Stochastic Variance-Reduced Heavy Ball Power Iteration

Cheolmin Kim [1] **Diego Klabjan** [1]

Abstract

We present a stochastic variance-reduced heavy ball power iteration algorithm for solving PCA and provide a convergence analysis for it. The algorithm is an extension of heavy ball power iteration, incorporating a step size so that progress can be controlled depending on the magnitude of the variance of stochastic gradients. The algorithm works with any size of the mini-batch, and if the step size is appropriately chosen, it attains global linear convergence to the first eigenvector of the covariance matrix in expectation. The global linear convergence result in expectation is analogous to those of stochastic variance-reduced gradient methods for convex optimization but due to non-convexity of PCA, it has never been shown for previous stochastic variants of power iteration since it requires very different techniques. We provide the first such analysis and stress that our framework can be used to establish convergence of the previous stochastic algorithms for any initial vector and in expectation. Experimental results show that the algorithm attains acceleration in a large batch regime, outperforming benchmark algorithms especially when the eigen-gap is small.

1. Introduction

Principal component analysis (PCA) is a fundamental tool for dimensionality reduction in machine learning and statistics. Given a data matrix $A = [a_1 a_2 \dots a_n] \in \mathbb{R}^{d\times n}$ consisting of n data vectors $a_1, a_2, \dots, a_n$ in $\mathbb{R}^d$, PCA finds a direction w onto which the projections of the data vectors have the largest variance. Assuming that the data vectors are standardized with a mean of zero and standard deviation of one, the PCA problem can be formulated as

$$\begin{aligned} \text{maximize} \quad & f(w) = \frac{1}{2n}\sum_{i=1}^{n}(a_i^T w)^2 \\ \text{subject to} \quad & \|w\|_2 = 1. \end{aligned} \tag{1}$$

Letting $C = \frac{1}{n}AA^T \in \mathbb{R}^{d\times d}$ be the covariance matrix, we can write the objective function as $f(w) = \frac{1}{2}w^T C w$. As the largest eigenvector u_1 of C maximizes $f(w)$, one can solve (1) by computing the singular value decomposition (SVD) of A. However, the runtime of SVD is $\mathcal{O}(\min\{nd^2, n^2 d\})$, which can be prohibitive in a large-scale setting.

An alternative way to solve (1) is to use power iteration (Golub & Van Loan, 2012) which repeatedly applies the following update step at each iteration

$$w_{t+1} = Cw_t, \quad w_{t+1} = \frac{w_{t+1}}{\|w_{t+1}\|_2}. \tag{2}$$

Since the gradient $\nabla f(w_t)$ is equal to Cw_t, the above update rule can be interpreted as obtaining the next iterate w_{t+1} by normalizing the gradient of the current iterate w_t. A sequence of iterates $\{w_t\}$ generated by power iteration (2) is guaranteed to achieve an ϵ-accurate solution after $\mathcal{O}\big(\frac{1}{\Delta}\log\frac{1}{\epsilon}\big)$ iterations, exhibiting linear convergence where $\Delta = \lambda_1 - \lambda_2$ is the eigen-gap and $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_d \geq 0$ are the eigenvalues of C. As each iteration involves multiplying a vector w_t with the matrix C, the total runtime becomes $\mathcal{O}\big(nd\frac{1}{\Delta}\log\frac{1}{\epsilon}\big)$. If n and d are both large, the runtime of power iteration is better than that of SVD. Nonetheless, it still largely depends on n and can be prohibitive when Δ is small. In order to reduce the dependence on Δ or n, the following variants of power iteration have been developed.

To reduce the dependence on Δ, (Xu et al., 2018) proposed power iteration with momentum (Power+M), which is a simple variant of power iteration utilizing the momentum idea of (Polyak, 1964). With the additional momentum term, it can be written as

$$w_{t+1} = 2Cw_t - \beta w_{t-1}. \tag{3}$$

In heavy ball power iteration (3), we have

$$(u_k^T w_t)^2 \leq \beta^t (u_k^T w_0)^2$$

[1]Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois, USA. Correspondence to: Cheolmin Kim <cheolminkim2019@u.northwestern.edu>.

if $\lambda_k \leq \beta$ and

$$(u_k^T w_t)^2 = \Theta\Big(\Big(\lambda_k + \sqrt{\lambda_k^2 - \beta}\Big)^{2t}\Big)(u_k^T w_0)^2$$

otherwise. Therefore, if $\beta = \lambda_2^2$, it achieves the optimal rate of convergence resulting in the runtime of $\mathcal{O}\big(nd\frac{1}{\sqrt{\Delta}}\log\frac{1}{\epsilon}\big)$, which greatly improves the dependence on Δ.

On the other hand, a stochastic algorithm utilizing a stochastic gradient $a_{i_t}a_{i_t}^T w_t$ rather than a full gradient Cw_t is introduced in (Oja, 1982). Since it requires just one data vector at a time, the computational cost per iteration is significantly reduced. However, due to the variance of stochastic gradients, a sequence of diminishing step sizes needs to be adopted in order to converge, making its progress slow near an optimum. Built on a recent stochastic variance-reduced gradient (SVRG) technique (Johnson & Zhang, 2013), (Shamir, 2015) proposed a stochastic variance-reduced version of Oja's algorithm (VR-PCA). By utilizing stochastic variance-reduced gradients with a constant step size, this sophisticated stochastic algorithm attains linear convergence, reducing the total runtime required to obtain an ϵ-accurate solution to $\mathcal{O}(d(n + \frac{1}{\Delta^2})\log\frac{1}{\epsilon})$.

Other works on the power method include the noisy (Hardt & Price, 2014), coordinate-wise (Lei et al., 2016), and shifted-and-inverted (Garber & Hazan, 2015; Garber et al., 2016) power methods. The noisy power method considers the power method in noise setting and (Balcan et al., 2016) extended it, providing an improved gap-dependency analysis. The shifted-and-inverted power method reduces the PCA problem to solving a series of convex least squares problems, which can be solved by optimization algorithms such as coordinate-descent (Wang et al., 2018), SVRG (Garber & Hazan, 2015; Garber et al., 2016), accelerated gradient descent or accelerated SVRG (Allen-Zhu & Li, 2016), and Riemannian gradient descent (Xu, 2018). Due to the presence of fast least squares solvers, the shifted-and-inverted approach has received much attention. However, since it involves solving a series of optimization problems, it is not simple to implement and hard to parallelize while the variance-reduced power methods are easy to implement and a single iteration can be parallelized in the obvious way.

In this paper, we present a stochastic variance-reduced algorithm (VR HB Power) for heavy ball power iteration (3). While a stochastic variance-reduced power iteration with momentum (VR Power+M) is introduced in (Xu et al., 2018), it is not practical to use since it requires that the size of the mini-batch needs to be sufficiently large. In this work, we enhance the algorithm by adding a step size which turns out to have a big impact. By incorporating the step size, the proposed algorithm can work with any size of the mini-batch. Given that the step size is appropriately chosen depending on the size of the mini-batch, the algorithm attains linear convergence to the first loading vector as VR-PCA. Furthermore, if the size of the mini-batch is chosen to be large, it attains accelerated convergence, outperforming VR-PCA. Table 1 summarizes the state-of-the-art.

For the algorithm, we provide a novel convergence analysis where the resulting convergence statement provides a bound for the ratio of two expectations that goes to zero at a linear rate. This result is analogous to those of stochastic variance-reduced gradient methods for convex optimization and stronger than probabilistic statements, appearing in (Shamir, 2015) and (Xu et al., 2018) in a sense that a probability parameter δ does not constraint the size of the mini-batch or the rate of convergence. Note that PCA studied herein is a non-convex problem and thus completely different techniques are needed. In order to obtain a convergence guarantee with high probability, the step size needs to be arbitrarily small in (Shamir, 2015) and the batch size needs to be arbitrarily large in (Xu et al., 2018). Complementary to them, our analysis does not have such requirements and the convergence is deterministically guaranteed in terms of expectation terms.

Moreover, our analysis allows random initialization while VR-PCA and VR Power+M do not. Since random initialization of $\tilde{w}_0$ results in $|u_1^T\tilde{w}_0| \leq \mathcal{O}(1/\sqrt{d})$ with high probability (Shamir, 2015), it is not trivial to obtain an initial iterate $\tilde{w}_0$ such that $|u_1^T\tilde{w}_0| \geq 1/2$, especially when d is large. To handle this issue, an initialization scheme that samples a point from the standard Gaussian distribution in $\mathbb{R}^d$ and performs a single power iteration is presented in (Shamir, 2016). After some stochastic iterations, this process yields an iterate $\tilde{w}_0$ satisfying $|u_1^T\tilde{w}_0| \geq 1/2$ with high probability. However, such an initialization scheme is not essentially necessary since VR-PCA practically works well with random initialization, as observed in (Shamir, 2015). Our convergence analysis resolves this gap, showing that the rate of convergence does not depend on how far an iterate is from u_1 but is kept the same across iterations, as in the case of deterministic power iteration. The framework used in the convergence analysis is not specific to the presented algorithm; it can be extended to analyze other stochastic variance-reduced PCA algorithms such as VR-PCA or VR Power+M, deriving in expectation bounds for them and resolving their initialization issues.

Our work has the following contributions.

1. We present a stochastic variance-reduced algorithm for heavy ball power iteration, which works with any size of the mini-batch and attains acceleration in a large-batch regime. Since there is no constraint on the size of the mini-batch, it is more practical than VR Power+M, and it outperforms VR-PCA in a large batch-setting, especially when the eigen-gap Δ is small.

2. We provide a novel convergence analysis for the algo-

Table 1. Comparison of stochastic variance-reduced methods for PCA and their convergence analyses. Algorithms are compared for two regimes (small-batch and large-batch), and types of convergence guarantee and conditions for the angle between an initial iterate $\tilde{w}_0$ and the first loading vector u_1 are summarized. "Local" means that there is a restriction on the angle while "global" implies no such restriction.

ALGORITHM	CONVERGENCE (SMALL-BATCH)	ACCELERATION (LARGE-BATCH)	CONVERGENCE GUARANTEE		REFERENCE
VR-PCA	$\surd$	$\times$	PROBABILISTIC	LOCAL	(SHAMIR, 2015)
VR POWER+M	$\times$	$\surd$	PROBABILISTIC	LOCAL	(XU ET AL., 2018)
VR HB POWER	$\surd$	$\surd$	EXPECTATION	GLOBAL	**(THIS PAPER)**

rithm. The convergence result does not require a good initialization, yet provides a bound for the ratio of two expectation terms, which has a similar form to those of stochastic variance-reduced gradient methods for convex optimization. The framework for the convergence analysis is general, therefore can be used to analyze other stochastic variance-reduced PCA algorithms. To this end, we are the first to establish convergence of VR-PCA and VR Power+M for any initial vector and in expectation.

3. We report numerical experiments on diverse datasets to investigate the empirical performance of the algorithm. Experimental results show that our algorithm is more efficient than VR-PCA in a large-batch setting, especially when the eigen-gap is small and it outperforms VR-Power+M in all cases.

The paper is organized as follows. We present the algorithm in Section 2 and the convergence analysis is provided in Section 3. Some practical considerations regarding the implementation of the algorithm are discussed in Section 4 and the experimental results are followed in Section 5.

2. Algorithm

In this section, we develop a stochastic variance-reduced algorithm for heavy ball power iteration (3). For eigenvalues and eigenvectors of C, we assume that the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_d$ satisfy

$$\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_d \geq 0$$

and the eigenvectors $u_1, u_2, \ldots, u_d$ form an orthonormal basis. Since a symmetric matrix is orthogonally diagonalizable, we can assume such eigenvectors exist without loss of generality.

Variance reduction algorithms periodically compute the exact gradient which is then used in the inner stochastic gradient type updates. This exact gradient reduces the variance of this inner loop. In summary, a variance reduction algorithm has an outer loop and an inner loop.

Let $\tilde{w}_s$ and w_t denote an outer-loop and inner-loop iterate, respectively. To get a stochastic variance-reduced gradient of an inner loop iterate w_t, we first decompose it into two parts as

$$w_t = \frac{(\tilde{w}_s^T w_t)}{\|\tilde{w}_s\|^2}\tilde{w}_s + \left(I - \frac{\tilde{w}_s\tilde{w}_s^T}{\|\tilde{w}_s\|^2}\right)w_t$$

using the outer loop iterate $\tilde{w}_s$. In the above decomposition, the former term represents the projection of w_t on $\tilde{w}_s$ while the latter term represents the remaining vector. Utilizing the exact gradient $\tilde{g}$ at $\tilde{w}_s$, the exact gradient at the first term can be computed as

$$\nabla f\left(\frac{(\tilde{w}_s^T w_t)}{\|\tilde{w}_s\|^2}\tilde{w}_s\right) = \frac{(\tilde{w}_s^T w_t)}{\|\tilde{w}_s\|^2}C\tilde{w}_s = \frac{(\tilde{w}_s^T w_t)}{\|\tilde{w}_s\|^2}\tilde{g}.$$

On the other hand, a stochastic sample S is used to compute a stochastic gradient at the second term as

$$\frac{1}{|S|}\sum_{l\in S} a_l a_l^T\left(I - \frac{\tilde{w}_s\tilde{w}_s^T}{\|\tilde{w}_s\|^2}\right)w_t$$

resulting in the following stochastic variance-reduced gradient g_t at w_t as

$$g_t = \frac{(\tilde{w}_s^T w_t)}{\|\tilde{w}_s\|^2}\tilde{w}_s + \frac{1}{|S|}\sum_{l\in S} a_l a_l^T\left(I - \frac{\tilde{w}_s\tilde{w}_s^T}{\|\tilde{w}_s\|^2}\right)w_t. \quad (4)$$

With the use of the stochastic variance-reduced gradient g_t, we obtain a stochastic variance-reduced heavy ball power iteration as

$$w_{t+1} \leftarrow 2\big((1-\eta)w_t + \eta g_t\big) - \beta w_{t-1} \quad (5)$$

where $\eta \in (0, 1]$ is the step size and β is the momentum parameter. Note that the deterministic update formula (3) can be obtained from (5) when the step size η is set to 1 and the exact gradient $g_t = Cw_t$ is used.

As pointed out in (Allen-Zhu, 2018), it is important to keep g_t close to the exact gradient $\nabla f(w_t)$ to obtain stochastic acceleration. An obvious way to achieve this goal is to set the mini-batch size $|S|$ large. If the size of the mini-batch is large, the variance of the stochastic part in (4) becomes

small, so that more accurate g_t can be obtained. Another way to control the variance of g_t is to decrease the step size η. If the step size η is small, the angle between the outer iterate $\tilde{w}_s$ and the inner iterate w_t can be kept close to 0. If the outer iterate $\tilde{w}_s$ is closely aligned with the inner iterate w_t, the first term dominates the second term in (4) making g_t close to the true gradient $\nabla f(w_t)$.

The mechanism of controlling the progress of the algorithm using the step size η is not present in Power+M. As a result, it fails to converge unless the mini-batch size $|S|$ is sufficiently large. To the contrary, our algorithm works with any size of the mini-batch due to the presence of the step size η. If $|S|$ is small, since the variance of the stochastic part is large, a small η needs to be chosen so that progress is made near the outer iterate $\tilde{w}_s$, making the inner iterate w_t closely aligned with the outer iterate $\tilde{w}_s$. On the other hand, if $|S|$ is large, g_t has a small variance even when it is far from the outer iterate $\tilde{w}_s$. Therefore, we can select a large η to make rapid progress.

Summarizing all the above, we obtain VR HB Power exhibited in Algorithm 1.

Algorithm 1 VR HB Power

Parameters: step size η, momentum β, mini-batch size $|S|$, epoch length m
Input: data vectors $a_i, i = 1, \dots, n$
Randomly initialize an outer iterate $\tilde{w}_0$
for $s = 0, 1, \dots$ **do**
  $\tilde{g} \leftarrow \frac{1}{n}\sum_{l=1}^{n} a_l a_l^T \tilde{w}_s$
  $w_0 \leftarrow \tilde{w}_s$
  $w_1 \leftarrow (1-\eta)w_0 + \eta\tilde{g}$
  for $t = 1, 2, \dots, m-1$ **do**
    Sample a mini-batch sample S_t uniformly at random
    $g_t \leftarrow \frac{1}{|S_t|}\sum_{l\in S_t} a_l a_l^T \left(w_t - \frac{(w_t^T w_0)w_0}{\|w_0\|^2}\right) + \frac{(w_t^T w_0)}{\|w_0\|^2}\tilde{g}$
    $w_{t+1} \leftarrow 2\big((1-\eta)w_t + \eta g_t\big) - \beta w_{t-1}$
  end for
  $\tilde{w}_s \leftarrow w_m$
end for

3. Convergence Analysis

In this section, we provide a convergence analysis for VR HB Power. Before presenting the convergence analysis, we first introduce some notations.

We define the sample covariance matrix C_t at inner iteration t and the projection matrix P to the space orthogonal to the outer iterate $w_0 = \tilde{w}_s$ as

$$C_t = \frac{1}{|S_t|}\sum_{i_t\in S_t} a_{i_t} a_{i_t}^T, \quad P = I - \frac{w_0 w_0^T}{\|w_0\|^2}. \tag{6}$$

Using (6), we can write g_t as

$$g_t = \eta C w_t + \eta(C_t - C)P w_t.$$

Since S_t is sampled uniformly at random, we have $E[C_t] = C$. Taking the expectation on the dot product of u_k and (5), we obtain

$$E[u_k^T w_{t+1}] = 2(1 - \eta + \eta\lambda_k)E[u_k^T w_t] - \beta E[u_k^T w_{t-1}].$$

Since an optimal solution of the PCA problem (1) is the first eigenvector u_1 of the covariance matrix C, the optimality gap is measured as $\sum_{k=2}^{d}(u_k^T w_t)^2/(u_1^T w_t)^2$, representing how closely w_t is aligned with u_1 (Golub & Van Loan, 2012). Note that if $w_t = u_1$, this ratio is 0. Our analysis studies it in expectation by providing a bound for $\sum_{k=2}^{d} E[(u_k^T w_t)^2]/E[(u_1^T w_t)^2]$.

Note that $1-\eta+\eta\lambda_k \geq 0$ holds for every k since $\eta \in (0, 1]$ and $\lambda_k \geq 0$. In the above, $2(1 - \eta + \eta\lambda_k)$ corresponds to $2\lambda_k$ in the dot product of u_k and (3). Taking the square of it, we define

$$\alpha_k(\eta) = 4(1 - \eta + \eta\lambda_k)^2$$

for $1 \leq k \leq d$. Analogous to the optimal momentum parameter $\beta = \lambda_2^2$ in (3), we also consider

$$\beta(\eta) = (1 - \eta + \eta\lambda_2)^2.$$

To characterize the variance of $C_t - C$, we define constant K as

$$K = \|E[(C_t - C)^2]\|.$$

Furthermore, let $P_{i,j}(K)$ denote a polynomial having the form of

$$P_{i,j}(K) = \sum_{l=i}^{j} c_l K^l$$

and let $p_t(\alpha, \beta)$ and $q_t(\alpha, \beta)$ be recurrence polynomials satisfying

$$\begin{aligned} p_t(\alpha,\beta) &= (\alpha-\beta)p_{t-1}(\alpha,\beta) - \beta(\alpha-\beta)p_{t-2}(\alpha,\beta) \\ &\quad + \beta^3 p_{t-3}(\alpha,\beta) \\ q_t(\alpha,\beta) &= (\alpha-\beta)q_{t-1}(\alpha,\beta) - \beta(\alpha-\beta)q_{t-2}(\alpha,\beta) \\ &\quad + \beta^3 q_{t-3}(\alpha,\beta) \end{aligned}$$

for $t \geq 3$ with

$$p_0(\alpha,\beta) = 1, \quad p_1(\alpha,\beta) = \frac{\alpha}{4}, \quad p_2(\alpha,\beta) = \left(\frac{\alpha}{2} - \beta\right)^2,$$
$$q_0(\alpha,\beta) = 1, \quad q_1(\alpha,\beta) = \alpha, \quad q_2(\alpha,\beta) = (\alpha-\beta)^2.$$

In the following Lemmas 3.1, 3.2, and 3.3, we consider a single epoch, which corresponds to one inner loop iteration.

Table 2. Comparison of the convergence rates of stochastic variance-reduced methods for PCA. The rates of convergence are obtained by analyzing the convergence of each algorithm using the framework presented in Section 3. For each algorithm, g and h compose the convergence rate ρ such that $\rho = g + h$.

ALGORITHM	PARAMETERS	$g(\cdot)$	$h(\cdot)$
VR-PCA	η, K	$\left[\frac{1+\eta\lambda_2}{1+\eta\lambda_1}\right]^{2m}$	$\eta^4 P_{1,m}(K)$
VR POWER+M	K	$\left[\frac{2\lambda_2^m}{\sum_{j=0}^{1}(\lambda_1+(-1)^j\sqrt{\lambda_1+\lambda_2}\sqrt{\Delta})^m}\right]^2$	$P_{1,m}(K)$
VR HB POWER	η, K	$\left[\frac{2(1-\eta+\eta\lambda_2)^m}{\sum_{j=0}^{1}\left(1-\eta+\eta\lambda_1+(-1)^j\sqrt{2-2\eta+\eta(\lambda_1+\lambda_2)}\sqrt{\eta\Delta}\right)^m}\right]^2$	$\eta^4 P_{1,m}(K)$

Lemma 3.1. *For any $0 < \eta \leq 1$, we have*

$$E[(u_k^T w_t)^2] = p_t(\alpha_k(\eta), \beta(\eta))E[(u_k^T w_0)^2] + 4\eta^2 \sum_{r=1}^{t-1} q_{t-r-1}(\alpha_k(\eta), \beta(\eta))E[w_r^T P M_k P w_r]$$

for $1 \leq k \leq d$.

In Lemma 3.1, $E[(u_k^T w_t)^2]$ is decomposed into two parts. The first term originates from $E[(u_k^T w_0)^2]$ while the second sum consisting of $E[w_t^T P M_k P w_t]$ stems from the stochastic variance of $g_t - \nabla f(w_t)$. Since the stochastic variance needs to be appropriately controlled to obtain convergence, we analyze it in the following lemma.

Lemma 3.2. *For any $0 < \eta \leq 1$, we have*

$$E[w_t^T P M_k P w_t] \leq \eta^2 P_{1,t}(K) \sum_{k=2}^{d} E\left[(u_k^T w_0)^2\right]$$

where $1 \leq k \leq d$.

Lemma 3.2 provides an upper bound for $E[w_t^T P M_k P w_t]$ which is a function of η, K, and $\sum_{k=2}^{d} E[(u_k^T w_0)^2]$. The upper bound can be interpreted in the following way. If the step size η or $\sum_{k=2}^{d} E[(u_k^T w_0)^2]$ is small, iterate updates are made near w_0, and therefore w_t is closely aligned with w_0. Since this makes g_t close to the exact gradient $\nabla f(w_t)$, the stochastic variance becomes small. On the other hand, when K is small (or $|S|$ is large), the stochastic variance of $C_t - C$ becomes small, resulting in more accurate g_t.

The next lemma establishes the error bound in expectation within a single epoch.

Lemma 3.3. *For any $0 < \eta \leq 1$, we have*

$$\frac{\sum_{k=2}^{d} E[(u_k^T w_m)^2]}{E[(u_1^T w_m)^2]} \leq \rho(\eta, K) \frac{\sum_{k=2}^{d} E[(u_k^T w_0)^2]}{E[(u_1^T w_0)^2]}$$

where

$$\rho(\eta, K) = g(\eta) + h(\eta, K)$$

and

$$g(\eta) = \frac{p_m(\alpha_2(\eta), \beta(\eta))}{p_m(\alpha_1(\eta), \beta(\eta))}, \quad h(\eta, K) = \eta^4 P_{1,m}(K).$$

Function $g(\eta)$ is a decreasing function of η on $(0, 1]$ and it equals to

$$g(\eta) = \left[\frac{(1-\eta+\eta\lambda_1+\sqrt{2-2\eta+\eta(\lambda_1+\lambda_2)}\sqrt{\eta\Delta})^m}{2(1-\eta+\eta\lambda)^m} - \frac{(1-\eta+\eta\lambda_1-\sqrt{2-2\eta+\eta(\lambda_1+\lambda_2)}\sqrt{\eta\Delta})^m}{2(1-\eta+\eta\lambda)^m}\right]^{-2}.$$

Moreover, there exists some $0 < \bar{\eta}(K)$ such that for every $\eta \in (0, \bar{\eta}(K)]$, we have

$$\rho(1, 0) \leq \rho(\eta, K) < 1.$$

While Lemmas 3.1, 3.2 and 3.3 deal with a single epoch, the next result establishes the convergence of the overall algorithm.

Theorem 3.4. *Suppose we execute Algorithm 1 for s epochs starting from an initial unit vector $\tilde{w}_0$ such that $u_1^T \tilde{w}_0 \neq 0$. There exists some $0 < \bar{\eta}(K)$ such that for every $\eta \in (0, \bar{\eta}(K)]$, we have*

$$\frac{\sum_{k=2}^{d} E[(u_k^T \tilde{w}_s)^2]}{E[(u_1^T \tilde{w}_s)^2]} \leq \rho(\eta, K)^s \left(\frac{1-(u_1^T \tilde{w}_0)^2}{(u_1^T \tilde{w}_0)^2}\right) \quad (7)$$

where $0 < \rho(\eta, K) < 1$.

Lemma 3.3 provides a convergence rate for a single epoch where the rate of convergence $\rho(\eta, K)$ consists of the expected rate $g(\eta)$ and the additional variance term $h(\eta, K)$ arising from stochastic errors. Owing to the momentum term in (5), $g(\eta)$ depends inversely on the square root of the eigen-gap Δ, making it appealing when Δ is small. In order to benefit from the $\sqrt{\Delta}$ term, η should not be too small. However, as increasing η also enlarges $h(\eta, K)$, K must be

controlled in order to achieve stochastic acceleration. Since decreasing K does not affect $g(\eta)$ but reduces the stochastic variance term $h(\eta, K)$, a large η can be tolerated with a small K. However, even if K is large, corresponding to a small mini-batch size, we can still obtain convergence by choosing a sufficiently small η. Since the stochastic variance term $h(\eta, K)$ vanishes very quickly as η approaches 0, we can always make $\rho(\eta, K)$ smaller than 1 by selecting a sufficiently small η as showed in Lemma 3.3. On the other hand, since $g(\eta)$ is a decreasing function of η, $\rho(\eta, K)$ is lower bounded by $\rho(1, 0)$ for any $\eta \in (0, 1]$.

Compared to VR-PCA and VR Power+M, VR HB Power is favorable since it works with any size of the mini-batch, yet enjoys acceleration when the mini-batch size is sufficiently large. Unlike VR Power+M, VR-PCA works with any size of the mini-batch size but does not accelerate when the size of the mini-batch is large since its rate does not depend inversely on the square root of Δ. On the other hand, although the convergence rate of VR Power+M has an inverse dependency on the square root of Δ, it lacks the measure to control the progress of the algorithm, making it fail to converge when the mini-batch size is not sufficiently large. Moreover, even when the mini-batch size is large, its performance cannot be superior to VR HB Power since it is a special case of VR HB Power with η being 1. By appropriately choosing the step size η, VR HB Power performs no worse than VR Power+M in all cases. Table 2 summarizes the convergence rates of these algorithms. Note that our convergence rate for VR Power+M is established by setting $\eta = 1$ in the rate for VR HB Power. The rate for VR-PCA is derived by using the recurrence polynomials without the momentum term.

Theorem 3.4 provides a convergence result for the entire algorithm. Specifically, it establishes the global linear convergence of the algorithm where the ratio of the expectation of the components orthogonal to u_1 to the expectation of the component aligned with u_1 goes to zero at a rate of $\rho(\eta, K) < 1$. Note that the rate of convergence $\rho(\eta, K)$ does not depend on the epoch but is kept the same across the epochs. Although VR-PCA and VR Power+M practically converge regardless of the initial iterate $\tilde{w}_0$, there has been no analysis proving their global convergence. However, by following the techniques developed in this section, their global convergence is proved by our framework for any initial iterate $\tilde{w}_0$ and in expectation (as opposed to the probabilistic statements in (Shamir, 2015) and (Xu et al., 2018)).

Note that the condition $\eta \in (0, \bar{\eta}(K)]$ and $\rho = \rho(\eta, K)$ can be changed to there exists $0 < \eta_2(K)$ such that for every $\eta_1 \in (0, \eta_2(K))$ and $\eta \in [\eta_1, \eta_2(K)]$, inequality (7) holds for $\rho = \rho(\eta_1, K)$. This can be easily established based on the proof.

4. Practical Considerations

In this section, we discuss some practical considerations of the algorithm. First, to make the algorithm numerically stable, we consider

$$w_t \leftarrow w_t/\|w_{t+1}\|_2, \quad w_{t+1} \leftarrow w_{t+1}/\|w_{t+1}\|_2$$

at the end of the inner loop as introduced in (Xu et al., 2018). Since the above scaling scheme does not impact the sample path of $w_t/\|w_t\|$, the same result can be obtained with numerical stability.

Another important issue is the estimation of λ_2, which is involved in determining the value of the momentum parameter. Using the two-loop structure of the algorithm and the fact that the gradients of the outer-loop iterates $\tilde{w}_s$ are exactly computed, we estimate λ_2 using the two consecutive outer-loop iterates $\tilde{w}_{s-1}$ and $\tilde{w}_s$ at a regular interval.

Using the Rayleigh quotient and the second eigenvector u_2 of the covariance matrix C, the second eigenvalue λ_2 can be expressed as

$$\lambda_2 = \frac{u_2^T C u_2}{u_2^T u_2}. \tag{8}$$

In deterministic power iteration and its variants, an outer-iterate $\tilde{w}_s$ first approaches the subspace spanned by u_1 and u_2 before converging to u_1. After a number of outer-iterations, vector $\tilde{w}_s$ can be approximated by a linear combination of u_1 and u_2 and the component of u_1 becomes dominant as the iterations proceed.

Based on this observation, we estimate u_2 using two consecutive outer-loop iterates $\tilde{w}_s$ and $\tilde{w}_{s-1}$ as

$$\hat{u}_{2,s} = \tilde{w}_{s-1} - (\tilde{w}_{s-1}^T \tilde{w}_s)\tilde{w}_s. \tag{9}$$

The idea of the above estimation is to project $\tilde{w}_{s-1}$ to the space orthogonal to $\tilde{w}_s$. If $\tilde{w}_s \approx u_1$ and $\tilde{w}_{s-1} \approx \alpha_1 u_1 + \alpha_2 u_2$ for some $\alpha_1, \alpha_2 (\neq 0)$, we have $\hat{u}_{2,s} \approx u_2$. By substituting u_2 with $\hat{u}_{2,s}$ in (8), we obtain

$$\hat{\lambda}_{2,s} = \frac{\tilde{w}_{s-1}^T C \tilde{w}_{s-1} - 2\theta_s \tilde{w}_s^T C \tilde{w}_{s-1} + \theta_s^2 \tilde{w}_s^T C \tilde{w}_s}{1 - \theta_s^2} \tag{10}$$

where

$$\theta_s = \tilde{w}_{s-1}^T \tilde{w}_s.$$

While two matrix-vector multiplications, $C\tilde{w}_{s-1}$ and $C\tilde{w}_s$, are involved in computing (10), they incur no extra computation since they are the exact gradients of $\tilde{w}_{s-1}$ and $\tilde{w}_s$, which are computed regardless of the estimation. As a result, we can obtain $\hat{\lambda}_2$ by only computing inner products. This update is repeated at the start of each outer-loop iteration after computing $\tilde{g}$ followed by setting the momentum parameter β_s at outer iteration s as

$$\beta_s = (1 - \eta + \eta\hat{\lambda}_{2,s})^2.$$

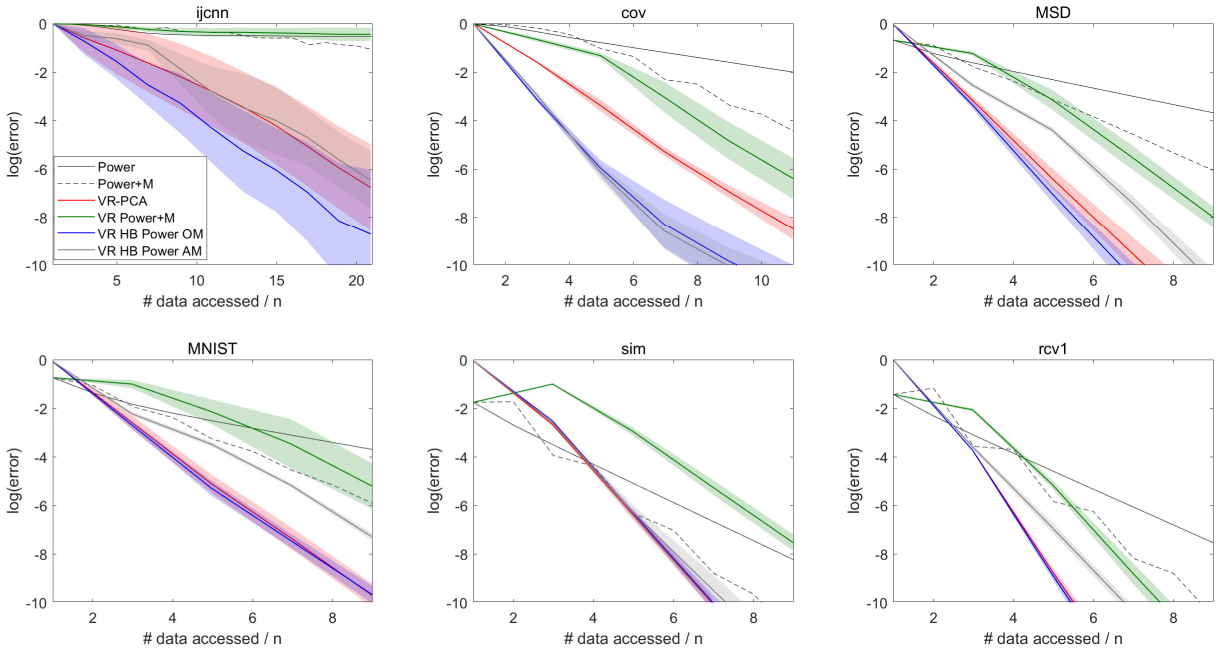

Figure 1. Experimental Results (small-batch)

5. Numerical Experiments

In this section, we numerically compare the performance of VR HB Power with that of (i) Power, (ii) Power+M, (iii) VR-PCA, and (iv) VR Power+M on finding the first eigenvector u_1 of covariance matrix C constructed by data vectors $a_i, i = 1, \ldots, n$ using real world datasets. We include variants of power iteration and other algorithms are excluded due to the complexity of implementation.

5.1. Datasets

The datasets include ijcnn (Prokhorov, 2001), covertype (Blackard & Dean, 1999), YearPredictionMSD (Bertin-Mahieux et al., 2011), MNIST (LeCun et al., 1998), real-sim and rcv1 (Lewis et al., 2004) as summarized in Tabel 3. All of them are obtained either from the UCI repository (Dheeru & Karra Taniskidou, 2017) or the LIBSVM library (Chang & Lin, 2011) and they are carefully chosen to incorporate a variety of datasets in terms of size and eigen-gap. The first four datasets are standardized with a mean of zero and standard deviation of one while the last two datasets are scaled to range between 0 and 1 to preserve their sparsity.

Table 3. Datasets

DATASET	n	d	SPARSITY	λ_2/λ_1
ICJNN(TEST)	91,701	22	59.09 %	0.9921
COV	581,012	54	22.00 %	0.7894
MSD	463,715	90	100.00 %	0.6776
MNIST	70,000	764	1.96 %	0.7167
SIM	72,309	20,958	0.24 %	0.4053
RCV1	804,414	47,236	0.16 %	0.4289

5.2. Settings

Since we consider the mini-batch setting of variance reduction methods (VR-PCA, VR Power+M, VR HB Power) which have the two-loop structure, it is necessary to choose the epoch length m and the mini-batch size $|S|$. For the choice of m and $|S|$, it is common to select m and $|S|$ such that $m \cdot |S| = n$. Following this principle, we consider $|S| = 1\% \cdot n$ and $m = 100$ for the small-batch case and $|S| = 5\% \cdot n$ and $m = 20$ for the large-batch case. For the momentum parameter β in VR HB Power, Power+M, and VR Power+M, we utilize the true value of λ_2 for Power+M and VR Power+M and consider both the true value of λ_2 (VR HB Power OM) and the adaptive estimation procedure of λ_2 (VR HB Power AM) presented in Section 4 for VR HB Power. For numerical stability, the scaling scheme from Section 4 is also used for VR HB Power and VR Power+M. To find the best performance, the step sizes η in VR HB Power OM and VR-PCA are chosen using grid search and the step size in VR HB Power AM is set to that of VR HB Power OM.

5.3. Results

Figure 1 and Figure 2 display the experimental results for the small and large batch cases, respectively. In the figures,

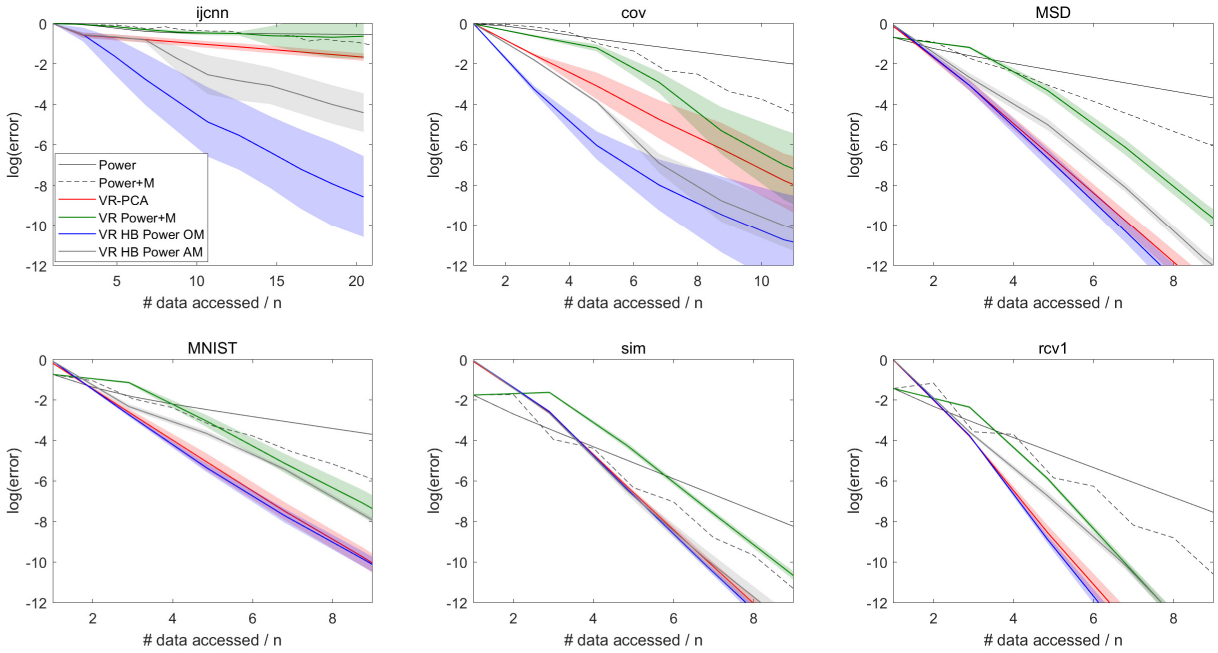

Figure 2. Experimental Results (large-batch)

the x-axis represents the number of data points accessed over the number of total data points and the y-axis represents the error gap, $1-(\tilde{w}_s^T u_1)^2$, in the log-scale. For each case, the stochastic algorithms are repeated 10 times by varying the random seed. The lines represent their means and the values within one standard deviation away from the means are shaded.

As shown in the figures, with the true value of λ_2, VR HB Power OM consistently outperforms the other algorithms in both cases. Particularly, it surpasses the other algorithms by a large margin when λ_2/λ_1 is close to 1 as seen in the cases of ijcnn and cov. If λ_2/λ_1 is not large, its performance is similar to that of VR-PCA in the small-batch setting and slightly better in the large-batch setting as shown in the cases of MNIST, sim, and rcv1. While VR-PCA is competitive to VR HB Power OM when λ_2/λ_1 and $|S|$ are small, VR Power+M always falls behind VR-PCA and VR HB Power OM. Moreover, it fails to converge in the small batch setting of ijcnn and is sometimes inferior to Power and Power+M.

On other hand, VR HB Power AM does not have the same performance as VR HB Power OM. However, as $\hat{\lambda}_{2,s}$ approaches λ_2, it asymptotically attains the same rate of convergence. If λ_2/λ_1 is close to 1, slow progress at the start can be compensated by asymptotic performance as $\hat{\lambda}_{2,s}$ approaches λ_2. Also, the estimation of λ_2 becomes stable if the batch size is large. Therefore, it exhibits superior performance to VR-PCA and VR Power+M when λ_2/λ_1 is close to 1 and the size of the mini-batch is large as seen in the cases of ijcnn and cov in Figure 2.

6. Conclusion

In this paper, we present a stochastic heavy ball power iteration algorithm for solving PCA and present a convergence analysis for it. By incorporating a step size, the presented algorithm works with any size of the mini-batch and attains acceleration if the mini-batch size is large, making it attractive in parallel settings. In the convergence analysis, we show that our algorithm attains global linear convergence to the first eigenvector of the covarince matrix in expectation. This result is analogous to those of stochastic variance-reduced gradient methods for convex optimization but has been never shown for previous stochastic power iteration algorithms since it requires completely different techniques. We stress that our analysis is the first such analysis and its framework can be applied to analyze previous stochastic power iteration algorithms, showing their global linear convergence in expectation. The experimental results show that if λ_2 is known, our algorithm consistently outperforms other algorithms, especially when the eigen-gap is small. Even if λ_2 is unknown, our algorithm can be run with the adaptive estimation procedure of λ_2. The numerical experiments exhibit that it still outperforms the previous stochastic algorithms if the eigen-gap is small and the size of the mini-batch is large.

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(221):1–51, 2018.

Allen-Zhu, Z. and Li, Y. LazySVD: Even faster SVD decomposition yet without agonizing pain. In *Advances in Neural Information Processing Systems*, pp. 974–982, 2016.

Balcan, M.-F., Du, S. S., Wang, Y., and Yu, A. W. An improved gap-dependency analysis of the noisy power method. In *Conference on Learning Theory*, pp. 284–309, 2016.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval*, 2011.

Blackard, J. A. and Dean, D. J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

Chang, C.-C. and Lin, C.-J. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.

Dheeru, D. and Karra Taniskidou, E. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Garber, D. and Hazan, E. Fast and simple PCA via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

Garber, D., Hazan, E., Jin, C., Kakade, S. M., Musco, C., Netrapalli, P., and Sidford, A. Faster eigenvector computation via shift-and-invert preconditioning. In *International Conference on Machine Learning*, pp. 2626–2634, 2016.

Golub, G. H. and Van Loan, C. F. *Matrix computations*, volume 3. JHU Press, 2012.

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

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lei, Q., Zhong, K., and Dhillon, I. S. Coordinate-wise power method. In *Advances in Neural Information Processing Systems*, pp. 2064–2072, 2016.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

Oja, E. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Prokhorov, D. Ijcnn 2001 neural network competition. *Slide Presentation in International Joint Conference on Neural Networks*, 1:97, 2001.

Shamir, O. A stochastic PCA and SVD algorithm with an exponential convergence rate. In *International Conference on Machine Learning*, pp. 144–152, 2015.

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

Wang, J., Wang, W., Garber, D., and Srebro, N. Efficient coordinate-wise leading eigenvector computation. In *Algorithmic Learning Theory*, pp. 806–820, 2018.

Xu, P., He, B., De Sa, C., Mitliagkas, I., and Re, C. Accelerated stochastic power iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 58–67, 2018.

Xu, Z. Gradient descent meets shift-and-invert preconditioning for eigenvector computation. In *Advances in Neural Information Processing Systems*, pp. 2830–2839, 2018.

A. Supplimentary Material

A.1. Main Results

Proof of Lemma 3.1. From

$$\begin{aligned} w_1 &= (1-\eta)w_0 + \eta\tilde{g} \\ &= (1-\eta)w_0 + \eta C w_0, \end{aligned}$$

we have

$$\begin{aligned} u_k^T w_1 &= (1-\eta)u_k^T w_0 + \eta u_k^T C w_0 \\ &= (1-\eta)u_k^T w_0 + \eta\lambda_k u_k^T w_0 \\ &= (1-\eta+\eta\lambda_k)u_k^T w_0. \end{aligned} \tag{11}$$

Taking the expectation of the square of (11), we obtain

$$E[(u_k^T w_1)^2] = (1-\eta+\eta\lambda_k)^2 E[(u_k^T w_0)^2] = \frac{\alpha_k(\eta)}{4} E[(u_k^T w_0)^2]. \tag{12}$$

Next, from (5), we have

$$\begin{aligned} w_{t+1} &= 2\bigg((1-\eta)w_t + \eta\frac{1}{|S_t|}\sum_{i_t\in S_t} a_{i_t}a_{i_t}^T\Big(w_t - \frac{(w_t^T w_0)}{\|w_0\|^2}w_0\Big) + \frac{(w_t^T w_0)}{\|w_0\|^2}\tilde{g}\bigg) - \beta(\eta)w_{t-1} \\ &= 2\bigg((1-\eta)w_t + \eta\frac{1}{|S_t|}\sum_{i_t\in S_t} a_{i_t}a_{i_t}^T\Big(I - \frac{w_0 w_0^T}{\|w_0\|^2}\Big)w_t + C\frac{w_0 w_0^T}{\|w_0\|^2}w_t\bigg) - \beta(\eta)w_{t-1} \\ &= 2\bigg((1-\eta)w_t + \eta C w_t + \eta\frac{1}{|S_t|}\sum_{i_t\in S_t}(a_{i_t}a_{i_t}^T - C)\Big(I - \frac{w_0 w_0^T}{\|w_0\|^2}\Big)w_t\bigg) - \beta(\eta)w_{t-1} \\ &= 2\big((1-\eta)w_t + \eta C w_t + \eta(C_t - C)Pw_t\big) - \beta(\eta)w_{t-1}, \end{aligned} \tag{13}$$

leading to

$$u_k^T w_{t+1} = 2\big((1-\eta+\eta\lambda_k)u_k^T w_t + \eta u_k^T(C_t - C)Pw_t\big) - \beta(\eta)u_k^T w_{t-1}. \tag{14}$$

Taking the square of (14), we have

$$\begin{aligned} (u_k^T w_{t+1})^2 = {} & 4(1-\eta+\eta\lambda_k)^2(u_k^T w_t)^2 + 4\eta^2 w_t^T P(C_t - C)u_k u_k^T(C_t - C)Pw_t + (\beta(\eta))^2(u_k^T w_{t-1})^2 \\ & + 8\eta(1-\eta+\eta\lambda_k)u_k^T w_t u_k^T(C_t - C)Pw_t - 4(1-\eta+\eta\lambda_k)\beta(\eta)u_k^T w_t u_k^T w_{t-1} \\ & - 4\eta\beta(\eta)u_k^T(C_t - C)Pw_t u_k^T w_{t-1}. \end{aligned} \tag{15}$$

Since S_t is sampled uniformly at random, C_t is independent of $S_1, \ldots, S_{t-1}$ and identically distributed with $E[C_t] = C$. Therefore,

$$E[u_k^T w_t(C_t - C)Pw_t] = E[E[u_k^T w_t u_k^T(C_t - C)Pw_t | w_0, S_1, \ldots, S_{t-1}]] = E[u_k^T w_t u_k^T E[C_t - C]Pw_t] = 0.$$

Similarly, we have

$$E[u_k^T(C_t - C)Pw_t u_k^T w_{t-1}] = 0. \tag{16}$$

As a result, we obtain

$$\begin{aligned} E[(u_k^T w_{t+1})^2] = {} & \alpha_k(\eta)E[(u_k^T w_t)^2] - 2\sqrt{\alpha_k(\eta)}\beta(\eta)E[(u_k^T w_t)(u_k^T w_{t-1})] + (\beta(\eta))^2 E[(u_k^T w_{t-1})^2] \\ & + 4\eta^2 E[w_t^T P M_k P w_t]. \end{aligned} \tag{17}$$

Using (11) and (12) for $t = 1$ in (17), we have

$$E[(u_k^T w_2)^2] = \left(\frac{\alpha_k(\eta)}{2} - \beta(\eta)\right)^2 E[(u_k^T w_0)^2] + 4\eta^2 E[w_1^T P M_k P w_1]. \tag{18}$$

Moreover, by using (14) with $t-1$, multiplying it with $u_k^T w_{t-1}$, taking expectation and using (16) with w_t being w_{t-1} (which can be derived in the same way as (16)) , we have

$$E[(u_k^T w_t)(u_k^T w_{t-1})] = \sqrt{\alpha_k(\eta)} E[(u_k^T w_{t-1})^2] - \beta(\eta) E[(u_k^T w_{t-1})(u_k^T w_{t-2})]. \tag{19}$$

Using (19), we can further write (17) as

$$\begin{aligned} E[(u_k^T w_{t+1})^2] = \alpha_k(\eta) E[(u_k w_t)^2] - \beta(\eta)(2\alpha_k(\eta) - \beta(\eta)) E[(u_k^T w_{t-1})^2] \\ + 2\sqrt{\alpha_k(\eta)}(\beta(\eta))^2 E[(u_k^T w_{t-1})(u_k^T w_{t-2})] + 4\eta^2 E[w_t^T P M_k P w_t]. \end{aligned} \tag{20}$$

With $t-1$ in (17), we have

$$\begin{aligned} E[(u_k^T w_t)^2] = \alpha_k(\eta) E[(u_k^T w_{t-1})^2] - 2\sqrt{\alpha_k(\eta)}\beta(\eta) E[(u_k^T w_{t-1})(u_k^T w_{t-2})] + (\beta(\eta))^2 E[(u_k^T w_{t-2})^2] \\ + 4\eta^2 E[w_{t-1}^T P M_k P w_{t-1}]. \end{aligned} \tag{21}$$

Adding (21) multiplied by $\beta(\eta)$ to (20), we obtain

$$\begin{aligned} E[(u_k^T w_{t+1})^2] = (\alpha_k(\eta) - \beta(\eta)) E[(u_k^T w_t)^2] - \beta(\eta)(\alpha_k(\eta) - \beta(\eta)) E[(u_k^T w_{t-1})^2] + (\beta(\eta))^3 E[(u_k^T w_{t-2})^2] \\ + 4\eta^2 E[w_t^T P M_k P w_t] + 4\eta^2 \beta(\eta) E[w_{t-1}^T P M_k P w_{t-1}]. \end{aligned} \tag{22}$$

With $t-1$ in (22), we finally have

$$\begin{aligned} E[(u_k^T w_t)^2] = (\alpha_k(\eta) - \beta(\eta)) E[(u_k^T w_{t-1})^2] - \beta(\eta)(\alpha_k(\eta) - \beta(\eta)) E[(u_k^T w_{t-2})^2] + (\beta(\eta))^3 E[(u_k^T w_{t-3})^2] \\ + 4\eta^2 E[w_{t-1}^T P M_k P w_{t-1}] + 4\eta^2 \beta(\eta) E[w_{t-2}^T P M_k P w_{t-2}] \end{aligned} \tag{23}$$

for $t \geq 3$.

Using Lemma A.4 for $E[(u_k^T w_t)^2]$ defined by (12), (18), and (23) with

$$\alpha = \alpha_k(\eta), \quad \beta = \beta(\eta), \quad L_0 = E[(u_k^T w_0)^2], \quad L_t = 4\eta^2 E[w_t^T P M_k P w_t],$$

we have

$$E[(u_k^T w_t)^2] = p_t(\alpha_k(\eta), \beta(\eta)) E[(u_k^T w_0)^2] + 4\eta^2 \sum_{r=1}^{t-1} q_{t-r-1}(\alpha_k(\eta), \beta(\eta)) E[w_r^T P M_k P w_r].$$

□

Proof of Lemma 3.2. From Lemma A.1, we have

$$E[w_t^T P M_k P w_t] \leq E[\|M_k\|] E[\|P w_t\|^2] \leq K E[\|P w_t\|^2]. \tag{24}$$

By the definition of P in (6), we have

$$E[\|P w_0\|^2] = E\left[\left\|\left(I - \frac{w_0 w_0^T}{\|w_0\|^2}\right) w_0\right\|^2\right] = E[\|w_0 - w_0\|^2] = 0. \tag{25}$$

Using Lemma A.3, we obtain

$$\begin{aligned}
E[\|Pw_1\|^2] &= E\left[\left\|\left(I - \frac{w_0 w_0^T}{\|w_0\|^2}\right)\left(\eta w_0 + \eta C w_0\right)\right\|^2\right] \\
&= E\left[\left\|\eta C w_0 - \eta \frac{w_0 w_0^T C w_0}{\|w_0\|^2}\right\|^2\right] \\
&= \eta^2 E\left[\|w_0\|^2\left(\frac{w_0^T C^2 w_0}{\|w_0\|^2} - \frac{(w_0^T C w_0)^2}{\|w_0\|^4}\right)\right] \\
&\le 2\eta^2 E\left[\|w_0\|^2\left(\lambda_1^2 - \lambda_1^2 \frac{(u_1^T w_0)^2}{\|w_0\|^2}\right)\right] \\
&= 2\eta^2 \lambda_1^2 \sum_{k=2}^{d} E\big[(u_k^T w_0)^2\big] \qquad (26)
\end{aligned}$$

where we have used the fact that $u_1, \ldots, u_d$ form an orthonormal basis for the last equality.

For $t \ge 2$, we consider

$$Pw_t = 2\big(P((1-\eta)I + \eta C)w_{t-1} + \eta P(C_t - C)Pw_{t-1}\big) - \beta(\eta)Pw_{t-2}. \qquad (27)$$

Taking the squared norm of (27), we have

$$\begin{aligned}
\|Pw_t\|^2 = \|2P((1-\eta)I + \eta C)w_{t-1} - \beta(\eta)Pw_{t-2}\|^2 + 4\eta^2\|P(C_t - C)Pw_{t-1}\|^2 \\
+ 4\eta(2P((1-\eta)I + \eta C)w_{t-1} - \beta(\eta)Pw_{t-2})^T P(C_t - C)Pw_{t-1}. \qquad (28)
\end{aligned}$$

Similarly to (16), we have

$$E[(P((1-\eta)I + \eta C)w_{t-1} - \beta(\eta)Pw_{t-2})^T P(C_t - C)Pw_{t-1}] = 0,$$

resulting in

$$E[\|Pw_t\|^2] = E[\|2P((1-\eta)I + \eta C)w_{t-1} - \beta(\eta)Pw_{t-2}\|^2] + 4\eta^2 E[\|P(C_t - C)Pw_{t-1}\|^2]. \qquad (29)$$

By the triangle inequality and $(a+b)^2 \le 2(a^2+b^2)$ for all a, b, we have

$$\begin{aligned}
E[\|2P((1-\eta)I + \eta C)w_{t-1} - \beta(\eta)Pw_{t-2}\|^2] &\le E[(\|2P((1-\eta)I + \eta C)w_{t-1}\| + \beta(\eta)\|Pw_{t-2}\|)^2] \\
&\le 2E[\|2P((1-\eta)I + \eta C)w_{t-1}\|^2] + 2(\beta(\eta))^2 E[\|Pw_{t-2}\|^2]. \qquad (30)
\end{aligned}$$

Using the definition of the spectral norm, we further have

$$\begin{aligned}
E[\|2P((1-\eta)I + \eta C)w_{t-1}\|^2] &= 4E[\|((1-\eta)I + \eta C)Pw_{t-1} + \eta(PC - CP)w_{t-1}\|^2] \\
&\le 8E[\|((1-\eta)I + \eta C)Pw_{t-1}\|^2] + 8E[\|\eta(PC - CP)w_{t-1}\|^2] \\
&\le 8E[\|(1-\eta)I + \eta C\|^2\|Pw_{t-1}\|^2] + 8\eta^2 E[\|PC - CP\|^2\|w_{t-1}\|^2] \\
&\le 8(1 - \eta + \eta\lambda_1)^2 E[\|Pw_{t-1}\|^2] + 8\eta^2 E[\|PC - CP\|^2\|w_{t-1}\|^2]. \qquad (31)
\end{aligned}$$

From Lemmas A.2 and A.3, we have

$$\begin{aligned}
E[\|PC - CP\|^2\|w_{t-1}\|^2] &\le E\left[\left(\frac{w_0^T C^2 w_0}{\|w_0\|^2} - \frac{(w_0^T C w_0)^2}{\|w_0\|^4}\right)\|w_{t-1}\|^2\right] \\
&\le E\left[2\left(\lambda_1^2 - \lambda_1^2 \frac{(u_1^T w_0)^2}{\|w_0\|^2}\right)\|w_{t-1}\|^2\right] \\
&= E\left[2\lambda_1^2\left(\|w_0\|^2 - (u_1^T w_0)^2\right)\frac{\|w_{t-1}\|^2}{\|w_0\|^2}\right] \\
&= 2\lambda_1^2 E\left[\frac{\|w_{t-1}\|^2}{\|w_0\|^2}\sum_{k=2}^{d}(u_k^T w_0)^2\right] \qquad (32)
\end{aligned}$$

where we again have used the fact that $u_1, \ldots, u_d$ form an orthonormal basis for the last equality.

On the other hand, by observing that $P^2 = P$, we have

$$E[\|P(C_t - C)Pw_{t-1}\|^2] = E[w_{t-1}^T P M_P P w_{t-1}] \leq E[\|M_P\|]E[\|Pw_{t-1}\|^2] \leq KE[\|Pw_{t-1}\|^2]. \tag{33}$$

Using (30), (31), (32), (33) in (29), we obtain

$$E[\|Pw_t\|^2] \leq (8(1-\eta+\eta\lambda_1)^2 + 4\eta^2 K)E[\|Pw_{t-1}\|^2] + 2(\beta(\eta))^2 E[\|Pw_{t-2}\|^2] + 16\eta^2\lambda_1^2 E\left[\frac{\|w_{t-1}\|^2}{\|w_0\|^2}\sum_{k=2}^{d}(u_k^T w_0)^2\right]. \tag{34}$$

Using Lemma A.5 for $E[\|Pw_t\|^2]$ defined by (25), (26), and (34) with

$$\alpha = 8(1-\eta+\eta\lambda_1)^2 + 4\eta^2 K, \quad \beta = 2(\beta(\eta))^2, \quad L_t = 16\eta^2\lambda_1^2 E\left[\frac{\|w_t\|^2}{\|w_0\|^2}\sum_{k=2}^{d}(u_k^T w_0)^2\right],$$

we obtain

$$E[\|Pw_t\|^2] \leq 16\eta^2\lambda_1^2 \sum_{l=0}^{t-1} r_{t-l-1}\big(8(1-\eta+\eta\lambda_1)^2 + 4\eta^2 K, 2(\beta(\eta))^2\big) E\left[\frac{\|w_l\|^2}{\|w_0\|^2}\sum_{k=2}^{d}(u_k^T w_0)^2\right]. \tag{35}$$

For $0 \leq t-l-1 \leq t-1$, we have

$$r_{t-l-1}\big(8(1-\eta+\eta\lambda_1)^2 + 4\eta^2 K, 2(\beta(\eta))^2\big) \leq P_{0,t-l-1}(K) \tag{36}$$

where we use the fact that the power of η is bounded by 1 since $\eta \in (0, 1]$.

Moreover, letting $\bar{c}$ be defined as

$$\bar{c} = \max_{i=1,\ldots,n} \frac{\max_{|S_t|=i}\|C_l - C\|^2}{\|E[(C_l - C)^2]\|},$$

we have

$$\max\|C_l - C\|^2 \leq \bar{c}K.$$

Since

$$\left\| \begin{bmatrix} 2\big((1-\eta)(I+\eta C) + \eta(C_l - C)P\big) & -\beta(\eta)I \\ I & 0 \end{bmatrix} \right\|^2 \leq 2\left\| \begin{bmatrix} 2(1-\eta)(I+\eta C) & -\beta(\eta)I \\ I & 0 \end{bmatrix} \right\|^2 + 8\eta^2 \left\| \begin{bmatrix} (C_l - C)P & 0 \\ 0 & 0 \end{bmatrix} \right\|^2$$

and

$$\left\| \begin{bmatrix} (C_l - C)P & 0 \\ 0 & 0 \end{bmatrix} \right\|^2 = \big\|(C_l - C)P\big\|^2 \leq \big\|C_l - C\big\|^2 \leq \bar{c}K,$$

we have

$$\left\| \begin{bmatrix} 2\big((1-\eta)(I+\eta C) + \eta(C_l - C)P\big) & -\beta(\eta)I \\ I & 0 \end{bmatrix} \right\|^2 \leq P_{0,1}(K).$$

Taking the the squared Euclidean norm on the both sides of

$$\begin{bmatrix} w_l \\ w_{l-1} \end{bmatrix} = \begin{bmatrix} 2\big((1-\eta)(I+\eta C) + \eta(C_{ld-1} - C)P\big) & -\beta(\eta)I \\ I & 0 \end{bmatrix} \cdots \begin{bmatrix} 2\big((1-\eta)(I+\eta C) + \eta(C_0 - C)P\big) & -\beta(\eta)I \\ I & 0 \end{bmatrix} \begin{bmatrix} w_0 \\ 0 \end{bmatrix},$$

we have

$$\|w_l\|^2 \leq \|(w_l, w_{l-1})\|^2 \leq P_{0,l}(K)\|w_0\|^2. \tag{37}$$

Using (36) and (37) for (35), we have

$$E[\|Pw_t\|^2] \leq 16\eta^2\lambda_1^2 \sum_{l=0}^{t-1} P_{0,t-l-1}(K)P_{0,l}(K)E\bigg[\sum_{k=2}^{d}(u_k^T w_0)^2\bigg] \leq \eta^2 P_{0,t-1}(K)E\bigg[\sum_{k=2}^{d}(u_k^T w_0)^2\bigg]. \tag{38}$$

Plugging (38) into (24), we finally obtain

$$E[w_t^T P M_k P w_t] \leq \eta^2 P_{1,t}(K)\sum_{k=2}^{d} E\big[(u_k^T w_0)^2\big].$$

□

Proof of Lemma 3.3. From Lemma 3.1, we have

$$E[(u_1^T w_m)^2] = p_m(\alpha_1(\eta), \beta(\eta))E[(u_1^T w_0)^2] + 4\eta^2 \sum_{t=1}^{m-1} q_{m-t-1}(\alpha_1(\eta), \beta(\eta))E[w_t^T P M_k P w_t]. \tag{39}$$

Using

$$E[w_t^T P M_k P w_t] = E[w_t^T P(C_t - C)u_k u_k^T (C_t - C)P w_t] = E[(w_t^T P(C_t - C)u_k)^2] \geq 0$$

and (57) in Lemma A.4 since $\alpha_1(\eta) > 4\beta(\eta)$, we have

$$E[(u_1^T w_m)^2] \geq p_m(\alpha_1(\eta), \beta(\eta))E[(u_1^T w_0)^2]. \tag{40}$$

On other hand, for $2 \leq k \leq d$, using Lemma 3.1 and (58) in Lemma A.4 since $\alpha_k(\eta) \leq \alpha_2(\eta) = 4\beta(\eta)$, we have

$$\begin{aligned} E[(u_k^T w_m)^2] &= p_m(\alpha_k(\eta), \beta(\eta))E[(u_k^T w_0)^2] + 4\eta^2 \sum_{t=1}^{m-1} q_{m-t-1}(\alpha_k(\eta), \beta(\eta))E[w_t^T P M_k P w_t] \\ &\leq p_m(\alpha_2(\eta), \beta(\eta))E[(u_k^T w_0)^2] + 4\eta^4 \sum_{t=1}^{m-1} q_{m-t-1}(\alpha_2(\eta), \beta(\eta))E[w_t^T P M_k P w_t]. \end{aligned}$$

Moreover, using Lemma 3.2, we further have

$$E[(u_k^T w_m)^2] \leq p_m(\alpha_2(\eta), \beta(\eta))E[(u_k^T w_0)^2] + 4\eta^4 \sum_{t=1}^{m-1} q_{m-t-1}(\alpha_2(\eta), \beta(\eta))P_{1,t}(K)\sum_{k=2}^{d} E[(u_k^T w_0)^2]. \tag{41}$$

Combining (40) and (41), we obtain

$$\frac{\sum_{k=2}^{d} E[(u_k^T w_m)^2]}{E[(u_1^T w_m)^2]} \leq \left(\frac{p_m(\alpha_2(\eta), \beta(\eta))}{p_m(\alpha_1(\eta), \beta(\eta))} + \frac{4\eta^4(d-1)\sum_{t=1}^{m-1} q_{m-t-1}(\alpha_2(\eta), \beta(\eta))P_{1,t}(K)}{p_m(\alpha_1(\eta), \beta(\eta))}\right)\frac{\sum_{k=2}^{d} E[(u_k^T w_0)^2]}{E[(u_1^T w_0)^2]}.$$

Using (55) and (56) in Lemma A.4 since $\alpha_1(\eta) > 4\beta(\eta)$, we have

$$q_{m-t-1}(\alpha_2(\eta), \beta(\eta)) = (m-t)^2(\beta(\eta))^{m-t-1}, \quad (\beta(\eta))^m = p_m(\alpha_2(\eta), \beta(\eta)) < p_m(\alpha_1(\eta), \beta(\eta)).$$

Therefore, we obtain

$$\frac{\sum_{k=2}^{d} E[(u_k^T w_m)^2]}{E[(u_1^T w_m)^2]} \leq \left(\frac{p_m(\alpha_2(\eta), \beta(\eta))}{p_m(\alpha_1(\eta), \beta(\eta))} + \eta^4 P_{1,m}(K)\right)\frac{\sum_{k=2}^{d} E[(u_k^T w_0)^2]}{E[(u_1^T w_0)^2]} \tag{42}$$

where we used the fact that $\beta(n) > \epsilon$ for a fixed small enough ϵ and any $\eta \geq 0$.

Next, let

$$\rho(\eta, K) = g(\eta) + c'\eta^4 K, \quad g(\eta) = \frac{p_m(\alpha_2(\eta), \beta(\eta))}{p_m(\alpha_1(\eta), \beta(\eta))}.$$

Using (55) and (56) in Lemma A.4, we have

$$\begin{aligned} g(\eta) &= \frac{4^{m+1}(\beta(\eta))^m}{\left[\left(\sqrt{\alpha_1(\eta)} + \sqrt{\alpha_1(\eta) - 4\beta(\eta)}\right)^m + \left(\sqrt{\alpha_1(\eta)} - \sqrt{\alpha_1(\eta) - 4\beta(\eta)}\right)^m\right]^2} \\ &= \left[\frac{2^{m+1}(\sqrt{\beta(\eta)})^m}{\left(\sqrt{\alpha_1(\eta)} + \sqrt{\alpha_1(\eta) - 4\beta(\eta)}\right)^m + \left(\sqrt{\alpha_1(\eta)} - \sqrt{\alpha_1(\eta) - 4\beta(\eta)}\right)^m}\right]^2 \qquad (43) \\ &= \left[\frac{2^{m+1}}{\left(\sqrt{\gamma(\eta)} + \sqrt{\gamma(\eta) - 4}\right)^m + \left(\sqrt{\gamma(\eta)} - \sqrt{\gamma(\eta) - 4}\right)^m}\right]^2 \qquad (44) \end{aligned}$$

where

$$\gamma(\eta) = \frac{\alpha_1(\eta)}{\beta(\eta)} = \frac{4(1 - \eta + \eta\lambda_1)^2}{(1 - \eta + \eta\lambda_2)^2}.$$

By Lemma A.6, we have $g(0) = 1, g'(0) = -2m^2(\lambda_1 - \lambda_2)$, and $g'(\eta) < 0$ for any $\eta \in (0, 1]$, implying that $g(\eta)$ is a decreasing function of η on $(0, 1]$. Moreover, since $g(\eta)$ is twice continuously differentiable on an open interval containing 0, by Taylor approximation at $\eta = 0$, we have

$$g(\eta) = 1 - 2m^2(\lambda_1 - \lambda_2)\eta + o(\eta^\delta) \qquad (45)$$

for any $1 < \delta < 2$. For all subsequent analysis, any such δ would do it and we pick $\delta = 3/2$ arbitrarily.

Plugging (45) into $\rho(\eta, K)$, we have

$$\rho(\eta, K) = 1 - 2m^2(\lambda_1 - \lambda_2)\eta + o(\eta^{3/2}) + c'\eta^4 K.$$

Since

$$\frac{\partial}{\partial \eta}\rho(\eta, K)\big|_{\eta=0} < 0, \quad \rho(0, K) = 1$$

hold, there exist some $\bar{\eta}(K) > 0$ such that for every $0 < \eta \leq \bar{\eta}(K)$, we have

$$\rho(1, 0) \leq \rho(\eta, K) < 1.$$

The lower bound follows the fact that $g(\eta)$ is decreasing and $c'\eta^4 K \geq 0$. □

Proof of Theorem 3.4. By Lemma 3.3, there exists some $\bar{\eta}(K)$ such that for every $\eta \in (0, \bar{\eta}(K)]$, we have

$$0 < \rho(1, 0) \leq \rho(\eta, K) < 1.$$

By repeatedly applying

$$\frac{\sum_{k=2}^d E[(u_k^T \tilde{w}_s)^2]}{E[(u_1^T \tilde{w}_s)^2]} = \frac{\sum_{k=2}^d E[(u_k^T w_m)^2]}{E[(u_1^T w_m)^2]} \leq \rho(\eta, K)\frac{\sum_{k=2}^d E[(u_k^T w_0)^2]}{E[(u_1^T w_0)^2]} = \rho(\eta, K)\frac{\sum_{k=2}^d E[(u_k^T \tilde{w}_{s-1})^2]}{E[(u_1^T \tilde{w}_{s-1})^2]},$$

we obtain

$$\frac{\sum_{k=2}^d E[(u_k^T \tilde{w}_s)^2]}{E[(u_1^T \tilde{w}_s)^2]} \leq \rho(\eta, K)^s \frac{\sum_{k=2}^d E[(u_k^T \tilde{w}_0)^2]}{E[(u_1^T \tilde{w}_0)^2]} = \rho(\eta, K)^s \left(\frac{1 - (u_1^T \tilde{w}_0)^2}{(u_1^T \tilde{w}_0)^2}\right).$$

□

A.2. Technical Lemmas

Lemma A.1. *Let w be a vector in $\mathbb{R}^d$, and let P, M be $d \times d$ symmetric matrices. Then, we have*

$$w^T PMPw \leq \|M\|\|Pw\|^2.$$

Proof. By the cyclic property of the trace, we have

$$w^T PMPw = \text{Tr}[w^T PMPw] = \text{Tr}[MPww^T P].$$

Since $Pww^T P$ is positive semi-definite, we have

$$\text{Tr}[MPww^T P] \leq \|M\|\text{Tr}[Pww^T P].$$

Again, by the cyclic property of the trace, we finally have

$$w^T PMPw \leq \|M\|\text{Tr}[Pww^T P] = \|M\|\text{Tr}[w^T PPw] = \|M\|\|Pw\|^2.$$

□

Lemma A.2. *Let w be a vector in $\mathbb{R}^d$ with $\|w\| = 1$ and let C be a $d \times d$ symmetric matrix. Then, for $P = I - ww^T$, we have*

$$\|PC - CP\|^2 = w^T C^2 w - (w^T Cw)^2.$$

Proof. Let $U = PC - CP$. Since the non-zero singular values of U are the square roots of the non-zero eigenvalues of $U^T U$, we focus on $U^T U$. By definition of P, we have

$$U = (I - ww^T)C - C(I - ww^T) = Cww^T - ww^T C,$$

resulting in

$$\begin{aligned} U^T U &= (Cww^T - ww^T C)^T (Cww^T - ww^T C) \\ &= ww^T C^2 ww^T - ww^T Cww^T C - Cww^T Cww^T + Cww^T ww^T C \\ &= (w^T C^2 w)ww^T - (w^T Cw)ww^T C - (w^T Cw)Cww^T + Cww^T C. \end{aligned}$$

For any vector u in $\mathbb{R}^d$, we have

$$\begin{aligned} U^T Uu &= (w^T C^2 w)ww^T u - (w^T Cw)ww^T Cu - (w^T Cw)Cww^T u + Cww^T Cu \\ &= \big[(w^T C^2 w)(w^T u) - (w^T Cw)(w^T Cu)\big]w + \big[w^T Cu - (w^T Cw)(w^T u)\big]Cw \end{aligned} \tag{46}$$

meaning that that $U^T Uu$ lies in the span of w and Cw. This implies that any eigenvector u for $U^T U$ corresponding to a non-zero eigenvalue is of the form

$$u = c_1 w + c_2 Cw. \tag{47}$$

By plugging (47) into (46), we have

$$\begin{aligned} U^T Uu &= c_1[(w^T C^2 w) - (w^T Cw)^2]w + c_2[w^T C^2 w - (w^T Cw)^2]Cw \\ &= [(w^T C^2 w) - (w^T Cw)^2](c_1 w + c_2 Cw) \\ &= [(w^T C^2 w) - (w^T Cw)^2]u. \end{aligned}$$

We conclude that all eigenvalues of $U^T U$ are $(w^T C^2 w) - (w^T Cw)^2$ and possibly 0. Therefore, the spectral radius of $U^T U$ is $|(w^T C^2 w) - (w^T Cw)^2|$. Since it is easy to check that $P^2 = P$ and the expansion of $\|PCw\|^2$ results in

$$\|PCw\|^2 = w^T CP^2 Cw = w^T CPCw = (w^T C^2 w) - (w^T Cw)^2 \geq 0,$$

we have

$$\|U\|^2 = \|PC - CP\|^2 = w^T C^2 w - (w^T Cw)^2.$$

□

Lemma A.3. *Let C be a positive semi-definite $d \times d$ matrix and (λ_1, u_1) be the largest eigenpair of C. Then, for any unit vector w in $\mathbb{R}^d$, we have*

$$w^T C^2 w - (w^T C w)^2 \leq 2\lambda_1^2(1 - (u_1^T w)^2).$$

Proof. Letting

$$w = (u_1^T w)u_1 + (I - u_1 u_1^T)w,$$

we have after some manipulations

$$w^T C w = \lambda_1 (u_1^T w)^2 + w^T (I - u_1 u_1^T) C (I - u_1 u_1^T) w \tag{48}$$

and

$$w^T C^2 w = \lambda_1^2 (u_1^T w)^2 + w^T (I - u_1 u_1^T) C^2 (I - u_1 u_1^T) w. \tag{49}$$

Since the second terms in (48) and (49) are non-negative due to C being positive semi-definite, we have

$$\lambda_1 (u_1^T w)^2 \leq w^T C w \leq \lambda_1, \quad \lambda_1^2 (u_1^T w)^2 \leq w^T C^2 w \leq \lambda_1^2.$$

Therefore,

$$w^T C^2 w - (w^T C w)^2 \leq \lambda_1^2 (1 - (u_1^T w)^4) = 2\lambda_1^2 (1 + (u_1^T w)^2)(1 - (u_1^T w)^2) \leq 2\lambda_1^2 (1 - (u_1^T w)^2)$$

where the last inequality follows from $(u_1^T w)^2 \leq \|u_1\|^2 \|w\|^2 = 1$. □

Lemma A.4. *Let w_t be a sequence of real numbers such that*

$$w_t = (\alpha - \beta) w_{t-1} - \beta(\alpha - \beta) w_{t-2} + \beta^3 w_{t-3} + L_{t-1} + \beta L_{t-2}$$

for $t \geq 3$ and $w_0 = L_0, w_1 = \frac{\alpha}{4} L_0, w_2 = \left(\frac{\alpha}{2} - \beta\right)^2 L_0 + L_1$. Then, we have

$$w_t = p_t(\alpha, \beta) L_0 + \sum_{r=1}^{t-1} q_{t-r-1}(\alpha, \beta) L_r \tag{50}$$

where $p_t(\alpha, \beta)$ and $q_t(\alpha, \beta)$ are recurrence polynomials defined as

$$p_t(\alpha, \beta) = (\alpha - \beta) p_{t-1}(\alpha, \beta) - \beta(\alpha - \beta) p_{t-2}(\alpha, \beta) + \beta^3 p_{t-3}(\alpha, \beta) \tag{51}$$

$$q_t(\alpha, \beta) = (\alpha - \beta) q_{t-1}(\alpha, \beta) - \beta(\alpha - \beta) q_{t-2}(\alpha, \beta) + \beta^3 q_{t-3}(\alpha, \beta) \tag{52}$$

for $t \geq 3$ with

$$p_0(\alpha, \beta) = 1, \quad p_1(\alpha, \beta) = \frac{\alpha}{4}, \quad p_2(\alpha, \beta) = \left(\frac{\alpha}{2} - \beta\right)^2, \tag{53}$$

$$q_0(\alpha, \beta) = 1, \quad q_1(\alpha, \beta) = \alpha, \quad q_2(\alpha, \beta) = (\alpha - \beta)^2. \tag{54}$$

Moreover, for $t \geq 0$, we have

- *if $0 \leq \alpha = 4\beta$,*

$$p_t(4\beta, \beta) = \beta^t \geq 0, \quad q_t(4\beta, \beta) = (t+1)^2 \beta^t \geq 0. \tag{55}$$

- *if $0 \leq 4\beta < \alpha$,*

$$p_t(\alpha, \beta) = \left[\frac{1}{2}\left(\frac{\sqrt{\alpha}}{2} + \frac{\sqrt{\alpha - 4\beta}}{2}\right)^t + \frac{1}{2}\left(\frac{\sqrt{\alpha}}{2} - \frac{\sqrt{\alpha - 4\beta}}{2}\right)^t\right]^2 > p_t(4\beta, \beta) \geq 0, \tag{56}$$

$$q_t(\alpha, \beta) = \frac{1}{\alpha - 4\beta}\left[\left(\frac{\sqrt{\alpha}}{2} + \frac{\sqrt{\alpha - 4\beta}}{2}\right)^{t+1} - \left(\frac{\sqrt{\alpha}}{2} - \frac{\sqrt{\alpha - 4\beta}}{2}\right)^{t+1}\right]^2 \geq 0. \tag{57}$$

- *if* $0 \leq \alpha < 4\beta$,

$$p_t(\alpha, \beta) \leq p_t(4\beta, \beta), \quad q_t(\alpha, \beta) \leq q_t(4\beta, \beta). \tag{58}$$

Proof. It is easy to check that w_0, w_1, and w_2 satisfy (50). Suppose that (50) holds for $t-1, t-2, t-3$. Then, we have

$$\begin{aligned} w_t &= (\alpha - \beta)w_{t-1} - \beta(\alpha - \beta)w_{t-2} + \beta^3 w_{t-3} + L_{t-1} + \beta L_{t-2} \\ &= p_t(\alpha, \beta)L_0 + L_{t-1} + \alpha L_{t-2} + (\alpha - \beta)^2 L_{t-3} + \sum_{r=1}^{t-4} q_{t-r-1}(\alpha, \beta)L_r \\ &= p_t(\alpha, \beta)L_0 + \sum_{r=1}^{t-1} q_{t-r-1}(\alpha, \beta)L_r. \end{aligned}$$

Therefore, (50) holds by induction.

Next, we prove (55), (56), (57) and (58). The characteristic equation of (51) is

$$r^3 - (\alpha - \beta)r^2 + \beta(\alpha - \beta)r - \beta^3 = 0. \tag{59}$$

If $0 \leq \alpha = 4\beta$, (59) has a cube root of $r = \beta$. From initial conditions (53) and (54), we obtain

$$p_t(4\beta, \beta) = \beta^t \geq 0, \quad q_t(4\beta, \beta) = (t+1)^2\beta^t \geq 0. \tag{60}$$

If $0 \leq 4\beta < \alpha$, the roots of (59) are

$$r = \beta, \frac{\alpha - 2\beta}{2} + \frac{\sqrt{\alpha^2 - 4\alpha\beta}}{2}, \frac{\alpha - 2\beta}{2} - \frac{\sqrt{\alpha^2 - 4\alpha\beta}}{2}.$$

With initial conditions (53), we obtain

$$\begin{aligned} p_t(\alpha, \beta) &= \frac{1}{4}\left(\frac{\alpha - 2\beta}{2} + \frac{\sqrt{\alpha^2 - 4\alpha\beta}}{2}\right)^t + \frac{1}{4}\left(\frac{\alpha - 2\beta}{2} - \frac{\sqrt{\alpha^2 - 4\alpha\beta}}{2}\right)^t + \frac{1}{2}\beta^t \\ &= \left[\frac{1}{2}\left(\frac{\sqrt{\alpha}}{2} + \frac{\sqrt{\alpha - 4\beta}}{2}\right)^t + \frac{1}{2}\left(\frac{\sqrt{\alpha}}{2} - \frac{\sqrt{\alpha - 4\beta}}{2}\right)^t\right]^2. \end{aligned}$$

The second equality can be verified by expanding the square expression.

By the Binomial Theorem and the fact that $\alpha > 4\beta$, we have

$$p_t(\alpha, \beta) \geq \frac{1}{2}\left(\frac{\alpha - 2\beta}{2}\right)^t + \frac{1}{2}\beta^t > \beta^t \geq 0.$$

On the other hand, using (54), we have

$$\begin{aligned} q_t(\alpha, \beta) &= \frac{1}{\alpha - 4\beta}\left[\left(\frac{\alpha - 2\beta}{2} + \frac{\sqrt{\alpha^2 - 4\alpha\beta}}{2}\right)^{t+1} + \left(\frac{\alpha - 2\beta}{2} - \frac{\sqrt{\alpha^2 - 4\alpha\beta}}{2}\right)^{t+1} - 2\beta^{t+1}\right] \\ &= \frac{1}{\alpha - 4\beta}\left[\left(\frac{\sqrt{\alpha}}{2} + \frac{\sqrt{\alpha - 4\beta}}{2}\right)^{t+1} - \left(\frac{\sqrt{\alpha}}{2} - \frac{\sqrt{\alpha - 4\beta}}{2}\right)^{t+1}\right]^2 \\ &\geq 0. \end{aligned}$$

Again, the second equality can be established by expanding the square expression.

If $0 \leq \alpha < 4\beta$, the roots of (59) are

$$r = \beta, \frac{\alpha - 2\beta}{2} + \frac{\sqrt{4\alpha\beta - \alpha^2}}{2}i, \frac{\alpha - 2\beta}{2} - \frac{\sqrt{4\alpha\beta - \alpha^2}}{2}i.$$

Setting

$$\cos\theta_p = \frac{\alpha - 2\beta}{2\beta}, \quad \sin\theta_p = \frac{\sqrt{4\alpha\beta - \alpha^2}}{2\beta}$$

it is easy to verify that

$$\begin{aligned} p_t(\alpha,\beta) &= \frac{1}{4}\beta^t\Big[\cos\theta_p + i\sin\theta_p\Big]^t + \frac{1}{4}\beta^t\Big[\cos\theta_p - i\sin\theta_p\Big]^t + \frac{1}{2}\beta^t \\ &= \frac{1}{4}(e^{i\theta t} + e^{-i\theta t})\beta^t + \frac{1}{2}\beta^t \\ &= \frac{1}{4}|e^{i\theta t} + e^{-i\theta t}|\beta^t + \frac{1}{2}\beta^t \\ &\le \frac{1}{4}(|e^{i\theta t}| + |e^{-i\theta t}|)\beta^t + \frac{1}{2}\beta^t \\ &= \beta^t. \end{aligned}$$

Moreover, with

$$\cos\theta_q = \frac{\alpha - 2\beta}{2\beta}, \quad \sin\theta_q = \frac{\sqrt{4\alpha\beta - \alpha^2}}{2\beta}, \quad \cos\phi_q = 1 - \frac{\alpha}{2\beta}, \quad \sin\phi_q = -\frac{\sqrt{4\alpha\beta - \alpha^2}}{2\beta},$$

it can be seen by using elementary calculus that

$$q_t(\alpha,\beta) = \left[\frac{2\beta}{4\beta - \alpha} + \frac{2\beta}{4\beta - \alpha}\cos(\phi_q + t\theta_q)\right]\beta^t. \tag{61}$$

Let

$$Q(t) = \frac{q_t(4\beta,\beta) - q_t(\alpha,\beta)}{\beta^t}.$$

Then, from (51) and (53), we have

$$Q(0) = 0, \quad Q(1) = \frac{4\beta - \alpha}{\beta}, \quad Q(2) = \frac{(4\beta - \alpha)(2\beta + \alpha)}{\beta^2}, \quad Q(3) = \frac{(\alpha^2 + 4\beta^2)(4\beta - \alpha)}{\beta^3} \tag{62}$$

resulting in

$$Q(2) - Q(0) = \frac{(4\beta - \alpha)(2\beta + \alpha)}{\beta^2} \ge 0, \quad Q(3) - Q(1) = \frac{(\alpha^2 + 3\beta^2)(4\beta - \alpha)}{\beta^3} \ge 0. \tag{63}$$

In order to show $Q(t) \ge 0$ for $t \ge 0$, we prove $Q(t+2) - Q(t) \ge 0$ for $t \ge 0$. Using (60), (61) and standard trigonometric equalities, it follows that

$$Q(t+2) - 2Q(t) + Q(t-2) = 8 + \frac{2\alpha}{\beta}\cos(\phi_q + t\theta_q).$$

In turn, we have

$$\begin{aligned} Q(t+2) - Q(t) &= Q(t) - Q(t-2) + 8 + \frac{2\alpha}{\beta}\cos(\phi_q + t\theta_q) \\ &\ge Q(t) - Q(t-2) + 8 - \frac{2\alpha}{\beta} \\ &= Q(t) - Q(t-2) + \frac{2(4\beta - \alpha)}{\beta} \\ &\ge Q(t) - Q(t-2). \end{aligned} \tag{64}$$

From (62), (63), and (64), we obtain $Q(t) \ge 0$ for $t \ge 0$ implying

$$q_t(\alpha,\beta) \le q_t(4\beta,\beta)$$

for $t \ge 0$. □

Lemma A.5. *Let w_t be a sequence of non-negative real numbers such that*

$$w_t \leq \alpha w_{t-1} + \beta w_{t-2} + L_{t-1}$$

for $t \geq 2$ with $w_0 = 0, w_1 \leq L_0$. If $\alpha, \beta \geq 0$, we have

$$w_t \leq \sum_{l=0}^{t-1} r_{t-l-1}(\alpha, \beta) L_l \tag{65}$$

where $r_t(\alpha, \beta)$ is a recurrence polynomial defined as

$$r_t(\alpha, \beta) = \alpha r_{t-1}(\alpha, \beta) + \beta r_{t-2}(\alpha, \beta) \tag{66}$$

for $t \geq 2$ with

$$r_0(\alpha, \beta) = 1, \quad r_1(\alpha, \beta) = \alpha. \tag{67}$$

Proof. From $w_0 = 0$ and $w_1 \leq L_0$, it is obvious that (65) holds for $t = 0$ and $t = 1$. Suppose that (65) holds for $t - 1$ and $t - 2$. Then, we have

$$\begin{aligned} w_t &\leq \alpha w_{t-1} + \beta w_{t-2} + L_{t-1} \\ &\leq \alpha \sum_{l=0}^{t-2} r_{t-l-2}(\alpha, \beta) L_l + \beta \sum_{l=0}^{t-3} r_{t-l-3}(\alpha, \beta) L_l + L_{t-1} \\ &= L_{t-1} + \alpha L_{t-2} + \sum_{l=0}^{t-3} (\alpha r_{t-l-2}(\alpha, \beta) + \beta r_{t-l-3}(\alpha, \beta)) L_l \\ &= \sum_{l=0}^{t-1} r_{t-l-1}(\alpha, \beta) L_l. \end{aligned}$$

Therefore, by mathematical induction, (65) holds for every t. □

Lemma A.6. *For*

$$g(\eta) = \left[\frac{2^{m+1}}{h(\eta)} \right]^2$$

where

$$h(\eta) = \left(\sqrt{\gamma(\eta)} + \sqrt{\gamma(\eta) - 4} \right)^m + \left(\sqrt{\gamma(\eta)} - \sqrt{\gamma(\eta) - 4} \right)^m, \quad \gamma(\eta) = \frac{4(1 - \eta + \eta\lambda_1)^2}{(1 - \eta + \eta\lambda_2)^2},$$

we have

$$g(0) = 1, \quad g'(0) = -2m^2(\lambda_1 - \lambda_2),$$

and

$$g'(\eta) < 0$$

for any $\eta \in (0, 1]$.

Proof. Since $\gamma(0) = 4$, it is obvious that $g(0) = 1$ holds. Next, by differentiating $\gamma(\eta)$, we have

$$\gamma'(\eta) = \frac{8(1 - \eta + \eta\lambda_1)(\lambda_1 - \lambda_2)}{(1 - \eta + \eta\lambda_2)^3} > 0$$

for $\eta \in (0, 1]$.

Using the chain rule on (44), we obtain

$$\begin{aligned}
g'(\eta) = -2\gamma'(\eta)\left[\frac{2^{m+1}}{h(\eta)}\right]\cdot\Bigg\{&\left[\frac{1}{2\sqrt{\gamma(\eta)}}+\frac{1}{2\sqrt{\gamma(\eta)-4}}\right]\left[\frac{m2^{m+1}\big(\sqrt{\gamma(\eta)}+\sqrt{\gamma(\eta)-4}\big)^{m-1}}{h(\eta)}\right]\\
&+\left[\frac{1}{2\sqrt{\gamma(\eta)}}-\frac{1}{2\sqrt{\gamma(\eta)-4}}\right]\left[\frac{m2^{m+1}\big(\sqrt{\gamma(\eta)}-\sqrt{\gamma(\eta)-4}\big)^{m-1}}{h(\eta)}\right]\Bigg\}\\
=-\frac{m2^{m+1}\gamma'(\eta)\sqrt{g(\eta)}}{h(\eta)^2}&\left[\frac{1}{\sqrt{\gamma(\eta)}}\Big[\big(\sqrt{\gamma(\eta)}+\sqrt{\gamma(\eta)-4}\big)^{m-1}+\big(\sqrt{\gamma(\eta)}-\sqrt{\gamma(\eta)-4}\big)^{m-1}\Big]\right.\\
&\left.+\frac{1}{\sqrt{\gamma(\eta)-4}}\Big[\big(\sqrt{\gamma(\eta)}+\sqrt{\gamma(\eta)-4}\big)^{m-1}-\big(\sqrt{\gamma(\eta)}-\sqrt{\gamma(\eta)-4}\big)^{m-1}\Big]\right].
\end{aligned}$$

By the Binomial theorem, we have

$$\big(\sqrt{\gamma(\eta)}+\sqrt{\gamma(\eta)-4}\big)^{m-1}=\sum_{k=0}^{m-1}\binom{m-1}{k}\big(\sqrt{\gamma(\eta)-4}\big)^{k}\big(\sqrt{\gamma(\eta)}\big)^{m-k-1}$$

and

$$\big(\sqrt{\gamma(\eta)}-\sqrt{\gamma(\eta)-4}\big)^{m-1}=\sum_{k=0}^{m-1}\binom{m-1}{k}\big(-\sqrt{\gamma(\eta)-4}\big)^{k}\big(\sqrt{\gamma(\eta)}\big)^{m-k-1},$$

resulting in

$$\big(\sqrt{\gamma(\eta)}+\sqrt{\gamma(\eta)-4}\big)^{m-1}+\big(\sqrt{\gamma(\eta)}-\sqrt{\gamma(\eta)-4}\big)^{m-1}=2\sum_{k=0}^{\lfloor(m-1)/2\rfloor}\binom{m-1}{2k}\big(\sqrt{\gamma(\eta)-4}\big)^{2k}\big(\sqrt{\gamma(\eta)}\big)^{m-2k-1} \tag{68}$$

and

$$\big(\sqrt{\gamma(\eta)}+\sqrt{\gamma(\eta)-4}\big)^{m-1}-\big(\sqrt{\gamma(\eta)}-\sqrt{\gamma(\eta)-4}\big)^{m-1}=2\sum_{k=0}^{\lfloor(m-2)/2\rfloor}\binom{m-1}{2k+1}\big(\sqrt{\gamma(\eta)-4}\big)^{2k+1}\big(\sqrt{\gamma(\eta)}\big)^{m-2k-2}.$$

As a result, we have

$$\begin{aligned}
g'(\eta) = -\frac{m2^{m+2}\gamma'(\eta)\sqrt{g(\eta)}}{h(\eta)^2}\cdot\Bigg[&\sum_{k=0}^{\lfloor(m-1)/2\rfloor}\binom{m-1}{2k}\big(\sqrt{\gamma(\eta)-4}\big)^{2k}\big(\sqrt{\gamma(\eta)}\big)^{m-2k-2}\\
&+\sum_{k=0}^{\lfloor(m-2)/2\rfloor}\binom{m-1}{2k+1}\big(\sqrt{\gamma(\eta)-4}\big)^{2k}\big(\sqrt{\gamma(\eta)}\big)^{m-2k-2}\Bigg].
\end{aligned}$$

Since $\gamma'(\eta)>0$, $\gamma(\eta)>4$ and $h(\eta)>0$ implying $\sqrt{g(\eta)}>0$ for $\eta\in(0,1]$, we have $g'(\eta)<0$ for any $\eta\in(0,1]$. Fact $h(\eta)>0$ can be established by using m instead of $m-1$ in (68). Moreover, for $\eta=0$, we have

$$g'(0)=-2m^2(\lambda_1-\lambda_2).$$

□