq (10). Consider first the update of Oja's algorithm in Eq (5), rotated by $\mathbf{Z}$. Here we emphasize that applying this rotation does not change the span ($\text{span}(\mathbf{U}) = \text{span}(\mathbf{UZ})$), and our goal is simply to prove the span of each iterate is identical in Eq (16).

$$\begin{aligned}\widehat{\mathbf{U}}_{t+1}\mathbf{Z} &= \mathbf{U}_t\mathbf{Z} + \eta_t\mathbf{x}_t\mathbf{w}_t^T\mathbf{Z} \\ &= \begin{bmatrix}\frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix} + \begin{bmatrix}\eta_t\|\mathbf{w}_t\|\mathbf{x}_t & 0\cdots 0\end{bmatrix} \\ &= \begin{bmatrix}\frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} + \eta_t\|\mathbf{w}_t\|\mathbf{x}_t & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix} \ .\end{aligned}$$

Now consider the orthogonalization in Eq (6). Since $\mathbf{U}_t\mathbf{w}_t$ is the orthogonal projection of $\mathbf{x}_t$ onto $\mathbf{U}_t$, by the orthogonality principle, $\mathbf{x}_t$ is orthogonal to $\mathbf{v}_2, \ldots, \mathbf{v}_k$. Therefore $\frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} + \eta_t\|\mathbf{w}_t\|\mathbf{x}_t$ is orthogonal to the $\mathbf{v}_i$. To orthonormalize the columns, we must only normalize $\frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} + \eta_t\|\mathbf{w}_t\|\mathbf{x}_t$:

$$\begin{aligned}\mathbf{U}_{t+1}^{\text{oja}} &= \Pi\left(\widehat{\mathbf{U}}_{t+1}\mathbf{Z}\right) \\ &= \begin{bmatrix}\frac{\frac{1}{\|\mathbf{w}_t\|}\mathbf{U}_t\mathbf{w}_t + \eta_t\|\mathbf{w}_t\|\mathbf{x}_t}{\left\|\frac{1}{\|\mathbf{w}_t\|}\mathbf{U}_t\mathbf{w}_t + \eta_t\|\mathbf{w}_t\|\mathbf{x}_t\right\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix} \ .\end{aligned} \tag{17}$$

We will now use the same manipulation for the GROUSE update. First consider the following form of the update from Eq (7).

$$\mathbf{U}_{t+1}^{\text{grouse}} = \mathbf{U}_t - \frac{\mathbf{p}_t}{\|\mathbf{p}_t\|}\frac{\mathbf{w}_t^T}{\|\mathbf{w}_t\|} + \frac{\mathbf{y}_t}{\|\mathbf{y}_t\|}\frac{\mathbf{w}_t^T}{\|\mathbf{w}_t\|}$$

where $\mathbf{y}_t = \cos(\theta_t\|\mathbf{r}_t\|\|\mathbf{p}_t\|)\frac{\mathbf{p}_t}{\|\mathbf{p}_t\|} + \sin(\theta_t\|\mathbf{r}_t\|\|\mathbf{p}_t\|)\frac{\mathbf{r}_t}{\|\mathbf{r}_t\|}$ (note that $\|\mathbf{y}_t\| = 1$ as defined). Rotating on the right by $\mathbf{Z}$, and noting $\mathbf{p}_t = \mathbf{U}_t\mathbf{w}_t$, we have that

$$\mathbf{U}_{t+1}^{\text{grouse}}\mathbf{Z} = \begin{bmatrix}\frac{\mathbf{y}_t}{\|\mathbf{y}_t\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix} \ .$$

Calculations show that $\mathbf{U}_{t+1}^{\text{grouse}}\mathbf{Z}$ is already a matrix with orthonormal columns (as expected, since GROUSE is derived with geodesics on the Grassmannian).

Finally, the same manipulation can be applied to PGF. Take first Eq (8):

$$\begin{aligned}\widehat{\mathbf{U}}_{t+1}\mathbf{Z} &= \mathbf{U}_t\mathbf{Z} + \gamma_t\mathbf{r}_t\mathbf{w}_t^T\mathbf{Z} \\ &= \begin{bmatrix}\frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix} + \begin{bmatrix}\gamma_t\|\mathbf{w}_t\|\mathbf{r}_t & 0\cdots 0\end{bmatrix} \\ &= \begin{bmatrix}\frac{\mathbf{U}_t\mathbf{w}_t}{\|\mathbf{w}_t\|} + \gamma_t\|\mathbf{w}_t\|\mathbf{r}_t & \mathbf{v}_2 & \cdots & \mathbf{v}_k\end{bmatrix} \ .\end{aligned}$$

As in Oja's orthonormalization step in Eq (17),

$$\begin{aligned}\mathbf{U}_{t+1}^{\text{pgf}} &= \Pi(\widehat{\mathbf{U}}_{t+1}\mathbf{Z}) \qquad (18)\\ &= \begin{bmatrix} \frac{\frac{1}{\|\mathbf{w}_t\|}\mathbf{U}_t\mathbf{w}_t+\gamma_t\|\mathbf{w}_t\|\mathbf{r}_t}{\left\|\frac{1}{\|\mathbf{w}_t\|}\mathbf{U}_t\mathbf{w}_t+\gamma_t\|\mathbf{w}_t\|\mathbf{r}_t\right\|} & \mathbf{v}_2 & \cdots & \mathbf{v}_k \end{bmatrix}.\end{aligned}$$

It remains only to compute the step size to make these three vectors equal. We drop the subscript t for the rest of the proof.

$$\frac{\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|\mathbf{x}}{\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|\mathbf{x}\right\|} \qquad \text{(oja)}$$

$$\cos(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{p}}{\|\mathbf{p}\|}+\sin(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{r}}{\|\mathbf{r}\|} \qquad \text{(grouse)}$$

$$\frac{\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\gamma\|\mathbf{w}\|\mathbf{r}}{\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\gamma\|\mathbf{w}\|\mathbf{r}\right\|} \qquad \text{(pgf)}$$

These three vectors are all linear combinations of $\mathbf{p} = \mathbf{U}\mathbf{w}$ and $\mathbf{r}$, and they are all on the unit circle in the plane spanned by $\mathbf{p}$ and $\mathbf{r}$. So to make them equal, we can compute the step size so that their angle with $\mathbf{U}\mathbf{w}$ is all the same. We start by computing the angle between $\mathbf{U}\mathbf{w}$ and Oja's update:

$$\begin{aligned}&\arccos\left(\frac{\left\langle \mathbf{U}\mathbf{w}, \mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|\mathbf{x}\right\rangle}{\|\mathbf{w}\|\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|\mathbf{x}\right\|}\right)\\ =&\arccos\left(\frac{\mathbf{w}^T\mathbf{U}^T\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|\mathbf{w}^T\mathbf{U}^T(\mathbf{U}\mathbf{w}+\mathbf{r})}{\|\mathbf{w}\|\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|(\mathbf{U}\mathbf{w}+\mathbf{r})\right\|}\right)\\ =&\arccos\left(\frac{1+\eta\|\mathbf{w}\|^2}{\left\|(1+\eta\|\mathbf{w}\|^2)\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\eta\|\mathbf{w}\|\mathbf{r}\right\|}\right)\\ =&\arccos\left(\frac{1}{\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\frac{\eta\|\mathbf{w}\|}{1+\eta\|\mathbf{w}\|^2}\mathbf{r}\right\|}\right)\end{aligned}$$

We will put the other two updates' angles into the same form. For GROUSE[3]:

$$\begin{aligned}&\arccos\left(\frac{\left\langle \mathbf{U}\mathbf{w}, \cos(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{p}}{\|\mathbf{p}\|}+\sin(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{r}}{\|\mathbf{r}\|}\right\rangle}{\|\mathbf{w}\|\left\|\cos(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{p}}{\|\mathbf{p}\|}+\sin(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{r}}{\|\mathbf{r}\|}\right\|}\right)\\ =&\arccos\left(\frac{\cos(\theta\|\mathbf{r}\|\|\mathbf{w}\|)}{\left\|\cos(\theta\|\mathbf{r}\|\|\mathbf{w}\|)\frac{\mathbf{U}\mathbf{w}}{\|\mathbf{w}\|}+\sin(\theta\|\mathbf{r}\|\|\mathbf{w}\|)\frac{\mathbf{r}}{\|\mathbf{r}\|}\right\|}\right)\\ =&\arccos\left(\frac{1}{\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\frac{\sin(\theta\|\mathbf{r}\|\|\mathbf{w}\|)}{\|\mathbf{r}\|\cos(\theta\|\mathbf{r}\|\|\mathbf{w}\|)}\mathbf{r}\right\|}\right)\end{aligned}$$

[3]We note that the update $\mathbf{y} = \cos(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{p}}{\|\mathbf{p}\|}\frac{\mathbf{w}^T}{\|\mathbf{w}\|} + \sin(\theta\|\mathbf{r}\|\|\mathbf{p}\|)\frac{\mathbf{r}}{\|\mathbf{r}\|}$ is already norm-one, but we use its norm in the denominator to match the form for the other two algorithms.

For PGF:

$$\begin{aligned}&\arccos\left(\frac{\left\langle \mathbf{U}\mathbf{w}, \frac{1}{\|\mathbf{w}\|}\mathbf{U}\mathbf{w}+\gamma\|\mathbf{w}\|\mathbf{r}\right\rangle}{\|\mathbf{w}\|\left\|\frac{1}{\|\mathbf{w}\|}\mathbf{U}\mathbf{w}+\gamma\|\mathbf{w}\|\mathbf{r}\right\|}\right)\\ =&\arccos\left(\frac{1}{\left\|\mathbf{U}\frac{\mathbf{w}}{\|\mathbf{w}\|}+\gamma\|\mathbf{w}\|\mathbf{r}\right\|}\right)\end{aligned}$$

We may conclude that we must have

$$\frac{\eta\|\mathbf{w}\|}{1+\eta\|\mathbf{w}\|^2} = \frac{\sin(\theta\|\mathbf{r}\|\|\mathbf{w}\|)}{\|\mathbf{r}\|\cos(\theta\|\mathbf{r}\|\|\mathbf{w}\|)} = \gamma\|\mathbf{w}\|\,. \qquad (19)$$

For the theorem's conclusion, we fix η and have

$$\theta = \frac{1}{\|\mathbf{r}\|\|\mathbf{w}\|}\arctan\left(\frac{\eta\|\mathbf{r}\|\|\mathbf{w}\|}{1+\eta\|\mathbf{w}\|^2}\right)$$

and

$$\gamma = \frac{\eta}{1+\eta\|\mathbf{w}\|^2}\,.$$

□

4 CONSEQUENCES OF EQUIVALENCE

Now that we have shown that with appropriate adjustments of step sizes Oja, GROUSE, and PGF are equivalent, any existing results on the methods can be ported to the others. While this is direct, we want to write two results here so as to help the reader consolidate their understanding of the algorithms and bring everything into common notation.

We first comment that Oja's algorithm has a benefit for analysis that the update step can be written as

$$\mathbf{U}_{t+1} = (\mathbf{I}+\eta_t\mathbf{x}_t\mathbf{x}_t^T)\mathbf{U}_t\,,$$

i.e., as the product of a random i.i.d matrix with the current iterate. This is actually helpful across iterations too, because as shown by (Allen-Zhu and Li, 2017, Lemma 2.2), for analysis purposes only, one can wait to do the orthonormalization step of (6) at the end. This was leveraged by Huang et al. (2021), where novel matrix concentration for random matrix products was applied. GROUSE and PGF, on the other hand, use $\mathbf{r}_t$ in their gradient, which depends itself on $\mathbf{U}_t$, making every update dependent on the previous ones. That is in part because GROUSE was designed for missing data or compressively sampled data, a context in which this dependence provides a way to interpolate the full-dimensional vector using the current subspace estimate. There are more limited but interesting results for the GROUSE algorithm (Balzano and Wright, 2015; Zhang and Balzano, 2022,

2016). We hope that our equivalence result will allow novel analysis of Oja's algorithm with missing or compressively sampled data, which has been previously discussed in (Balzano et al., 2018; Wang et al., 2018).

4.1 Global convergence results

First, we can use the global convergence results of Oja's algorithm to prove the convergence of the other two methods. In the following we restate the result from Allen-Zhu and Li (2017), which proves order optimal convergence of Oja's algorithm to the top eigenspace of the population covariance matrix $\mathbf{\Sigma}$, and add convergence guarantees for GROUSE and PGF. For a more detailed account of the state-of-the-art convergence results on PCA we refer the reader to (Allen-Zhu and Li, 2017, Table 1).

Theorem 2 (Minor extension of (Allen-Zhu and Li, 2017), Theorem 1)**.** *Suppose we observe an i.i.d. stream of data vectors $\mathbf{x}_t$ drawn from a bounded*[4] *distribution with covariance $\mathbf{\Sigma}$. Initialize the algorithms with a matrix drawn from a uniformly continuous distribution on the Grassmannian*[5]*. Let λ_i be the i^{th} eigenvalue of $\mathbf{\Sigma}$, $\Delta := \lambda_k - \lambda_{k+1} \in (0, \frac{1}{k}]$, and $\Lambda := \sum_{i=1}^{k} \lambda_i \in (0, 1]$. Then for $\delta \in (0, 1)$, define*

$$T_0 = C_0 \frac{k\Lambda}{\Delta^2 \delta^2},\ T_1 = C_1 \frac{\Lambda}{\Delta^2},$$

and let the step size schedule for Oja's algorithm be

$$\eta_t = \begin{cases} C_2 \frac{1}{\Delta T_0} & 1 \le t \le T_0; \\ C_3 \frac{1}{\Delta T_1} & T_0 < t \le T_0 + T_1; \\ C_4 \frac{1}{\Delta (t - T_0)} & t > T_0 + T_1. \end{cases}$$

where C_i in all cases denotes a function of absolute constants as well as $\log(\frac{1}{\delta}), \log(\frac{1}{\Delta})$, and $\log d$, but with no other dependence on problem parameters[6]*.*

Using equation (14)*, set the step size schedule for GROUSE to be*

$$\theta_t = \begin{cases} \frac{1}{\|\mathbf{r}\|\|\mathbf{w}\|} \arctan\left(\frac{C_2 \|\mathbf{r}\|\|\mathbf{w}\|}{\Delta T_0 + C_2 \|\mathbf{w}\|^2}\right) & 1 \le t \le T_0; \\ \frac{1}{\|\mathbf{r}\|\|\mathbf{w}\|} \arctan\left(\frac{C_3 \|\mathbf{r}\|\|\mathbf{w}\|}{\Delta T_1 + C_3 \|\mathbf{w}\|^2}\right) & T_0 < t \\ & \le T_0 + T_1; \\ \frac{1}{\|\mathbf{r}\|\|\mathbf{w}\|} \arctan\left(\frac{C_4 \|\mathbf{r}\|\|\mathbf{w}\|}{\Delta (t - T_0) + C_4 \|\mathbf{w}\|^2}\right) & t > T_0 + T_1 \end{cases}$$

and using equation (15)*, set the step size schedule for PGF to be*

$$\gamma_t = \begin{cases} \frac{C_2}{\Delta T_0 + C_2 \|\mathbf{w}\|^2} & 1 \le t \le T_0; \\ \frac{C_3}{\Delta T_1 + C_3 \|\mathbf{w}\|^2} & T_0 < t \le T_0 + T_1; \\ \frac{C_4}{\Delta (t - T_0) + C_4 \|\mathbf{w}\|^2} & t > T_0 + T_1. \end{cases}$$

Let $\mathbf{Q} \in \mathbb{R}^{d \times (d-k)}$ be a matrix with orthonormal columns spanning the same space as all the eigenvectors of $\mathbf{\Sigma}$ with eigenvalues no more than λ_{k+1}.

Then for $\varepsilon \in (0, 1)$, and every $T = T_0 + T_1 + C_5 \frac{T_1}{\varepsilon}$ the outputs $\mathbf{U}_T \in \mathbb{R}^{d \times k}$ of all three algorithms are equal and satisfy $\|\mathbf{Q}^T \mathbf{U}_T\|_F^2 \le \varepsilon$ with probability $\ge 1 - \delta$.

This convergence result divides the step size schedule into three periods. Until time $T_0 + T_1$, the step size does not diminish with t. After that, the step size diminishes, smoothing the effect of the random gradients.

Further results by Allen-Zhu and Li (2017) show that these results can be extended to be "gap-free," which means that the error is bounded as $\|\mathbf{P}^T \mathbf{U}_T\|_F^2 \le \varepsilon$, where $\mathbf{P}$ includes only the eigenvectors associated with eigenvalues of $\mathbf{\Sigma}$ no more than $\lambda_k - \rho$ for some $\rho \in (0, 1)$. Additionally they propose a gradual initialization that improves the length of the initial phase roughly when $\varepsilon < 1/k$.

4.2 Local convergence with compressively sampled data

Another interesting modern machine learning problem is that of learning principal components when data vectors are sampled with a compressive linear sampling operator. This problem is called "low-rank matrix sensing" and has been studied extensively in the batch setting; see (Recht et al., 2010; Candes and Plan, 2011; Chi et al., 2019). A related problem is low-rank matrix completion, where the compressive sampling operator samples individual entries of the low-rank matrix. Several matrix completion algorithms have guarantees for recovering the underlying components in the batch setting (Candès and Recht, 2009; Keshavan et al., 2010; Koltchinskii et al., 2011; Zilber and Nadler, 2021). For both problems, guarantees for streaming matrix recovery algorithms are quite few

[4] More precisely, in (Allen-Zhu and Li, 2017), they assume $\|\mathbf{x}_t\| \le 1$ with probability 1, but this can of course be extended to any bound by adjusting the results.

[5] The result in (Allen-Zhu and Li, 2017) uses a random Gaussian matrix, which is possible for Oja's algorithm because it does not technically require the initialization to have orthonormal columns. But this easily extends to a random matrix with orthonormal columns using standard high-dimensional probability results (Vershynin, 2018).

[6] This is in place of the order notation in (Allen-Zhu and Li, 2017), where $\widetilde{\Theta}$ hides constants and log dependency on the mentioned terms, e.g. $T_0 = \widetilde{\Theta}\left(\frac{k\Lambda}{\Delta^2 \delta^2}\right)$.

and far between (Balzano et al., 2018). The GROUSE algorithm, which was originally developed to handle missing data, has local convergence guarantees in both cases. However, these results assume that $\mathbf{\Sigma}$ is exactly low-rank, and even further, that the nonzero eigenvalues of $\mathbf{\Sigma}$ are all exactly one. Here we state the theorem for compressively sampled data, and we save the result for missing data for the supplement.

In order to handle missing or compressively sampled data, we introduce a $m \times d$ linear measurement operator $\mathbf{A}_t$ to measure each streaming vector. The algorithms must only be altered slightly as follows. Let our stream of data vectors $\mathbf{x}_t$ be observed as $\mathbf{y}_t = \mathbf{A}_t\mathbf{x}_t$, and we will abuse notation to create $\mathbf{x}_t$ that is used by the algorithms:

$$\mathbf{w}_t = \arg\min_{\mathbf{a}} \|\mathbf{y}_t - \mathbf{A}_t\mathbf{U}_t\mathbf{a}_t\|_2 \ , \quad \mathbf{p}_t = \mathbf{U}_t\mathbf{w}_t$$
$$\mathbf{r}_t = \mathbf{A}_t^T\left(\mathbf{y}_t - \mathbf{A}_t\mathbf{U}_t\mathbf{w}_t\right) \ , \quad \mathbf{x}_t = \mathbf{U}_t\mathbf{w}_t + \mathbf{r}_t \ .$$

With these substitutions, the arguments of Theorem 1 still apply. This is the GROUSE algorithm analyzed by Zhang and Balzano (2022) to have expected local linear convergence, which we now state formally for all algorithms.

Theorem 3 (Adapted from (Zhang and Balzano, 2022), Theorem 8 and Corollary 13)**.** *Suppose we observe compressive measurements of an i.i.d. stream of data vectors $\mathbf{x}_t$ drawn from a distribution with covariance $\mathbf{\Sigma} = \bar{\mathbf{U}}\bar{\mathbf{U}}^T$ for some matrix $\bar{\mathbf{U}} \in \mathbb{R}^{d\times k}$ with orthonormal columns. Let the compressive measurement operator $\mathbf{A}_t$ at each time step be of size $m \times d$ with i.i.d $\mathcal{N}(0, 1/d)$ entries, i.e., we observe $\mathbf{A}_t\mathbf{x}_t$. At time t, let the principal angles ϕ_i, $i = 1, \ldots, k$ between $\mathbf{U}_t$ and $\bar{\mathbf{U}}$ be such that $\sum_{i=1}^k \sin^2(\phi_i) \le \varepsilon < 1$. By convention (Golub and Loan, 2012, Section 6.4.3), $0 \le \phi_1 \le \cdots \le \phi_k \le 1$.*

At time t, let the step size of GROUSE be $\theta_t = \arctan\left(\frac{\|\mathbf{r}_t\|}{\|\mathbf{w}_t\|}\right)$, the step size of Oja's algorithm be

$$\eta_t = \frac{\tan(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)}{\|\mathbf{r}_t\|\|\mathbf{w}_t\| - \|\mathbf{w}_t\|^2\tan(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)} \ ,$$

and the step size of PGF to be

$$\gamma_t = \frac{\tan(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)}{\|\mathbf{r}_t\|\|\mathbf{w}_t\|} \ .$$

Then all three algorithms' outputs are equal.

Let $\kappa_t = 1 - \det(\bar{\mathbf{U}}^T\mathbf{U}_t\mathbf{U}_t^T\bar{\mathbf{U}})$; κ_t is small when the subspaces are close. Then if

$$m \ge C_6 \max\{\log d + k, (\tan(\phi_k) + k)^2\}$$

for absolute constant C_6, then with probability at least $1 - 2/d^2 - e^{-k/128}$ with respect to the random compressive measurement operator $\mathbf{A}_t$,

$$\mathbb{E}[\kappa_{t+1}] \le \left(1 - \frac{2}{3}\frac{m}{d}\frac{1-\varepsilon}{k}\right)\kappa_t \ ,$$

where expectation is taken with respect to the random data $\mathbf{x}_t$.

This theorem establishes expected linear convergence in a local region of the planted subspace $\bar{\mathbf{U}}$. This could be extended to high-probability linear convergence if the sequence κ_t were monotonic using e.g. (Richtárik and Takáč, 2014, Theorem 1), but in general with compressive measurements κ_t will not be monotonic. More sophisticated Martingale arguments are more difficult to apply because of the dependence of the gradient on $\mathbf{U}_t$, which can potentially be avoided with analysis of Oja's algorithm as mentioned before, though the algorithm as described for compressive measurements does introduce some dependence. The local region requires that $\sum_{i=1}^k \sin^2(\phi_i) \le \varepsilon < 1$; we note that this sum could be as large as k when the initial subspace $\mathbf{U}_0$ is orthogonal to the planted subspace $\bar{\mathbf{U}}$, so the requirement is somewhat restrictive. Since the theorem makes a very strict assumption on the data generation process, specifically that $\mathbf{\Sigma} = \bar{\mathbf{U}}\bar{\mathbf{U}}^T$ is exactly low-rank, a step size diminishing with t is not required.

The missing-data version, where $\mathbf{A}_t$ is an operator that samples a subset of the entries of $\mathbf{x}_t$, of Oja's algorithm was discussed in (Balzano et al., 2018; Wang et al., 2018) and analyzed in the asymptotic regime in (Wang et al., 2018). The missing-data version of the GROUSE algorithm was analyzed in (Balzano and Wright, 2015; Zhang and Balzano, 2016, 2022), and we include one such result in the supplement.

5 DISCUSSION AND CONCLUSION

This paper has proven the equivalence of Oja's algorithm and the GROUSE algorithm for streaming PCA and subspace tracking. This equivalence result was then used to port global convergence results from Oja's algorithm to GROUSE, as well as to a variant we called PGF, which is also equivalent. We were also able to port local convergence results for compressively sampled vectors from GROUSE to Oja's algorithm and PGF.

PCA is used in a wide variety of scientific applications, and streaming PCA specifically seeks to reduce the computation and memory footprint of the PCA computation. If our theory provides insight and allows others to improve streaming PCA algorithms, we hope it has a positive impact for reducing computational requirements. An important future direction

for streaming PCA with missing or compressed data is to prove global convergence, possibly using ideas from recent works such as (Ge et al., 2017) that use regularization to improve algorithmic properties.

There is significant room for improvement and generalization in this work. As we discussed in the related work section, there are numerous algorithms for streaming PCA, and one would not expect researchers to go through individually and prove equivalence. Pursuing a direction similar to (Zhao et al., 2021), that generalizes the notions of equivalence in algorithms but applies to non-convex objectives or even PCA specifically, is of great interest.

Acknowledgements

Thanks to Kyle Gilman, Alex Ritchie, Peng Wang, Can Yaras, Dejiao Zhang, and the anonymous reviewers whose comments strengthened this manuscript. The work was supported by AFOSR YIP award FA9550-19-1-0026.

References

Z. Allen-Zhu and Y. Li. First efficient convergence for streaming k-PCA: A global, gap-free, and near-optimal rate. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 487–492, Oct 2017. doi: 10.1109/FOCS.2017.51.

Ehsan Amid and Manfred K Warmuth. An implicit form of Krasulina's k-PCA update without the orthonormality constraint. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3179–3186, 2020.

Raman Arora and Teodor Vanislavov Marinov. Efficient convex relaxations for streaming PCA. *Advances in Neural Information Processing Systems*, 32:10496–10505, 2019.

Raman Arora, Andy Cotter, and Nati Srebro. Stochastic optimization of PCA with capped MSG. In *Advances in Neural Information Processing Systems*, pages 1815–1823, 2013.

Samir Attallah and Karim Abed-Meraim. Fast algorithms for subspace tracking. *IEEE Signal Processing Letters*, 8(7):203–206, 2001.

Roland Badeau, Bertrand David, and Gaël Richard. Fast approximated power iteration subspace tracking. *IEEE Transactions on Signal Processing*, 53(8): 2931–2941, 2005.

Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems*, pages 3174–3182, 2013.

L. Balzano and S. J. Wright. On GROUSE and incremental SVD. In *IEEE Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2013.

Laura Balzano and Stephen J Wright. Local convergence of an algorithm for subspace identification from partial data. *Foundations of Computational Mathematics*, 15(5):1279–1314, 2015.

Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *48th Annual Allerton Conference on Communication, Control, and Computing*, pages 704–711. IEEE, 2010.

Laura Balzano, Yuejie Chi, and Yue M Lu. Streaming PCA and subspace tracking: The missing data case. *Proceedings of the IEEE*, 106(8):1293–1310, 2018.

Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010 (1-38):3, 2011.

Christian H. Bischof and Gautam M. Shroff. On updating signal subspaces. *IEEE Transactions on Signal Processing*, 40(1), January 1992.

Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006.

J.R. Bunch and C.P. Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31: 111–129, 1978. ISSN 0029-599X.

Emmanuel J Candes and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Shing-Chow Chan, Yu Wen, and Ka-Leung Ho. A robust past algorithm for subspace tracking in impulsive noise. *IEEE transactions on signal processing*, 54(1):105–116, 2005.

Shing-Chow Chan, Hai-Jun Tan, Jian-Qiang Lin, and Bin Liao. A new local polynomial modeling based variable forgetting factor and variable regularized past algorithm for subspace tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 54(3): 1530–1544, 2018.

Chanchal Chatterjee. Adaptive algorithms for first principal eigenvector computation. *Neural Networks*, 18(2):145–159, 2005.

Tianping Chen, Yingbo Hua, and Wei-Yong Yan. Global convergence of Oja's subspace algorithm for

principal component extraction. *IEEE Transactions on Neural Networks*, 9(1):58–67, 1998.

Yuejie Chi, Yonina C Eldar, and Robert Calderbank. Petrels: Subspace estimation and tracking from partial observations. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3301–3304. IEEE, 2012.

Yuejie Chi, Yonina C Eldar, and Robert Calderbank. Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations. *IEEE Transactions on Signal Processing*, 61(23): 5947–5959, 2013.

Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

Pierre Comon and Gene H Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.

Christopher De Sa, Kunle Olukotun, and Christopher Ré. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 2332–2341. JMLR. org, 2015.

Xenofon G Doukopoulos and George V Moustakides. Fast and stable subspace tracking. *IEEE Transactions on Signal Processing*, 56(4):1452–1465, 2008.

Alan Edelman, Tomas A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

Dan Garber. On the regret minimization of nonconvex online gradient ascent for online PCA. In *Conference on Learning Theory*, pages 1349–1373. PMLR, 2019.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

Ian Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. Eigengame: Pca as a nash equilibrium. In *International Conference on Learning Representations*, 2020.

Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

Gene Golub and Charles Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2012.

Tony Gustafsson. Instrumental variable subspace tracking using projection approximation. *IEEE transactions on signal processing*, 46(3):669–681, 1998.

Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. In *Advances in Neural Information Processing Systems*, pages 2861–2869, 2014.

Amelia Henriksen and Rachel Ward. Adaoja: Adaptive learning rates for streaming PCA. *arXiv preprint arXiv:1905.12115*, 2019.

Yingbo Hua, Yong Xiang, Tianping Chen, Karim Abed-Meraim, and Yongfeng Miao. A new look at the power method for fast subspace tracking. *Digital Signal Processing*, 9(4):297–314, 1999.

De Huang, Jonathan Niles-Weed, and Rachel Ward. Streaming k-PCA: Efficient guarantees for Oja's algorithm, beyond rank-one updates. In *Conference on Learning Theory*, pages 2463–2498. PMLR, 2021.

Prateek Jain, Chi Jin, Sham M Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming PCA: Matching matrix bernstein and near-optimal finite sample guarantees for Oja's algorithm. In *Conference on Learning Theory*, pages 1147–1164, 2016.

Sajid Javed, Praneeth Narayanamurthy, Thierry Bouwmans, and Namrata Vaswani. Robust PCA and robust subspace tracking: A comparative evaluation. In *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pages 836–840. IEEE, 2018.

Ilkka Karasalo. Estimating the covariance matrix by signal subspace averaging. *IEEE Transactions on acoustics, speech, and signal processing*, 34(1):8–12, 1986.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6): 2980–2998, 2010.

Vladimir Koltchinskii, Karim Lounici, Alexandre B Tsybakov, et al. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

Wojciech Kotłowski and Gergely Neu. Bandit principal component analysis. In *Conference On Learning Theory*, pages 1994–2024. PMLR, 2019.

T.P. Krasulina. The method of stochastic approximation for the determination of the least eigenvalue of a symmetrical matrix. *USSR Computational Mathematics and Mathematical Physics*, 9(6):189 – 195, 1969. ISSN 0041-5553.

Chun-Liang Li, Hsuan-Tien Lin, and Chi-Jen Lu. Rivalry of two families of algorithms for memory-restricted streaming PCA. In *Artificial Intelligence and Statistics*, pages 473–481, 2016.

Xingguo Li, Junwei Lu, Raman Arora, Jarvis Haupt, Han Liu, Zhaoran Wang, and Tuo Zhao. Symmetry, saddle points, and global optimization landscape of nonconvex matrix factorization. *IEEE Transactions on Information Theory*, 65(6):3489–3514, 2019.

Xin Liang. On the optimality of the Oja's algorithm for online PCA. *arXiv preprint arXiv:2104.00512*, 2021.

Robert Lunde, Purnamrita Sarkar, and Rachel Ward. Bootstrapping the error of oja's algorithm. *Advances in Neural Information Processing Systems*, 34, 2021.

Per-Gunnar Martinsson and Joel A Tropp. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica*, 29:403–572, 2020.

G Mathew, Vellenki Reddy, and Soura Dasgupta. Adaptive estimation of eigensubspace. *IEEE Transactions on Signal Processing*, 43(2), February 1995.

Tyler Maunu, Teng Zhang, and Gilad Lerman. A well-tempered landscape for non-convex robust subspace recovery. *Journal of Machine Learning Research*, 20 (37), 2019.

Marc Moonen, Paul Van Dooren, and Joos Vandewalle. An SVD updating algorithm for subspace tracking. *SIAM Journal of Matrix Analysis and Applications*, 13:1015–1038, 1992.

Jiazhong Nie, Wojciech Kotłowski, and Manfred K Warmuth. Online PCA with optimal regret. *The Journal of Machine Learning Research*, 17(1):6022–6070, 2016.

Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985.

Edward C Real, Donald W Tufts, and James W Cooley. Two algorithms for fast approximate subspace tracking. *IEEE Transactions on Signal Processing*, 47(7):1936–1945, 1999.

Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

Ohad Shamir. Fast stochastic algorithms for SVD and PCA: Convergence properties and convexity. In *International Conference on Machine Learning*, pages 248–256. PMLR, 2016.

Steven T. Smith. *Geometric Optimization Methods for Adaptive Filtering*. PhD thesis, Harvard University, 1993.

G. W. Stewart. An updating algorithm for subspace tracking. *IEEE Transactions on Signal Processing*, 1992.

Peter Strobach. The fast recursive row-householder subspace tracking algorithm. *Signal Processing*, 89 (12):2514–2528, 2009.

Cheng Tang. Exponentially convergent stochastic k-PCA without variance reduction. *Advances in Neural Information Processing Systems*, 32:12393–12404, 2019.

Nilesh Tripuraneni, Nicolas Flammarion, Francis Bach, and Michael I Jordan. Averaging stochastic gradient descent on riemannian manifolds. In *Conference on Learning Theory*, pages 650–687, 2018.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Chuang Wang, Yonina C Eldar, and Yue M Lu. Subspace estimation from incomplete observations: A high-dimensional analysis. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1240–1252, 2018.

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

David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Peng Xu, Bryan He, Christopher De Sa, Ioannis Mitliagkas, and Chris Re. Accelerated stochastic power iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 58–67. PMLR, 2018.

Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal processing*, 43(1): 95–107, 1995.

Jar-Ferr Yang and Mostafa Kaveh. Adaptive eigensubspace algorithms for direction or frequency estimation and tracking. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(2), February 1988.

Puyudi Yang, Cho-Jui Hsieh, and Jane-Ling Wang. History PCA: A new algorithm for streaming PCA. *arXiv preprint arXiv:1802.05447*, 2018.

Zhang Yi, Mao Ye, Jian Cheng Lv, and Kok Kiong Tan. Convergence analysis of a deterministic discrete time system of Oja's PCA learning algorithm. *IEEE Transactions on Neural Networks*, 16 (6):1318–1328, 2005.

Jinchun Zhan, Brian Lois, Han Guo, and Namrata Vaswani. Online (and offline) robust PCA: Novel algorithms and performance guarantees. In *Artificial Intelligence and Statistics*, pages 1488–1496, 2016.

Dejiao Zhang and Laura Balzano. Global convergence of a grassmannian gradient descent algorithm for subspace estimation. In *Proceedings of Artificial Intelligence and Statistics*, 2016.

Dejiao Zhang and Laura Balzano. Convergence of a grassmannian gradient descent algorithm for subspace estimation from undersampled data. Technical report, University of Michigan, 2022. Available at `https://dx.doi.org/10.7302/4151` with historical versions at `https://arxiv.org/abs/1610.00199`.

Shipu Zhao, Laurent Lessard, and Madeleine Udell. An automatic system to detect equivalence between iterative algorithms. *arXiv preprint arXiv:2105.04684*, 2021.

Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B Wakin. The global optimization geometry of low-rank matrix optimization. *IEEE Transactions on Information Theory*, 67(2):1308–1331, 2021.

Pini Zilber and Boaz Nadler. Gnmr: A provable one-line algorithm for low rank matrix recovery. *arXiv preprint arXiv:2106.12933*, 2021.

Supplementary Material: On the equivalence of Oja's algorithm and GROUSE

A GRADIENT CALCULATIONS

The two objective functions are given as:

$$\mathrm{Tr}(\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U}) = \sum_{t=1}^{n} \mathrm{Tr}(\mathbf{U}^T\mathbf{x}_t\mathbf{x}_t^T\mathbf{U}) =: \sum_{t=1}^{n} F_t^{(\mathrm{Trace})}(\mathbf{U}) \ ,$$

$$\left\|\mathbf{U}\mathbf{U}^T\mathbf{X} - \mathbf{X}\right\|_F^2 = \sum_{t=1}^{n} \left\|\mathbf{U}\mathbf{U}^T\mathbf{x}_t - \mathbf{x}_t\right\|_F^2 =: \sum_{t=1}^{n} F_t^{(\mathrm{Frob})}(\mathbf{U}) \ .$$

As we know well, these two objective functions are equivalent when $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. When we first derive the gradients without enforcing this constraint, we have the following.

$$\nabla_{\mathbf{U}} \left(\sum_{t=1}^{n} F_t^{(\mathrm{Trace})}(\mathbf{U}) \right) = \nabla_{\mathbf{U}} \left(\sum_{t=1}^{n} \mathrm{Tr}(\mathbf{U}^T\mathbf{x}_t\mathbf{x}_t^T\mathbf{U}) \right) = 2\sum_{t=1}^{n} \mathbf{x}_t\mathbf{x}_t^T\mathbf{U} = 2\sum_{t=1}^{n} \mathbf{x}_t\mathbf{w}_t^T$$

$$\begin{aligned}
\nabla_{\mathbf{U}} \left(\sum_{t=1}^{n} F_t^{(\mathrm{Frob})}(\mathbf{U}) \right) &= \sum_{t=1}^{n} \nabla_{\mathbf{U}} \left(\mathrm{Tr}((\mathbf{U}\mathbf{U}^T\mathbf{x}_t - \mathbf{x}_t)^T(\mathbf{U}\mathbf{U}^T\mathbf{x}_t - \mathbf{x}_t)) \right) \\
&= \sum_{t=1}^{n} \nabla_{\mathbf{U}} \left(\mathbf{x}_t^T\mathbf{U}\mathbf{U}^T\mathbf{U}\mathbf{U}^T\mathbf{x}_t - 2\mathbf{x}_t^T\mathbf{U}\mathbf{U}^T\mathbf{x}_t + \mathbf{x}_t^T\mathbf{x}_t \right) \\
&= 2\sum_{t=1}^{n} \mathbf{U}\mathbf{U}^T\mathbf{x}_t\mathbf{x}_t^T\mathbf{U} + \mathbf{x}_t\mathbf{x}_t^T\mathbf{U}\mathbf{U}^T\mathbf{U} - 2\mathbf{x}_t\mathbf{x}_t^T\mathbf{U} \\
&= 2\sum_{t=1}^{n} \mathbf{p}_t\mathbf{w}_t^T + \mathbf{x}_t\mathbf{w}_t^T(\mathbf{U}^T\mathbf{U} - 2\mathbf{I}) \ .
\end{aligned}$$

Now imposing $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ results in the gradient used by GROUSE, $-2\sum_{t=1}^{n} \mathbf{r}_t\mathbf{w}_t^T$. It remains an interesting open question as to what other constraints might guarantee that the problems or their gradient algorithms are equivalent, such as a constraint on column norms or on Frobenius norm of $\mathbf{U}$.

B EMPIRICAL VALIDATION

While the empirical validation of our results is not especially interesting, we include it for completeness. We plot a few example runs for $d = 100, k = 10$ and fully observed vectors. We fixed $\eta_t = 0.01$ for Oja's algorithm and computed the corresponding Grouse step size θ_t at every iteration. We plot the error to the true subspace in Figure 2. The errors (and the subspace estimates) are identical at *every* iteration. In Figure 3, the planted subspace varies with time, and again the estimates are identical.

C THEORETICAL DETAILS

In the main paper we provided two theorems without proof, based on existing theorems in the literature. Theorem 2 provided the global convergence of GROUSE and PGF, based on (Allen-Zhu and Li, 2017, Theorem 1). There

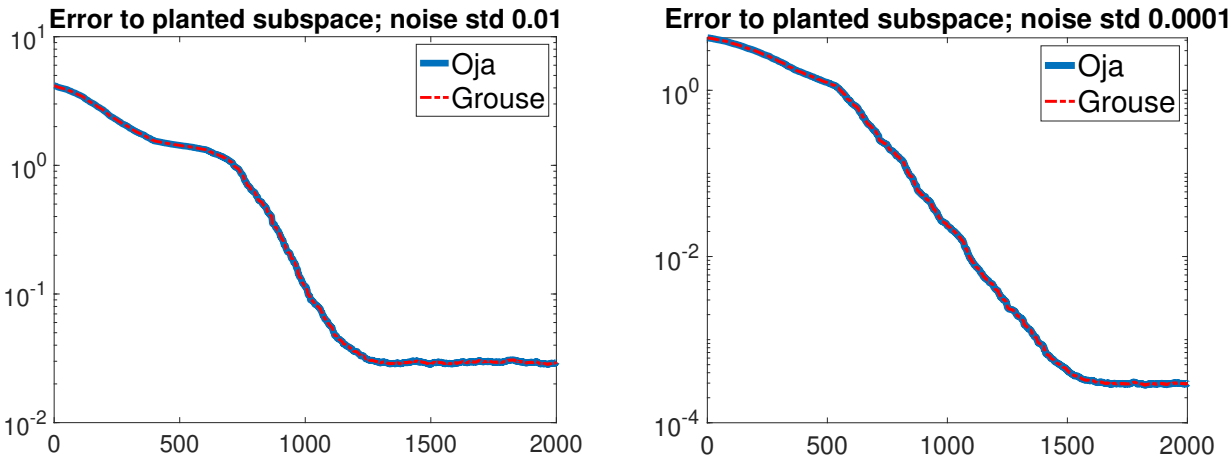

Figure 2: Oja and Grouse performance on a given run with the theoretical step size to make them equivalent. On the y-axis we plot the error computed as $\|U_{\text{est}}U_{\text{est}}^T - U_{\text{true}}U_{\text{true}}^T\|_F$ over 2000 iterations. The lines overlap exactly. We also computed the Frobenius norm error between the projection matrices generated by the two algorithms; the maximum error over all iterations was 2.1553×10^{-14}.

are no missing pieces for this theorem, as we used the result directly. Theorem 3 provided local convergence of Oja's algorithm and PGF, based primarily on (Zhang and Balzano, 2022, Theorem 8), but using other pieces of that work. Additionally, we only loosely defined the algorithms using compressive measurements. Here we will flesh out those details.

First we detail the three algorithms with compressive or missing measurements. As in the theorem, the compressive measurement operator of size $m \times d$ may be, for example, a matrix with i.i.d $\mathcal{N}(0, 1/d)$ entries. But in general, it could be any sketch that preserves the geometry[7] of $\bar{\mathbf{U}}$, the planted subspace in Theorem 3. For the streaming matrix completion problem, $\mathbf{A}_t$ would have a 1 in each column where that entry of $\mathbf{x}_t$ is observed, and a zero otherwise. The updates are easiest to interpret in that case: they are a linear combination of 1) the predicted projection of the full data onto $\mathbf{U}_t$: $\mathbf{p}_t = \mathbf{U}_t\mathbf{w}_t$, where $\mathbf{w}_t$ are the best fit weights given the compressed measurements, and 2) the residual only on the observed entries $\Omega_t \subset \{1, \ldots, d\}$:

$$\mathbf{r}_t = \mathbf{A}_t^T(\mathbf{y}_t - \mathbf{A}_t\mathbf{p}_t) = \begin{cases} \mathbf{x}_t(i) - \mathbf{p}_t(i) & i \in \Omega_t \\ 0 & \text{otherwise} \end{cases},$$

where we have denoted the i^{th} vector entry by $\mathbf{x}_t(i)$. Once again we can see that all three updates are a linear combination of these same two vectors, $\mathbf{p}_t$ and $\mathbf{r}_t$, slightly redefined.

Proof of Theorem 3. The paper we are referencing defines a similarity metric $\zeta_t = 1 - \kappa_t$ as we defined it in the theorem.

Our result uses the final statement of (Zhang and Balzano, 2022, Theorem 8) with their $\delta = 1/4$. Let $\beta = \frac{10}{9/64}$ and $\gamma_1 = \frac{\frac{3}{4}(1-\frac{1}{2}\sqrt{\frac{m}{d}})}{(1+\sqrt{5\frac{k}{m}})^2}$. Let

$$m \geq k \max\left\{512\log(96d^{2/k}), \beta\left(\tan(\phi_k) + \frac{1}{4}k\cos(\phi_k)\right)\left(\tan(\phi_k) + \frac{1}{4}k\cos(\phi_k) + \frac{1}{2}\right)\right\}$$

then with probability at least $1 - 2/d^2 - e^{-k/128}$ with respect to the random compressive measurement operator $\mathbf{A}_t$, we have

$$\mathbb{E}[1 - \kappa_{t+1}] \geq \left(1 + \frac{1}{2\gamma_1}\frac{m}{d}\frac{\kappa_t}{k}\right)(1 - \kappa_t) \quad \forall t\,,$$

where expectation is taken with respect to the random data $\mathbf{x}_t$.

[7] In other words, it could be any subspace embedding (Martinsson and Tropp, 2020; Woodruff, 2014).

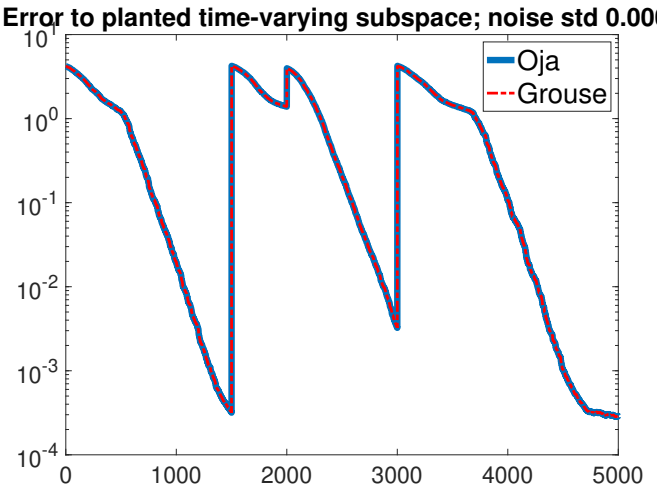

Figure 3: Oja and Grouse performance in a setting where the planted subspace changes at time 1500, 2000, and 3000. Again this is the outcome on a given run with the theoretical step size to make them equivalent. On the y-axis we plot the error computed as $\|U_{\text{est}}U_{\text{est}}^T - U_{\text{true}}U_{\text{true}}^T\|_F$ over 5000 iterations. The lines overlap exactly.

To get our statement we first get a larger lower bound for m by seeing that $\beta < 72$, $k\log(96d^{2/k}) = 2\log d + k\log(96)$, and since $\cos(\phi_k) \leq 1$ and $k \geq 1$,

$$\left(\tan(\phi_k) + \frac{1}{4}k\cos(\phi_k)\right)\left(\tan(\phi_k) + \frac{1}{4}k\cos(\phi_k) + \frac{1}{2}\right) < (\tan(\phi_k) + k)^2 \ .$$

To get the expected decrease in κ_t (as opposed to increase in $1-\kappa_t$) we need to rearrange. First plug in γ_1 to see that

$$\begin{aligned}\mathbb{E}[1-\kappa_{t+1}] \geq \left(1 + \frac{1}{2\gamma_1}\frac{m}{d}\frac{\kappa_t}{k}\right)(1-\kappa_t) &= \left(1 + \frac{(1+\sqrt{5\frac{k}{m}})^2}{2\frac{3}{4}(1-\frac{1}{2}\sqrt{\frac{m}{d}})}\frac{m}{d}\frac{\kappa_t}{k}\right)(1-\kappa_t) \\ &= \left(1 + \frac{2}{3}\frac{(\sqrt{m}+\sqrt{5k})^2}{(d-\frac{1}{3}\sqrt{md})}\frac{\kappa_t}{k}\right)(1-\kappa_t) \geq \left(1 + \frac{2}{3}\frac{m}{d}\frac{\kappa_t}{k}\right)(1-\kappa_t) \ .\end{aligned}$$

Now we have

$$\mathbb{E}[\kappa_{t+1}] \leq 1 - \left(1 + \frac{2}{3}\frac{m}{d}\frac{\kappa_t}{k}\right)(1-\kappa_t) = \left(1 - \frac{2}{3}\frac{m}{d}\frac{1-\kappa_t}{k}\right)\kappa_t \tag{25}$$

We will now use the assumption that $\sum_{i=1}^k \sin^2(\phi_i) \leq \varepsilon < 1$. In the proof of (Zhang and Balzano, 2022, Corollary 13), they show that

$$1 - \kappa_t = \prod_{i=1}^k \cos^2(\phi_k) \geq 1 - \sum_{i=1}^k \sin^2(\phi_i) \geq 1 - \varepsilon > 0$$

which when we substitute into (25) gives

$$\mathbb{E}[\kappa_{t+1}] \leq \left(1 - \frac{2}{3}\frac{m}{d}\frac{1-\varepsilon}{k}\right)\kappa_t \ .$$

□

D MISSING DATA

To handle missing data we need the definition of coherence.

Definition 1. *A subspace spanned by columns in* $\mathbf{U}$ *has coherence parameter* μ *if*

$$\max_{i\in\{1,\dots,d\}} \|\mathcal{P}_{\mathbf{U}}\mathbf{e}_i\|_2^2 \leq \frac{\mu k}{d} ,$$

where $\mathbf{e}_i$ *is the* i^{th} *standard basis vector and* $\mathcal{P}_{\mathbf{U}}$ *is the orthogonal projection onto the column space of* $\mathbf{U}$*.*

This definition of coherence has $1 \leq \mu \leq \frac{d}{k}$ and can also be applied to a vector. This theorem uses $\mu(\bar{\mathbf{U}})$, the coherence of the true underlying subspace, as well as $\mu(\mathbf{r}_t)$, the coherence of the residual vector. The work in (Balzano and Wright, 2015) argues that this $\mu(\mathbf{r}_t)$ term is generally observed to be bounded, but they do not provide the bound; instead they provide an assumption on the bound of $\mu(\mathbf{r}_t)$ supported by empirical evidence.

Theorem 4 (Adapted from (Zhang and Balzano, 2022) Theorem 12)**.** *Suppose we observe vectors* $\mathbf{x}_t$ *on a subset of* m *entries selected uniformly with replacement, whose indices are stored in* $\Omega \subset \{1,\dots,m\}$*.*

At each time t*, let the step size of GROUSE be* $\theta_t = \arctan\left(\frac{\|\mathbf{r}_t\|}{\|\mathbf{p}_t\|}\right)$*, the step size of Oja's algorithm be*

$$\eta_t = \frac{\tan(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)}{\|\mathbf{r}_t\|\|\mathbf{w}_t\| - \|\mathbf{w}_t\|^2\tan(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)} ,$$

and the step size of PGF to be

$$\gamma_t = \frac{\tan(\theta_t\|\mathbf{r}_t\|\|\mathbf{w}_t\|)}{\|\mathbf{r}_t\|\|\mathbf{w}_t\|} .$$

Then all three algorithms' outputs are equal for all t*.*

Suppose $\sum_{i=1}^{k}\sin^2(\phi_k) \leq \varepsilon$*. If*

$$m > C_7 \max\left\{k\mu(\bar{\mathbf{U}})\log(d\sqrt{k}), \mu(\mathbf{r}_t)^2\log d, k\mu(\bar{\mathbf{U}})\mu(\mathbf{r}_t)\log d\right\}$$

then with probability at least $1 - 3/d^2$ *we have*

$$\mathbb{E}[\kappa_{t+1}] \leq \left(1 - \frac{1}{4}\frac{m}{d}\frac{1-\varepsilon}{k}\right)\kappa_t .$$

This theorem comes almost directly from (Zhang and Balzano, 2022) Theorem 12 and Corollary 13 and our equivalence Theorem 1. The only adjustment is in the third term for the lower bound on m, where we simplify a term from $k\mu(\bar{\mathbf{U}})(1+2\sqrt{\mu(\mathbf{r}_t)\log d})^2 \leq 9k\mu(\bar{\mathbf{U}})\mu(\mathbf{r}_t)\log d$.

Algorithm 4 Oja's algorithm with compressive or missing measurements (Balzano et al., 2018; Wang et al., 2018)

Given $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$;
Given step size scheme $\eta_t > 0$;
Set $t := 0$;
repeat
  Given sampling matrix $\mathbf{A}_t \in \mathbb{R}^{m \times d}$ and observation $\mathbf{y}_t = \mathbf{A}_t \mathbf{x}_t$;
  Define $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{y}_t - \mathbf{A}_t \mathbf{U}_t \mathbf{w}\|_2^2$;
  Define $\mathbf{p}_t := \mathbf{U}_t \mathbf{w}_t$ and $\mathbf{r}_t = \mathbf{A}_t^T (\mathbf{y}_t - \mathbf{A}_t \mathbf{p}_t)$.
  Update:

$$\widehat{\mathbf{U}}_{t+1} = \mathbf{U}_t + \eta_t (\mathbf{p}_t + \mathbf{r}_t) \mathbf{w}_t^T \tag{20}$$

$$\mathbf{U}_{t+1} = \Pi(\widehat{\mathbf{U}}_{t+1}) \tag{21}$$

  $t := t + 1$;
until termination

Algorithm 5 GROUSE (Balzano et al., 2010; Zhang and Balzano, 2022)

Given $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$;
Given step size scheme $\theta_t > 0$;
Set $t := 0$;
repeat
  Given sampling matrix $\mathbf{A}_t \in \mathbb{R}^{m \times d}$ and observation $\mathbf{y}_t = \mathbf{A}_t \mathbf{x}_t$;
  Define $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{y}_t - \mathbf{A}_t \mathbf{U}_t \mathbf{w}\|_2^2$;
  Define $\mathbf{p}_t := \mathbf{U}_t \mathbf{w}_t$ and $\mathbf{r}_t = \mathbf{A}_t^T (\mathbf{y}_t - \mathbf{A}_t \mathbf{p}_t)$.
  Update:

$$\mathbf{U}_{t+1} = \mathbf{U}_t + (\cos(\theta_t \|\mathbf{r}_t\| \|\mathbf{p}_t\|) - 1) \frac{\mathbf{p}_t}{\|\mathbf{p}_t\|} \frac{\mathbf{w}_t^T}{\|\mathbf{w}_t\|} + \sin(\theta_t \|\mathbf{r}_t\| \|\mathbf{p}_t\|) \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|} \frac{\mathbf{w}_t^T}{\|\mathbf{w}_t\|} \tag{22}$$

  $t := t + 1$;
until termination

Algorithm 6 PGF with compressive or missing measurements

Given $\mathbf{U}_0$, a $d \times k$ matrix with orthonormal columns, $0 < k < d$;
Given step size scheme $\gamma_t > 0$;
Set $t := 0$;
repeat
  Given sampling matrix $\mathbf{A}_t \in \mathbb{R}^{m \times d}$ and observation $\mathbf{y}_t = \mathbf{A}_t \mathbf{x}_t$;
  Define $\mathbf{w}_t := \arg\min_{\mathbf{w}} \|\mathbf{y}_t - \mathbf{A}_t \mathbf{U}_t \mathbf{w}\|_2^2$;
  Define $\mathbf{p}_t := \mathbf{U}_t \mathbf{w}_t$ and $\mathbf{r}_t = \mathbf{A}_t^T (\mathbf{y}_t - \mathbf{A}_t \mathbf{p}_t)$.
  Update:

$$\widehat{\mathbf{U}}_{t+1} = \mathbf{U}_t + \gamma_t \mathbf{r}_t \mathbf{w}_t^T \tag{23}$$

$$\mathbf{U}_{t+1} = \Pi(\widehat{\mathbf{U}}_{t+1}) \tag{24}$$

  $t := t + 1$;
until termination