



A linearly convergent algorithm for distributed principal component analysis^{☆☆☆}

Arpita Gang*, Waheed U. Bajwa

Department of Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ 08854, United States

ARTICLE INFO

Article history:

Received 31 December 2020

Revised 16 September 2021

Accepted 21 November 2021

Available online 23 November 2021

Keywords:

Dimensionality reduction

Distributed feature learning

Generalized Hebbian learning

Principal component analysis

ABSTRACT

Principal Component Analysis (PCA) is the workhorse tool for dimensionality reduction in this era of big data. While often overlooked, the purpose of PCA is not only to reduce data dimensionality, but also to yield features that are uncorrelated. Furthermore, the ever-increasing volume of data in the modern world often requires storage of data samples across multiple machines, which precludes the use of centralized PCA algorithms. This paper focuses on the dual objective of PCA, namely, dimensionality reduction and decorrelation of features, but in a distributed setting. This requires estimating the eigenvectors of the data covariance matrix, as opposed to only estimating the subspace spanned by the eigenvectors, when data is distributed across a network of machines. Although a few distributed solutions to the PCA problem have been proposed recently, convergence guarantees and/or communications overhead of these solutions remain a concern. With an eye towards communications efficiency, this paper introduces a feedforward neural network-based one time-scale distributed PCA algorithm termed Distributed Sanger's Algorithm (DSA) that estimates the eigenvectors of the data covariance matrix when data is distributed across an undirected and arbitrarily connected network of machines. Furthermore, the proposed algorithm is shown to converge linearly to a neighborhood of the true solution. Numerical results are also provided to demonstrate the efficacy of the proposed solution.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

The modern era of machine learning involves leveraging *mas-* sive amounts of *high-dimensional* data, which can have large computational and storage costs. To combat the complexities arising because of the high dimensions of data, dimensionality reduction and feature learning techniques play a pivotal and necessary role in information processing. The most common and widely used technique for this task is Principal Component Analysis (PCA) [2] which, in the simplest of terms, transforms data

into uncorrelated features that aid conversion of data from a high-dimensional space to a low-dimensional space while retaining maximum information. Simultaneously, the enormity of the amount of available data makes it difficult to manage it at a single location. There are multiple and an increasing number of scenarios where data is distributed across different locations, either due to storage constraints or by its inherent nature like in the Internet-of-Things [3]. This aspect of the modern-world data have led researchers to explore distributed algorithms, which can process information across different locations/machines [4]. These aforementioned issues have motivated us to study and develop algorithms for distributed PCA that are efficient in terms of computations and communications among multiple machines, and that can also be proven to converge at a fast rate.

When the data is available at a single location, one of the goals of PCA is to find a K -dimensional subspace, given by the column space of a matrix $\mathbf{X} \in \mathbb{R}^{d \times K}$, such that the zero-mean data samples $\mathbf{y} \in \mathbb{R}^d$ ($d \gg K$) retain maximum information when projected onto \mathbf{X} . In other words, when reconstructed as $\mathbf{X}\mathbf{X}^T\mathbf{y}$ (subject to $\mathbf{X}^T\mathbf{X} = \mathbf{I}$), the data samples should have minimum reconstruction

[☆] Preliminary versions of some of the results reported in this paper were presented at the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, May 12–17 2019, [1].

^{☆☆} This work was supported in part by the National Science Foundation under Awards CCF-1453073, CCF-1907658, and OAC-1940074, by the Army Research Office under Awards W911NF-17-1-0546 and W911NF-21-1-0301, and by the DARPA Lagrange Program under ONR/NIWC Contract N660011824020.

* Corresponding author.

E-mail addresses: arpita.gang@rutgers.edu (A. Gang), waheed.bajwa@rutgers.edu (W.U. Bajwa).

error. It can be shown that this minimal error solution is given by the projection of data onto the subspace spanned by the eigenvectors of data covariance matrix. This implies that for dimensionality reduction, learning any basis of that subspace is sufficient. This is referred to as the *subspace learning* problem. But while simple dimension reduction does not necessarily need uncorrelated features, most downstream machine learning tasks like classification, pattern matching, regression, etc., work more efficiently when the data features are uncorrelated. In the case of image coding, e.g., PCA is known as the Karhunen–Loeve transform [5], wherein images are compressed by decorrelating neighbouring pixels. With this goal in mind, one needs to aim to find the specific directions that not only have maximum variance, but that also lead to uncorrelated features when data is projected onto those directions. These specific directions are given by the eigenvectors (also called the principal components) of the data covariance matrix, and not just any set of orthogonal basis vectors spanning the same space. Mathematically, along with minimum reconstruction error, the other goal of PCA is to ensure the condition that the off-diagonal entries of $\mathbb{E}[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X}]$ are zero (i.e., the data gets decorrelated), while finding the eigenspace of the covariance matrix $\mathbb{E}[\mathbf{y} \mathbf{y}^T]$.

As explained above, the true and complete purpose of PCA is served when the search for the optimal solution ends with the specific set of eigenvectors of the data covariance matrix, and not just with the subspace it spans. It is known that getting the principal components from any other basis of the subspace would only require performing singular value decomposition (SVD) of the obtained subspace. Although true, the SVD operation has a high computational complexity, which makes it an expensive step for big data. The traditional solutions for PCA were developed to overcome the cost of SVD and hence reverting back to it defeats the whole purpose.

Thus, even though the problem of dimensionality reduction of data has many optimal solutions (corresponding to all the sets of basis vectors spanning the K -dimensional space), our goal is to find only the ones that give the eigenvectors as the basis. In terms of optimization geometry of the PCA problem in which one tries to minimize the mean-squared reconstruction error under an orthogonality constraint, it is a non-convex strict-saddle function. In a strict-saddle function, all the stationary points except the local minima are strict saddles wherein the Hessians have at least one negative eigenvalue that helps in escaping these saddle points [6,7]. Also, in the case of PCA the local minima are the same as the global minima. These geometric aspects make PCA, despite being non-convex, a “nice and solvable” problem whose optimal solution can be reached efficiently. However, note that the set of global minima contains, along with the set of eigenvectors as basis, all other possible bases that are rotated with respect to the eigenvectors. And our goal is not to find just any of the global minima but to look into a very particular subset of it, where the basis is not rotated.

A very popular tool that has been used to learn features of data, and hence compress it, is autoencoders. It was shown in Baldi and Hornik [8] that the globally optimum weights of an autoencoder for minimum reconstruction error are the principal components of the covariance matrix of the input data. In [9], Oja described how using the Hebbian rule for updating the weights of a linear neural network would extract the first principal component from the input data. Several other Hebbian-based rules like Rubner’s model, APEX model [10], Generalized Hebbian Algorithm (GHA) [11], etc., were proposed to extend this idea of training a neural network for finding the first eigenvector to extract the first K principal components (eigenvectors) of the input covariance matrix. Given the parallelization potential, a feedforward linear neural network-based solution for PCA seems to be very attractive.

The other aspect of modern day data is, as mentioned earlier, its massive size. Collating the huge amount of raw data is usually prohibitive due to communications overhead and/or privacy concerns. These reasons have encouraged researchers over the last couple of decades to develop algorithms that can solve various problems for non-collocated data. The algorithms developed to deal with such scenarios can be broadly classified into two categories: (1) the setups where a central entity/server is required to co-ordinate among the various data centers to yield the final result, and (2) the setup where the data is scattered over an arbitrary network of interconnected data centers with no availability of a central co-ordinating node. The authors in Nokleby et al. [3], Yang et al. [12] talk about these different setups and the algorithms developed for each of them in more detail. The second scenario is more generic and usually algorithms developed for such setups can be easily modified to be applied to the first scenario. The terms *distributed* and *decentralized* are used interchangeably for both the setups in the literature. In this paper, we focus on the latter scenario of arbitrarily connected networks and henceforth call it *distributed* setup. Hence, here our goal is to solve the PCA problem in the distributed manner when data is scattered over a network of interconnected nodes such that all the nodes in the network eventually agree with each other and converge to the true principal components of the distributed data.

1.1. Relation to prior work

PCA was developed to find simpler models of smaller dimensions that can approximately fit some data. Some seminal work was done by Pearson [13], who aimed at fitting a line to a set of points, and by Hotelling [2], where a method for the classical PCA problem of decorrelating the features of a given set of data points (observations) by finding the principal components was proposed. Later, some fast iterative methods like the power method, Lanczos algorithm, and orthogonal iterations [14] were proposed, which were proved to converge to the eigenvectors at a linear rate in the case of symmetric matrices. Many other iterative methods have been proposed over the last few decades that are based on the well-known Hebbian learning rule [15] like Oja’s method [9], generalized Hebbian algorithm [11], APEX [10], etc. The analysis for Oja’s algorithm has been provided in Yi et al. [16], which shows that in the deterministic setting the convergence to the first eigenvector is guaranteed at a linear rate for some conditions on the step size and initial estimates. The work in Lv et al. [17] extended the analysis to the generalized Hebbian case for convergence to the first K principal eigenvectors for a specific choice of step size.

While ways to solve PCA in the centralized case when data is available at a single location have been around for nearly a century, distributed solutions are very recent. Within our distributed setup where we assume a network of arbitrarily connected nodes with no central server, the data distribution can be broadly classified into two types: (1) distribution by features, and (2) distribution by samples. The PCA algorithms for these two different kinds of distribution are significantly different. While both are completely distributed, the first kind [18–21] involves estimating only one (or a subset) of feature(s) at each node. In this paper, we focus on finding the eigenvectors when the distribution is by samples, which requires estimation of the whole set of eigenvectors at each node of the network. For this type of distribution, power method was adapted for the distributed setup as a subroutine in Raja and Bajwa [22], Wai et al. [23], Raja and Bajwa [24] to extract the first principal component of the global covariance matrix. Such methods make use of an explicit consensus loop [25] after each power iteration to ensure that the nodes (approximately) agree with each other. While a novel approach that reaches the required solu-

tion at the nodes accurately (albeit with a small error due to the consensus iterations), the two time-scale aspect makes it a relatively slow algorithm in terms of communications efficiency. Furthermore, finding multiple principal components with these approaches would require a sequential approach where subsequent components are determined by using a covariance matrix residue that is left after projection on estimates of the higher-order components. In contrast, the work in Raja and Bajwa [26] focuses on finding the top eigenvector in the distributed setup for the streaming data case. A detailed review of various distributed PCA algorithms that exist for different setups is provided in Wu et al. [27].

Next, note that some distributed optimization-based algorithms for non-convex problems are being studied only since recently and those dealing with constrained problems are even fewer. In [28], it is shown that an *unconstrained* non-convex problem converges to a stationary solution at a sublinear rate. The methods proposed in Bianchi and Jakubowicz [29], Wai et al. [30] deal with non-convex objective functions in a distributed setup when the constraint set is convex and [31] works with convex approximations of non-convex problems. Thus, none of these methods are directly applicable to the distributed PCA problem in our setup.

Finally, we proposed an efficient distributed PCA solution in Gang et al. [1] for a distributed network when the data is split sample-wise among the interconnected nodes. In this paper, we extend the preliminary work in Gang et al. [1] and provide a detailed mathematical analysis of the proposed algorithm along with exact convergence rates and extensive numerical experiments.

1.2. Our contributions

The main contributions of this paper are (1) a novel algorithm for distributed PCA, (2) theoretical guarantees for the proposed distributed algorithm with a linear convergence rate to a small neighborhood of the true PCA solution, and (3) experimental results to further demonstrate the efficacy of the proposed algorithm.

Our focus in this paper is to solve the distributed PCA problem so as to find a solution that not only enables dimensionality reduction, but that also provides uncorrelated features of data distributed over a network. That is, our goal is to estimate the true eigenvectors, not just any subspace spanned by them, of the covariance matrix of the data that is distributed across an arbitrarily connected network. Also, we focus on providing a solution that is efficient in terms of communications between the interconnected nodes of an arbitrary network. To that end, we propose a distributed algorithm that is based on the generalized Hebbian algorithm (GHA) proposed by Sanger [11], wherein the nodes perform local computations along with information exchange with their directly connected neighbors, similar to the idea followed in the distributed gradient descent (DGD) approach in Nedic and Ozdaglar [32]. The local computations do not involve the calculation of any gradient, but we instead use a “psuedo gradient,” which we henceforth call *Sanger’s direction*. In our proposed solution, termed the *Distributed Sanger’s Algorithm (DSA)*, we have also done away with the need of explicit consensus iterations for making the nodes agree with each other, thereby making it a one time-scale solution that is more communications efficient. Theoretical guarantees are also provided for our proposed distributed PCA algorithm when using a constant step size. The analysis shows that, when using a constant step size α , the DSA solution reaches within a $\mathcal{O}(\alpha)$ -neighborhood of the optimal solution at a linear rate when the error metric is the angles between the estimated vectors and

the true eigenvectors.¹ We also provide experimental results and comparisons with centralized orthogonal iteration [14], centralized GHA [11], a sequential distributed power method-based approach and distributed projected gradient descent. The results support our claims and analysis.

To the best of our knowledge, this is the first solution for distributed PCA that uses a Hebbian update, achieves network agreement without the use of explicit consensus iterations, and still provably reaches the globally optimum solution (within an error margin) at all nodes at a linear rate.

1.3. Notation and organization

The following notation is used in this paper. Scalars and vectors are denoted by lower-case and lower-case bold letters, respectively, while matrices are denoted by upper-case bold letters. The operator $|\cdot|$ denotes the absolute value of a scalar quantity. The superscript in $\mathbf{a}^{(t)}$ denotes time (or iteration) index, while a^t denotes the exponentiation operation. The superscript $(\cdot)^T$ denotes the transpose operation, $\|\cdot\|_F$ denotes the Frobenius norm of matrices, while both $\|\cdot\|$ and $\|\cdot\|_2$ denote the ℓ_2 -norm of vectors. Given a matrix \mathbf{A} , both a_{ij} and $(\mathbf{A})_{ij}$ denote its entry at the i th row and j th column, while \mathbf{a}_j denotes its j th column.

The rest of the paper is organized as follows. In Section 2, we describe and mathematically formulate the distributed PCA problem, while Section 3 describes the proposed distributed algorithm, which is based on the generalized Hebbian algorithm. In Section 4, we derive a general result for a modified generalized Hebbian algorithm that aids in the convergence analysis of the proposed distributed algorithm, while convergence guarantees for the proposed algorithm are provided in Section 5. We provide numerical results in Section 6 to show efficacy of the proposed method and provide concluding remarks in Section 7. The statements and proofs of some auxiliary lemmas, which are needed for the proofs of the main lemmas that are used within the convergence analysis in Sections 4 and 5, are given in Appendix A, while Appendices Appendix B–Appendix F contain the formal statements and proofs of the main lemmas.

2. Problem formulation

Principal Component Analysis (PCA) aims at finding the basis of a low-dimensional space that can decorrelate the features of data points and also retain maximum information. More formally, for a random vector $\mathbf{y} \in \mathbb{R}^d$ with $\mathbb{E}[\mathbf{y}] = \mathbf{0}$, PCA involves finding the top- K eigenvectors of the covariance matrix $\Sigma := \mathbb{E}[\mathbf{y}\mathbf{y}^T]$. The zero mean assumption is taken here without loss of generality as the mean can be subtracted in case data is not centered. Mathematically, PCA can be formulated as

$$\begin{aligned} \mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \quad & \mathbb{E}[\|\mathbf{y} - \mathbf{X}\mathbf{X}^T\mathbf{y}\|_2^2] \quad \text{such that} \\ & \forall l \neq q, \quad \left(\mathbb{E}[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X}] \right)_{lq} = 0. \end{aligned} \quad (1)$$

The constraint $\left(\mathbb{E}[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X}] \right)_{lq} = 0, \forall l \neq q$, ensures that \mathbf{X} decorrelates the features of \mathbf{y} . Now, $\mathbb{E}[\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X}] = \mathbf{X}^T \mathbb{E}[\mathbf{y} \mathbf{y}^T] \mathbf{X} = \mathbf{X}^T \Sigma \mathbf{X}$ and it is straightforward to see that this quantity is diagonal only if \mathbf{X} contains the eigenvectors of Σ . This explains why the search for a solution of PCA ends with the eigenvectors and not the subspace

¹ Our results can also be extrapolated to guarantee exact convergence with decaying step size, albeit at a slower than linear rate.

spanned by them. In practice, we do not have access to Σ and so a covariance matrix estimated from the samples of \mathbf{y} is used instead. Specifically, for a dataset with N samples $\{\mathbf{y}_i\}_{i=1}^N$, or equivalently, for a data matrix $\mathbf{Y} := [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N]$, the sample covariance matrix can be written as $\mathbf{C} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^T$ such that $\Sigma := \mathbb{E}[\mathbf{C}]$. The true solution for PCA is then obtained by finding the eigenvectors of the covariance matrix \mathbf{C} , which are also the left singular vectors of the data matrix \mathbf{Y} . The empirical form of (1) is thus

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} f(\mathbf{X}) = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \|\mathbf{Y} - \mathbf{X}\mathbf{X}^T\mathbf{Y}\|_F^2 \quad \text{such that} \\ \forall l \neq q, \left(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \right)_{lq} = 0. \quad (2)$$

In the literature, however, PCA is usually posed with a ‘relaxed’ orthogonality constraint of $\mathbf{X}^T \mathbf{X} = \mathbf{I}$ instead of $\left(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \right)_{lq} = 0, \forall l \neq q$, as follows:

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}, \mathbf{X}^T \mathbf{X} = \mathbf{I}} f(\mathbf{X}) = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}, \mathbf{X}^T \mathbf{X} = \mathbf{I}} \|\mathbf{Y} - \mathbf{X}\mathbf{X}^T\mathbf{Y}\|_F^2. \quad (3)$$

The optimization formulation in (3) with this constraint will only lead to a subspace spanned by the eigenvectors of \mathbf{C} as the solution, thus actually making it a Principal Subspace Analysis (PSA) formulation. In other words, although the formulation (3) gives a solution on the Stiefel manifold, the actual PCA formulation (2) requires the solution to be within a very specific subset of that manifold that corresponds to the eigenvectors of \mathbf{C} . The accuracy of the solutions given by the PCA and PSA formulations will be the same when measured in terms of the principal angles between the subspace estimates and the true subspace spanned by the eigenvectors of the covariance matrix. Specifically, if $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_K]$ is an estimate of the basis of the space spanned by the eigenvectors $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_K]$, then the principal angles between \mathbf{Q} and \mathbf{X} given by both (2) and (3) will be the same. But a more suitable measure of accuracy for any PCA solution should be the angles between \mathbf{x}_i and \mathbf{q}_i for all $i = 1, \dots, K$, which motivates us to judge the efficacy of any solution with respect to this metric instead of the principal subspace angles.

In the distributed setup considered in this paper, we consider a network of M nodes such that the undirected graph, $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, describing the network is connected. Here $\mathcal{V} = \{1, 2, \dots, M\}$ is the set of nodes and \mathcal{E} is the set of edges, i.e., $(i, j) \in \mathcal{E}$ if there is a direct path between i and j . The set of neighbors for any node i is denoted by \mathcal{N}_i . Under the setup of samples being distributed over the M nodes, let us assume that the i th node has a data matrix \mathbf{Y}_i containing N_i samples such that $N = \sum_{i=1}^M N_i$. Thus each node has access to only a local covariance matrix $\mathbf{C}_i = \frac{1}{N_i} \mathbf{Y}_i \mathbf{Y}_i^T$ instead of the global covariance matrix but one can see that $\mathbf{NC} = \sum_{i=1}^M N_i \mathbf{C}_i$. In this setting, a straightforward approach might be that each node finds its own solution independent of the data at all the other nodes. While this might seem viable, this approach will have major drawbacks. Recall that the sample covariance \mathbf{C} approximates the population covariance Σ at a rate of $\mathcal{O}(f(N^{-1}))$, where f is some function (depending on the distribution) of the number of samples N [33]. Since the local data has smaller number of samples than the global data, working with the local covariance matrix \mathbf{C}_i alone instead of somehow using the whole data will lead to a larger error in estimation of the eigenvectors. Also, since uniform sampling from the underlying data distribution is not guaranteed in distributed setups, the samples at a node may end up being from a narrow part of the entire distribution, thus being more biased away from the true distribution. This invites the need for the nodes to collaborate amongst themselves in a way that all the data is utilized to find estimates of the eigenvectors at each node while ensuring that all the nodes agree with each other. Thus, for

a distributed setting, the PCA problem in (1) can be rewritten here as

$$\mathbf{X} = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \sum_{i=1}^M f_i(\mathbf{X}) = \arg \min_{\mathbf{X} \in \mathbb{R}^{d \times K}} \sum_{i=1}^M \|\mathbf{Y}_i - \mathbf{X}\mathbf{X}^T\mathbf{Y}_i\|_F^2 \quad \text{such that} \\ \forall l \neq q, \left(\mathbf{X}^T \left(\sum_{i=1}^M \mathbf{Y}_i \mathbf{Y}_i^T \right) \mathbf{X} \right)_{lq} = 0. \quad (4)$$

It is easy to see that $\sum_{i=1}^M f_i(\mathbf{X}) = f(\mathbf{X})$. Also, in a distributed setup, each node i maintains its own copy \mathbf{X}_i of the variable \mathbf{X} due to the difference in local information (local data) they carry. Thus, all nodes need to agree with each other to ensure the entire network reaches the same true solution. Hence, the true distributed PCA objective is written as

$$\arg \min_{\mathbf{X}_i \in \mathbb{R}^{d \times K}} \sum_{i=1}^M \|\mathbf{Y}_i - \mathbf{X}_i \mathbf{X}_i^T \mathbf{Y}_i\|_F^2 \quad \text{such that} \quad \forall j \in \mathcal{N}_i, \mathbf{X}_i = \mathbf{X}_j \quad \text{and} \\ \forall l \neq q, \left(\mathbf{X}^T \left(\sum_{i=1}^M \mathbf{Y}_i \mathbf{Y}_i^T \right) \mathbf{X} \right)_{lq} = 0. \quad (5)$$

Note that (1)–(5) are non-convex optimization problems due to the non-convexity of the constraint set. One possible solution to the PCA problem is to instead solve a convex relaxation of the original non-convex function [34,35]. The issue with these solutions is that they require $\mathcal{O}(d^2)$ memory and computation, which can be prohibitive in high-dimensional settings. In addition, due to $\mathcal{O}(d^2)$ iterate size these solutions are not ideal for distributed settings. Also, these formulations, without any further constraints, will not necessarily give a basis that is the set of dominant eigenvectors. Instead, they might end up giving a rotated basis as explained earlier, thereby not completing the task of decorrelating features. Hence, in this paper we use an algebraic method based on GHA for neural network training, which has $\mathcal{O}(dK)$ memory and computation requirements, to solve the distributed PCA problem. Our goal is to converge to the true eigenvectors of the global covariance matrix \mathbf{C} at every node of the network. As noted earlier in Section 1.1, distributed variants of the power method exist in the literature [22–24] that can find the dominant eigenvector but these methods employ two time-scale approaches that involve several consensus averaging rounds for each iteration of the power method. Such two time-scale approaches can be expensive in terms of communications cost. In this paper, we propose a one time-scale method that finds the top K eigenvectors of the global sample covariance matrix \mathbf{C} at each node through local computations and information exchange with neighbors. The proposed method also converges linearly up to a neighborhood of the true solution when the error metric considered is the angle between the estimates and the true eigenvectors.

3. The proposed algorithm

In [11], Sanger proposed a generalized Hebbian algorithm (GHA) to train a neural network and find the eigenvectors of the input autocorrelation matrix (same as the covariance matrix for zero-mean input). The outputs of such a network, when the weights are given by the eigenvectors, are the uncorrelated features of the input data that allow data reconstruction with minimal error, hence serving the true purpose of PCA. The algorithm was originally developed to tackle the centralized PCA problem in the case of streaming data, where a new data sample $\mathbf{y}_t, t = 1, 2, \dots$, arrives at each time instance.

In this paper we consider a batch setting, but the alignment of GHA with our basic goal of finding the eigenvectors motivates us to leverage it for our distributed setup. The rationale behind the

idea of extrapolating the streaming case to a distributed batch setting is simple: since $\mathbb{E}[\mathbf{y}_t \mathbf{y}_t^T] = \mathbb{E}[\mathbf{Y}_i \mathbf{Y}_i^T] = \mathbf{\Sigma}$, the sample-wise distributed data setting can be seen as a *mini-batch* variant of the streaming data setting. In the context of neural network training, our approach can be viewed as training a network at each node with a mini-batch of samples in a way that all nodes end up with the same trained network whose weights are given by the eigenvectors of the autocorrelation matrix of the entire batch of samples.

The iterate for the GHA as given in Sanger [11] has the following update for the matrix of eigenvectors (i.e., the neural network weight matrix) \mathbf{X} when the t th sample \mathbf{y}_t arrives at the input of the neural network:

$$\mathbf{X}^{(t+1)} = \mathbf{X}^{(t)} + \alpha_t \left[\mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} - \mathbf{X}^{(t)} \mathcal{U} \left((\mathbf{X}^{(t)})^T \mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} \right) \right], \quad (6)$$

where $\mathcal{U} : \mathbb{R}^{K \times K} \rightarrow \mathbb{R}^{K \times K}$ is an operator that sets all the elements below the diagonal to zero and α_t is the step size. For $K = 1$, and defining $\mathbf{\Sigma}_t = \mathbf{y}_t \mathbf{y}_t^T$, it was shown in Oja and Karhunen [9] that the term $(\mathbf{X}^{(t)})^T \mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} = (\mathbf{X}^{(t)})^T \mathbf{\Sigma}_t \mathbf{X}^{(t)}$ is the consequence of a power series approximation of Oja's rule in lieu of the explicit normalization used in the case of the power method. In the case of $K > 1$, $\mathcal{U} \left((\mathbf{X}^{(t)})^T \mathbf{y}_t \mathbf{y}_t^T \mathbf{X}^{(t)} \right) = \mathcal{U} \left((\mathbf{X}^{(t)})^T \mathbf{\Sigma}_t \mathbf{X}^{(t)} \right)$ helps combine Oja's algorithm with the Gram-Schmidt orthogonalization step as follows:

$$\begin{aligned} & \mathbf{X}^{(t)} \mathcal{U} \left((\mathbf{X}^{(t)})^T \mathbf{\Sigma}_t \mathbf{X}^{(t)} \right) \\ &= \mathbf{X}^{(t)} \mathcal{U} \left(\begin{bmatrix} (\mathbf{x}_1^{(t)})^T \\ \vdots \\ (\mathbf{x}_K^{(t)})^T \end{bmatrix} \mathbf{\Sigma}_t \begin{bmatrix} \mathbf{x}_1^{(t)} & \dots & \mathbf{x}_K^{(t)} \end{bmatrix} \right) \\ &= \mathbf{X}^{(t)} \mathcal{U} \left(\begin{bmatrix} (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_2^{(t)} & \dots & (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \\ (\mathbf{x}_2^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_2^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_2^{(t)} & \dots & (\mathbf{x}_2^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{x}_K^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_K^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_2^{(t)} & \dots & (\mathbf{x}_K^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \end{bmatrix} \right) \\ &= \mathbf{X}^{(t)} \left(\begin{bmatrix} (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_1^{(t)} & (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_2^{(t)} & \dots & (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \\ 0 & (\mathbf{x}_2^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_2^{(t)} & \dots & (\mathbf{x}_2^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\mathbf{x}_K^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \end{bmatrix} \right) \\ &= \begin{bmatrix} (\mathbf{x}_1^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_1^{(t)} & \sum_{p=1}^2 (\mathbf{x}_p^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_2^{(t)} \mathbf{x}_p^{(t)} & \dots & \sum_{p=1}^K (\mathbf{x}_p^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_K^{(t)} \mathbf{x}_p^{(t)} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}. \quad (7) \end{aligned}$$

Thus, for any $k = 1, \dots, K$, the term involving $\mathcal{U}(\cdot)$ in (6) includes an implicit normalization term $(\mathbf{x}_k^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_k^{(t)}$ as well as an orthogonalization term $\sum_{p=1}^{k-1} (\mathbf{x}_p^{(t)})^T \mathbf{\Sigma}_t \mathbf{x}_k^{(t)} \mathbf{x}_p^{(t)}$, which—analogue to the Gram-Schmidt orthogonalization procedure—forces the estimate $\mathbf{x}_k^{(t)}$ to be orthogonal to all the estimates $\mathbf{x}_p^{(t)}$, $p = 1, \dots, k-1$. Another important thing to note about the GHA algorithm is that, in order to estimate the dominant K eigenvectors, it only requires the corresponding top K eigenvalues to be distinct (and nonzero). In other words, it does not require the covariance matrix to be non-singular.

In the deterministic setting, where we have the full-batch instead of new samples every instance, this iterate changes to

$$\begin{aligned} \mathbf{X}^{(t+1)} &= \mathbf{X}^{(t)} + \alpha_t \left[\mathbf{C} \mathbf{X}^{(t)} - \mathbf{X}^{(t)} \mathcal{U} \left((\mathbf{X}^{(t)})^T \mathbf{C} \mathbf{X}^{(t)} \right) \right] \\ &= \mathbf{X}^{(t)} + \alpha_t \mathcal{H}(\mathbf{C}, \mathbf{X}^{(t)}). \end{aligned} \quad (8)$$

Here, we term $\mathcal{H} : \mathbb{R}^{d \times d} \times \mathbb{R}^{d \times K} \rightarrow \mathbb{R}^{d \times K}$, $\mathcal{H}(\mathbf{C}, \mathbf{X}^{(t)}) := \left(\mathbf{C} \mathbf{X}^{(t)} - \mathbf{X}^{(t)} \mathcal{U} \left((\mathbf{X}^{(t)})^T \mathbf{C} \mathbf{X}^{(t)} \right) \right)$ as the Sanger direction. An iterate similar

Algorithm 1 Distributed Sanger's algorithm (DSA).

Input: $\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_M, [w_{ij}], \alpha, K$

Initialize: $\forall i, \mathbf{X}_i^{(0)} \leftarrow \mathbf{X}_{\text{init}} : \mathbf{X}_{\text{init}} \in \mathbb{R}^{d \times K}, \mathbf{X}_{\text{init}}^T \mathbf{X}_{\text{init}} = \mathbf{I}$

for $t = 1, 2, \dots$ **do**

 Communicate $\mathbf{X}_i^{(t-1)}$ from each node i to its neighbors

 Estimate of eigenvectors at node i : $\mathbf{X}_i^{(t)} \leftarrow$

$\sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j^{(t-1)} + \alpha \mathcal{H}_i(\mathbf{X}_i^{(t-1)})$

end for

Return: $\mathbf{X}_i^{(t)}, i = 1, 2, \dots, M$

to (8) has been proven to have global convergence in Lv et al. [17] for some very specific choice of the step sizes that are dependent on the iterate itself. Its straightforward extension to the distributed case is not possible as that would lead to different step sizes at different nodes of the network, making it difficult to talk about its convergence guarantees. Hence, to adapt this iterative method to our distributed setup, we use the typical combine and update strategy used quite richly in the literature for distributed algorithms such as Nedic and Ozdaglar [32], Shi et al. [36], Cattivelli and Sayed [37], Kar and Moura [38]. The main contributions of such works lie in showing that the resulting distributed algorithms achieve consensus (i.e., all nodes will have the same iterate values eventually) and, in addition, the consensus value is the same as the centralized solution. The convergence guarantees for these methods are mainly restricted to convex and strongly convex problems though. Our distributed version of (8) for PCA, which is non-convex, is based on similar principles of combine and update.

Specifically, the node i at iteration t carries a local copy $\mathbf{X}_i^{(t)}$ of the estimate of the eigenvectors of the global covariance matrix \mathbf{C} . In the combine step, each node i exchanges the iterate values with its immediate neighbors $j \in \mathcal{N}_i$, where \mathcal{N}_i denotes the neighborhood of node i , and then takes a weighted sum of the iterates received along with its local iterate. Then this sum is updated independently at all nodes using their respective local information. Since node i in the network only has access to its local sample covariance \mathbf{C}_i , the update is in the form of a local Sanger's direction given as

$$\mathcal{H}_i(\mathbf{C}_i, \mathbf{X}_i^{(t)}) = \mathbf{C}_i \mathbf{X}_i^{(t)} - \mathbf{X}_i^{(t)} \mathcal{U} \left((\mathbf{X}_i^{(t)})^T \mathbf{C}_i \mathbf{X}_i^{(t)} \right). \quad (9)$$

The details of the proposed distributed PCA algorithm, called the Distributed Sanger's Algorithm (DSA), are given in Algorithm 1. The weight matrix $\mathbf{W} = [w_{ij}]$ in this algorithm is a doubly stochastic matrix conforming to the network topology [25] in the sense that for $i \neq j$, $w_{ij} \neq 0$ when $(i, j) \in \mathcal{E}$ and $w_{ij} = 0$ otherwise. Also, $\forall i, w_{ii} \neq 0$, i.e., there is a self loop at each node. Note that connectivity of the network, as discussed in Section 2, is a necessary condition for convergence of DSA. The connectivity assumption, in turn, ensures the Markov chain underlying the graph \mathcal{G} is aperiodic and irreducible, which implies that the second-largest (in magnitude) eigenvalue of \mathbf{W} , $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$, is strictly less than 1. While DSA shares algorithmic similarities with first-order distributed optimization methods [32,39] in which the combine-and-update strategy is used, our challenge is characterizing its convergence behavior due to the non-convex and constrained nature of the distributed PCA problem. To this end, we first provide a general result in Section 4 where we prove the convergence of a modified form of GHA. Then we utilize that result, along with some linear algebraic tools and additional lemmas provided in the appendices, to characterize the dynamics of the distributed setup in Section 5 and prove the convergence of the proposed algorithm.

4. Convergence analysis of a modified GHA

Let $\mathbf{X}^{(t)} = [\mathbf{x}_1^{(t)} \ \mathbf{x}_2^{(t)} \ \dots \ \mathbf{x}_K^{(t)}] \in \mathbb{R}^{d \times K}$, $K \leq d$, be an estimate of the K -dimensional subspace spanned by the eigenvectors of the covariance matrix \mathbf{C} after t iterations and \mathbf{q}_l , $l = 1, \dots, d$, be the eigenvectors of \mathbf{C} with corresponding eigenvalues λ_l . On expanding (8) using (7), it is clear that the GHA update equation for estimation of the k th eigenvector using a constant step size α is as follows:

$$\mathbf{x}_k^{(t+1)} = \mathbf{x}_k^{(t)} + \alpha (\mathbf{C}\mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_p^{(t)} (\mathbf{x}_p^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}). \quad (10)$$

We now slightly modify (10) by replacing $\mathbf{x}_p^{(t)}$ for $p < k$ by the true eigenvectors \mathbf{q}_p . We term the resulting update equation *modified GHA* and note that this is not an algorithm in the true sense of the term as it cannot be implemented because of its dependence on the true eigenvectors \mathbf{q}_p . The sole purpose of this modified GHA is to help in our ultimate goal of providing convergence guarantee for the DSA algorithm. The update equation of the modified GHA for “estimation” of the k th eigenvector of \mathbf{C} , $k = 1, \dots, K$, has the form

$$\mathbf{x}_k^{(t+1)} = \mathbf{x}_k^{(t)} + \alpha (\mathbf{C}\mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}\mathbf{x}_k^{(t)}). \quad (11)$$

Note that similar to the original GHA, this modified GHA assumes that \mathbf{C} has K distinct eigenvalues, i.e., $\lambda_1 > \lambda_2 > \dots > \lambda_K > \lambda_{K+1} \geq \dots \geq \lambda_d \geq 0$. Now, since \mathbf{q}_l , $l = 1, \dots, d$, are the eigenvectors of a real symmetric matrix, they form a basis for \mathbb{R}^d and can be used for expansion of any $\mathbf{x}_k^{(t)}$ as

$$\mathbf{x}_k^{(t)} = \sum_{l=1}^d z_{k,l}^{(t)} \mathbf{q}_l, \quad (12)$$

where $z_{k,l}^{(t)}$ is the coefficient corresponding to the eigenvector \mathbf{q}_l in the expansion of $\mathbf{x}_k^{(t)}$. Multiplying both sides of (11) by \mathbf{q}_l^T and using the fact that $\mathbf{q}_l^T \mathbf{q}_{l'} = 0$ for $l \neq l'$, we get

$$z_{k,l}^{(t+1)} = z_{k,l}^{(t)} + \alpha (\mathbf{q}_l^T \mathbf{C}\mathbf{x}_k^{(t)} - \mathbf{q}_l^T (\sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}\mathbf{x}_k^{(t)}) - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} z_{k,l}^{(t)}).$$

This gives

$$z_{k,l}^{(t+1)} = z_{k,l}^{(t)} - \alpha (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} z_{k,l}^{(t)}, \quad \text{for } l = 1, \dots, k-1, \quad (13)$$

$$\text{and } z_{k,l}^{(t+1)} = z_{k,l}^{(t)} + \alpha (\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}) z_{k,l}^{(t)}, \quad \text{for } l = k, \dots, d. \quad (14)$$

It has been shown in Yi et al. [16] that the update equation given by

$$\mathbf{x}_1^{(t+1)} = \mathbf{x}_1^{(t)} + \alpha (\mathbf{C}\mathbf{x}_1^{(t)} - (\mathbf{x}_1^{(t)})^T \mathbf{C}\mathbf{x}_1^{(t)} \mathbf{x}_1^{(t)})$$

for $k = 1$ converges to $\pm \mathbf{q}_1$ at a linear rate for a certain condition on the step size α . Specifically, it was proven that $(z_{1,1}^{(t)})^2 \rightarrow 1$ and $\sum_{l=2}^d (z_{1,l}^{(t)})^2 \leq b_1 \rho_1^t$, where $b_1 > 0$ is some constant and $\rho_1 = (\frac{1+\alpha\lambda_2}{1+\alpha\lambda_1})^2 < 1$. Here, we extend the proof to a general k and show that the update equation given in the form of (11) for any $k = 1, \dots, K$, $K < d$, converges to the k th dominant eigenvector.

Theorem 1. Suppose $\alpha \leq \frac{1}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, $\mathbf{q}_k^T \mathbf{x}_k^{(0)} \neq 0$, and $\|\mathbf{x}_k^{(0)}\| = 1$ for all k . Then the modified GHA iterate for $\mathbf{x}_k^{(t)}$ given by (11) converges at a linear rate to the eigenvector $\pm \mathbf{q}_k$ corresponding to the k th largest eigenvalue λ_k of the covariance matrix \mathbf{C} .

Proof. The convergence of $\mathbf{x}_k^{(t)}$ to \mathbf{q}_k requires convergence of the lower-order coefficients $z_{k,1}^{(t)}, \dots, z_{k,k-1}^{(t)}$ and the higher-order coefficients $z_{k,k+1}^{(t)}, \dots, z_{k,d}^{(t)}$ to 0 and convergence of $z_{k,k}^{(t)}$ to ± 1 . Now,

$$\begin{aligned} |\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}| &= |\lambda_k - \sum_{l=1}^d \lambda_l (z_{k,l}^{(t)})^2| \\ &= |\lambda_k - \lambda_k (z_{k,k}^{(t)})^2 - \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 - \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2| \\ &\geq |\lambda_k - \lambda_k (z_{k,k}^{(t)})^2| - |\sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2| \\ &\quad - |\sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2| \end{aligned}$$

$$\text{or, } \lambda_k |1 - (z_{k,k}^{(t)})^2| \leq |\sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2| + |\sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2| + |\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}|. \quad (15)$$

Thus, convergence of the lower-order and the higher-order coefficients to 0 along with convergence of the term $|\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}|$ will also imply the convergence of $z_{k,k}^{(t)}$ to ± 1 . To this end, Lemma 5 in the appendix proves linear convergence of the lower-order coefficients $z_{k,1}^{(t)}, \dots, z_{k,k-1}^{(t)}$ to 0 by showing $\sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 < a_1 \gamma^{t+1}$ for some constants $a_1 > 0$, $\gamma < 1$. Furthermore, Lemma 6 in the appendix shows that $\sum_{l=k+1}^d (z_{k,l}^{(t+1)})^2 \leq a_2 \rho_k^{t+1}$, where $a_1, a_2 > 0$ and $\gamma, \rho_k < 1$, thereby proving linear convergence of the higher-order coefficients to 0. Finally, Lemma 7 in the appendix shows that $|\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}| \leq ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\})$, where $a_4 > 0$ and $\delta, \gamma_1 < 1$. The formal statements and proofs of Lemmas 5–7 are given in Appendices Appendix B–Appendix D, respectively.

Thus,

$$\begin{aligned} \lambda_k |1 - (z_{k,k}^{(t)})^2| &\leq |\sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2| + |\sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2| + ta_4(\delta^{t+1} \\ &\quad + \max\{\delta^t, \gamma_1^t\}) \\ &= \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d \lambda_l (z_{k,l}^{(t)})^2 + ta_4(\delta^{t+1} \\ &\quad + \max\{\delta^t, \gamma_1^t\}) \\ &< \lambda_1 (\sum_{l=1}^{k-1} (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d (z_{k,l}^{(t)})^2) + ta_4(\delta^{t+1} \\ &\quad + \max\{\delta^t, \gamma_1^t\}) \\ &< \lambda_1 (a_1 \gamma^t + a_2 \rho_k^t) + ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}). \end{aligned}$$

Clearly, $\lim_{t \rightarrow \infty} |1 - (z_{k,k}^{(t)})^2| = 0$. Therefore, Theorem 1 shows that with an update equation of the form (11), the iterates $\mathbf{x}_k^{(t)}$ converge linearly to eigenvectors \mathbf{q}_k of the covariance matrix \mathbf{C} . \square

5. Convergence analysis of distributed Sanger's algorithm (DSA)

With the analysis of the modified GHA in hand, let us proceed to analyze the proposed DSA algorithm. The iterate of DSA at node i for the dominant K -dimensional eigenspace estimate ($K \leq d$) is given as

$$\begin{aligned} \mathbf{X}_i^{(t+1)} &= \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j^{(t)} + \alpha \mathcal{H}_i(\mathbf{C}_i, \mathbf{X}_i^{(t)}) \\ &= \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{X}_j^{(t)} + \alpha (\mathbf{C}_i \mathbf{X}_i^{(t)} - \mathbf{X}_i^{(t)} \mathcal{U}((\mathbf{X}_i^{(t)})^T \mathbf{C}_i \mathbf{X}_i^{(t)})), \quad (16) \end{aligned}$$

where $\mathbf{x}_i^{(t)} = [\mathbf{x}_{i,1}^{(t)} \ \mathbf{x}_{i,2}^{(t)} \ \dots \ \mathbf{x}_{i,K}^{(t)}] \in \mathbb{R}^{d \times K}$ is an estimate of the K -dimensional subspace of the global covariance matrix \mathbf{C} at the i th node after t iterations, $\mathcal{H}_i(\mathbf{C}_i, \mathbf{x}_i^{(t)})$ is local Sanger's direction, and $w_{ij} \geq 0$ is a weight that node i assigns to $\mathbf{x}_j^{(t)}$ based on the connectivity between nodes i and j as mentioned before. The Sanger's direction and the update equation for an estimate of the k th eigenvector is thus given as

$$\mathcal{H}_i(\mathbf{C}_i, \mathbf{x}_{i,k}^{(t)}) = \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,p}^{(t)} \quad (17)$$

$$\text{and, } \mathbf{x}_{i,k}^{(t+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_{j,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}) \quad (18)$$

Now, let the average of $\mathbf{x}_{1,k}^{(t)}, \mathbf{x}_{2,k}^{(t)}, \dots, \mathbf{x}_{M,k}^{(t)}$ after t th iteration be denoted as $\bar{\mathbf{x}}_k^{(t)} = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_{i,k}^{(t)}$ and given by taking average of (18) over all the nodes $i = 1, \dots, M$ as

$$\begin{aligned} \bar{\mathbf{x}}_k^{(t+1)} &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} \sum_{i=1}^M \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) + \alpha \mathbf{h}_k^{(t)} \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) + \alpha \mathbf{h}_k^{(t)}, \end{aligned}$$

where $\mathbf{h}_k^{(t)} = \frac{1}{M} \sum_{i=1}^M (\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}))$. We present analysis of the DSA algorithm by first proving convergence of the average $\bar{\mathbf{x}}_k^{(t)}$ to a neighborhood of the eigenvector \mathbf{q}_k of the global covariance matrix \mathbf{C} while using a constant step size. Then with the help of Lemma 8 in Appendix E, which proves that the deviation of the iterates $\mathbf{x}_{i,k}^{(t)}$ at each node from the average $\bar{\mathbf{x}}_k^{(t)}$ is upper bounded, we prove that the iterates at each node also converge to a neighborhood of the true solution. It is noteworthy that the analysis of DSA does not require additional constraints on eigenvalues of \mathbf{C}_i , i.e., similar to GHA, we only require the top K eigenvalues of \mathbf{C} to be distinct and non-zero.

The complete proof of convergence of DSA is done by induction. First, we show the convergence of $\mathbf{x}_{i,1}^{(t)}$ to a $\mathcal{O}(\alpha)$ neighborhood of \mathbf{q}_1 and then analyze the rest of the eigenvector estimates $\mathbf{x}_{i,k}^{(t)}, k = 2, \dots, K$, by assuming that the higher-order estimates have converged.

Case I for induction – $k = 1$ The iterate for the dominant eigenvector is

$$\mathbf{x}_{i,1}^{(t+1)} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_{j,1}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,1}^{(t)} - ((\mathbf{x}_{i,1}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,1}^{(t)}) \mathbf{x}_{i,1}^{(t)}). \quad (19)$$

Theorem 2. Suppose $\alpha \leq \frac{\min_i w_{ii}}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, $\mathbf{q}_1^T \mathbf{x}_{i,1}^{(0)} \neq 0$, and $\|\mathbf{x}_{i,1}^{(0)}\| = 1$. Then the DSA iterate for $\mathbf{x}_{i,1}^{(t)}$ given by (19) converges at a linear rate to an $\mathcal{O}(\alpha)$ neighborhood of the eigenvector $\pm \mathbf{q}_1$ corresponding to the largest eigenvalue λ_1 of the global covariance matrix \mathbf{C} at every node of the network.

Proof. We know that

$$\|\mathbf{x}_{i,1}^{(t)} - \mathbf{x}_1^*\| \leq \|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\| + \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\|, \quad \text{where } \mathbf{x}_1^* = \pm \mathbf{q}_1. \quad (20)$$

The term $\|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\|$ is a measure of consensus in the network and we prove in Lemma 8 in Appendix E that this difference decreases linearly until it reaches a level of $\mathcal{O}(\alpha)$. More precisely,

$$\|\mathbf{x}_{i,1}^{(t)} - \bar{\mathbf{x}}_1^{(t)}\| \leq b_1 (\beta^t + \frac{\alpha}{1-\beta}), \quad (21)$$

where $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\}$. In particular, it is well known that for a connected graph $\beta < 1$. Now, the average iterate of DSA for the estimate of the dominant eigenvector ($k = 1$) is

$$\bar{\mathbf{x}}_1^{(t)} = \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) + \alpha \mathbf{h}_1^{(t-1)}.$$

Thus,

$$\begin{aligned} \bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^* &= \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) - \mathbf{x}_1^* + \alpha \mathbf{h}_1^{(t-1)}) \\ \text{or, } \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &= \|\bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) - \mathbf{x}_1^* + \alpha \mathbf{h}_1^{(t-1)})\| \\ \text{or, } \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &\leq \|\bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)}) - \mathbf{x}_1^*\| \\ &\quad + \alpha \|\mathbf{h}_1^{(t-1)}\|. \end{aligned} \quad (22)$$

We saw in Section 4 that an iterate of the form

$$\bar{\mathbf{x}}_1^{(t)} = \bar{\mathbf{x}}_1^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} - (\bar{\mathbf{x}}_1^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_1^{(t-1)} \bar{\mathbf{x}}_1^{(t-1)})$$

converges linearly to $\mathbf{x}_1^* = \pm \mathbf{q}_1$ for certain conditions on the step size and the initial point. Thus,

$$\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| \leq \rho_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + \alpha \|\mathbf{h}_1^{(t-1)}\|, \quad \text{where } \rho_1 = \frac{1 + \frac{\alpha}{M} \lambda_2}{1 + \frac{\alpha}{M} \lambda_1}.$$

The term $\mathbf{h}_1^{(t-1)}$ in the above equation appears due to the distributed nature of the algorithm and can be bounded separately. Specifically, we prove in Lemma 9, whose formal statement and proof is given in Appendix F, that

$$\|\mathbf{h}_1^{(t-1)}\| \leq 9\lambda_1 b_1 (\beta^{t-1} + \frac{\alpha}{1-\beta}).$$

Thus,

$$\begin{aligned} \|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| &\leq \rho_1 \|\bar{\mathbf{x}}_1^{(t-1)} - \mathbf{x}_1^*\| + 9\alpha\lambda_1 b_1 (\beta^{t-1} + \frac{\alpha}{1-\beta}) \\ &\leq \rho_1 \left(\rho_1 \|\bar{\mathbf{x}}_1^{(t-2)} - \mathbf{x}_1^*\| + 9\alpha\lambda_1 b_1 \beta^{t-2} + 9\alpha\lambda_1 b_1 (\frac{\alpha}{1-\beta}) \right) \\ &\quad + 9\alpha\lambda_1 b_1 \beta^{t-1} + 9\alpha\lambda_1 b_1 (\frac{\alpha}{1-\beta}) \\ &\leq \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + 9\alpha\lambda_1 b_1 \sum_{r=0}^{t-1} (\rho_1 \beta^{-1})^r \beta^{t-1} \\ &\quad + \frac{1}{1-\rho_1} 9\alpha\lambda_1 b_1 (\frac{\alpha}{1-\beta}). \end{aligned}$$

Since $\rho_1, \beta < 1$, we have the following two cases:

1. $\rho_1 \leq \beta \Rightarrow \rho_1 \beta^{-1} \leq 1$. Then, $\sum_{r=0}^{t-1} (\rho_1 \beta^{-1})^r \beta^{t-1} \leq \sum_{r=0}^{t-1} \beta^{t-1} = t\beta^{t-1}$.
2. $\rho_1 > \beta$. Then $\sum_{r=0}^{t-1} (\rho_1 \beta^{-1})^r \beta^{t-1} = \beta^{t-1} + \rho_1 \beta^{t-2} + \dots + \rho_1^{t-1} \beta^{t-1} < \rho_1^{t-1} + \dots + \rho_1^{t-1} = t\rho_1^{t-1}$.

Therefore,

$$\|\bar{\mathbf{x}}_1^{(t)} - \mathbf{x}_1^*\| \leq \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + c_1 t \max\{\rho_1^{t-1}, \beta^{t-1}\} + \frac{c_1}{1-\rho_1} \left(\frac{\alpha}{1-\beta} \right), \quad \text{where } c_1 = 9\alpha\lambda_1 b_1. \quad (23)$$

Consequently, from (21) and (23), we get

$$\begin{aligned} \|\mathbf{x}_{i,1}^{(t)} - \mathbf{x}_1^*\| &\leq b_1(\beta^t + \frac{\alpha}{1-\beta}) + \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| \\ &\quad + c_1 t \max\{\rho_1^{t-1}, \beta^{t-1}\} + \frac{c_1}{1-\rho_1} \left(\frac{\alpha}{1-\beta} \right) \\ &= \rho_1^t \|\bar{\mathbf{x}}_1^{(0)} - \mathbf{x}_1^*\| + b_1 \beta^t + c_1 t \max\{\rho_1^{t-1}, \beta^{t-1}\} \\ &\quad + \left(\frac{c_1}{1-\rho_1} + b_1 \right) \left(\frac{\alpha}{1-\beta} \right). \end{aligned}$$

This proves that $\mathbf{x}_{i,1}^{(t)}$ converges to a neighborhood of $\mathbf{x}_1^* = \mathbf{q}_1$ or $\mathbf{x}_1^* = -\mathbf{q}_1$ at a linear rate. \square

Case II for induction $-1 < k \leq K$ For the remainder of the eigenvectors, we proceed with the proof of convergence by induction. Since we have already proven the base case, we can assume there exist constants $c_{i,p} > 0$ and $\theta_{i,p} < 1$ such that

1. $\|\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T - \mathbf{q}_p \mathbf{q}_p^T\| \leq c_{i,p}(\theta_{i,p}^t + \frac{\alpha}{1-\beta})$, $\forall p = 1, \dots, k-1$, and
2. $\|\mathbf{x}_{i,p}^{(t)}\|^2 \leq 3$, $p = 1, \dots, k-1$, $i = 1, \dots, M$.

Using the inequality in 1) above, we can write $\mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T = \mathbf{q}_p \mathbf{q}_p^T + \phi_{i,p}^{(t)}$, $p = 1, \dots, k-1$ such that $\|\phi_{i,p}^{(t)}\| \leq c_{i,p}(\theta_{i,p}^t + \frac{\alpha}{1-\beta})$. This therefore implies $\frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} = \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{q}_p \mathbf{q}_p^T + \phi_{i,p}^{(t)}) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} = \frac{\alpha}{M} \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} + \alpha \tilde{\psi}_k^{(t)}$, where $\tilde{\psi}_k^{(t)} = \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \phi_{i,p}^{(t)} \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}$.

Thus, we have

$$\begin{aligned} \|\tilde{\psi}_k^{(t)}\| &\leq \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \lambda_1 \|\phi_{i,p}^{(t)}\| \|\bar{\mathbf{x}}_k^{(t)}\| \\ &\leq \frac{1}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \sqrt{3} \lambda_1 c_{i,p} (\theta_{i,p}^t + \frac{\alpha}{1-\beta}) \\ &\leq \frac{1}{M} \sqrt{3} \lambda_1 (k-1) M \bar{c} (\bar{\theta}^t + \frac{\alpha}{1-\beta}) \\ &= \sqrt{3} \lambda_1 (k-1) \bar{c} (\bar{\theta}^t + \frac{\alpha}{1-\beta}), \end{aligned} \quad (24)$$

where $\bar{c} = \max_{i,p} \{c_{i,p}\}$ and $\bar{\theta} = \max_{i,p} \{\theta_{i,p}\} < 1$.

Consequently,

$$\begin{aligned} \bar{\mathbf{x}}_k^{(t+1)} &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) \\ &\quad - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) + \alpha \mathbf{h}_k^{(t)} \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{i=1}^M \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) \\ &\quad + \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} (\mathbf{q}_p \mathbf{q}_p^T - \mathbf{x}_{i,p}^{(t)}(\mathbf{x}_{i,p}^{(t)})^T) \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} + \alpha \mathbf{h}_k^{(t)} \\ &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}) \\ &\quad - \frac{\alpha}{M} \sum_{i=1}^M \sum_{p=1}^{k-1} \phi_{i,p}^{(t)} \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} + \alpha \mathbf{h}_k^{(t)} \end{aligned}$$

$$\begin{aligned} &= \bar{\mathbf{x}}_k^{(t)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t)} - (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t)}) \\ &\quad - \alpha \tilde{\psi}_k^{(t)} + \alpha \mathbf{h}_k^{(t)}. \end{aligned} \quad (25)$$

We can now proceed with the final theorem that characterizes the convergence behavior of DSA.

Theorem 3. Suppose $\alpha \leq \frac{\min_i w_{ii}}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, $\mathbf{q}_k^T \mathbf{x}_{i,k}^{(0)} \neq 0$ and $\|\mathbf{x}_{i,k}^{(0)}\| = 1$, $\forall k = 2, \dots, K$. Then the DSA iterate for $\mathbf{x}_{i,k}^{(t)}$ given by (18) converges at a linear rate to an $\mathcal{O}(\alpha)$ neighborhood of the eigenvector \mathbf{q}_k corresponding to the k th largest eigenvalue λ_k of the global covariance matrix \mathbf{C} at each node of the network.

Proof. We know

$$\|\mathbf{x}_{i,k}^{(t)} - \mathbf{x}_k^*\| \leq \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\|, \quad \text{where } \mathbf{x}_k^* = \pm \mathbf{q}_k. \quad (26)$$

Also, from Lemma 8 in the appendix we know that

$$\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq b_k(\beta^t + \frac{\alpha}{1-\beta}).$$

Now, the average iterate of DSA for estimating the k th eigenvector is

$$\begin{aligned} \bar{\mathbf{x}}_k^{(t)} &= \bar{\mathbf{x}}_k^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t-1)} - ((\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) \bar{\mathbf{x}}_k^{(t-1)}) \\ &\quad - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) + \alpha \mathbf{h}_k^{(t-1)} + \alpha \tilde{\psi}_k^{(t-1)} \\ \text{or, } \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| &\leq \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t-1)} - ((\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) \bar{\mathbf{x}}_k^{(t-1)}) \\ &\quad - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) - \mathbf{x}_k^*\| + \alpha \|\mathbf{h}_k^{(t-1)}\| + \alpha \|\tilde{\psi}_k^{(t-1)}\|. \end{aligned}$$

We know from the discussion in Section 4 that for an iterate of the form

$$\bar{\mathbf{x}}_k^{(t)} = \bar{\mathbf{x}}_k^{(t-1)} + \frac{\alpha}{M} (\mathbf{C} \bar{\mathbf{x}}_k^{(t-1)} - ((\bar{\mathbf{x}}_k^{(t-1)})^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}) \bar{\mathbf{x}}_k^{(t-1)}) - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \bar{\mathbf{x}}_k^{(t-1)}),$$

there exists a constant $\rho'_k < 1$ such that $\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\|$. Thus,

$$\|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| \leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha \|\mathbf{h}_k^{(t-1)}\| + \alpha \|\tilde{\psi}_k^{(t-1)}\|.$$

Now, the term $\|\mathbf{h}_k^{(t-1)}\|$ was bounded in Lemma 9 in the appendix as

$$\|\mathbf{h}_k^{(t-1)}\| \leq 3(k+2)\lambda_1 b_k (\beta^{t-1} + \frac{\alpha}{1-\beta}). \quad (27)$$

Thus, using (24) and (27), we can write

$$\begin{aligned} \|\bar{\mathbf{x}}_k^{(t)} - \mathbf{x}_k^*\| &\leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + \alpha (3(k+2)\lambda_1 b_k (\beta^{t-1} + \frac{\alpha}{1-\beta})) \\ &\quad + \alpha (\sqrt{3} \lambda_1 (k-1) \bar{c} (\bar{\theta}^{t-1} + \frac{\alpha}{1-\beta})) \\ &\leq \rho'_k \|\bar{\mathbf{x}}_k^{(t-1)} - \mathbf{x}_k^*\| + c_k \max\{\beta^{t-1}, \bar{\theta}^{t-1}\} + c_k \frac{\alpha}{1-\beta}, \\ &\quad c_k = \max\{\alpha (3(k+2)\lambda_1 b_k), \alpha (\sqrt{3} \lambda_1 (k-1) \bar{c})\} \\ &\leq \rho'_k (\rho'_k \|\bar{\mathbf{x}}_k^{(t-2)} - \mathbf{x}_k^*\| + c_k \max\{\beta^{t-2}, \bar{\theta}^{t-2}\} + c_k \frac{\alpha}{1-\beta}) \\ &\quad + c_k \max\{\beta^{t-1}, \bar{\theta}^{t-1}\} + c_k \frac{\alpha}{1-\beta} \\ &\leq \rho_k^t \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + c_k \sum_{r=0}^{t-1} (\rho_k' \max\{\beta, \bar{\theta}\}^{-1})^r \max\{\beta, \bar{\theta}\}^{t-1} \\ &\quad + \frac{c_k}{1-\rho_k'} \left(\frac{\alpha}{1-\beta} \right) \end{aligned}$$

$$\leq \rho_k^t \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + c_k t \max\{\rho_k^{t-1}, \beta^{t-1}, \bar{\theta}^{t-1}\} \\ + \frac{c_k}{1 - \rho_k} \left(\frac{\alpha}{1 - \beta} \right).$$

Consequently, from (26) and Lemma 8 we get

$$\|\mathbf{x}_{i,k}^{(t)} - \mathbf{x}_k^*\| \leq b_k \left(\beta^t + \frac{\alpha}{1 - \beta} \right) + \rho_k^t \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| \\ + c_k t \max\{\rho_k^{t-1}, \beta^{t-1}, \bar{\theta}^{t-1}\} + \frac{c_k}{1 - \rho_k} \left(\frac{\alpha}{1 - \beta} \right) \\ = \rho_k^t \|\bar{\mathbf{x}}_k^{(0)} - \mathbf{x}_k^*\| + b_k \beta^t + c_k (t - 1) \\ \times \max\{\rho_k^{t-1}, \beta^{t-1}, \bar{\theta}^{t-1}\} + \left(\frac{c_k}{1 - \rho_k} + b_k \right) \left(\frac{\alpha}{1 - \beta} \right).$$

This proves that $\mathbf{x}_{i,k}^{(t)}$ converges to a neighborhood of $\mathbf{x}_k^* = \mathbf{q}_k$ or $\mathbf{x}_k^* = -\mathbf{q}_k$ at a linear rate. \square

It is noteworthy that if decaying step sizes α_t are used such that $\alpha_t \rightarrow 0$ as $t \rightarrow \infty$ (instead of constant α), the convergence will be exact but not linear. The rate in that case will be dominated by the rate of decay of α_t .

6. Experimental results

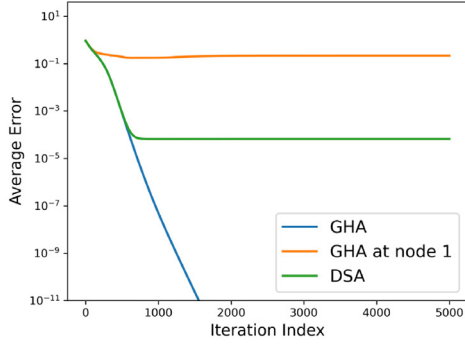
In this section, we provide results that demonstrate the efficacy of the proposed DSA algorithm. The need for collaboration between the nodes of a network is a vital part of any distributed algorithm, as already pointed out in Section 2. We first verify that necessity along with the effect of step size on DSA by performing some experiments. In these experiments, the weight matrix \mathbf{W} that conforms to the underlying graph topology is generated using the Metropolis constant edge-weight approach [40]. The performance of DSA in comparison to some baseline methods is also evaluated in additional experiments. We provide experimental results for DSA on synthetic and real data and compare the results with centralized generalized Hebbian algorithm (GHA) [11], centralized orthogonal iteration (OI) [14], distributed projected gradient descent (DPGD) and sequential distributed power method (SeqDistPM). For both the centralized methods, all the data is assumed to be at a single location with the difference being that GHA uses the Hebbian update whereas OI uses the well-known orthogonal iterations to estimate the top K eigenvectors of the covariance matrix \mathbf{C} . DPGD involves two significant steps per iteration. The first is a distributed gradient descent step at every node i given by $\sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_j + \alpha \nabla f_i(\mathbf{x}_i)$ as in Nedic and Ozdaglar [32] using trace maximization $f_i(\mathbf{x}_i) = \max \text{Trace}(\mathbf{x}_i^T \mathbf{C}_i \mathbf{x}_i)$ as the objective. This is followed by a projection step to ensure the orthogonality constraint $\mathbf{x}_i^T \mathbf{x}_i = \mathbf{I}$. The orthogonalization is accomplished using QR decomposition, an approach that ensures projection onto the Stiefel manifold [41] and whose computational complexity is $\mathcal{O}(K^2 d)$, at each node in each iteration. In contrast, SeqDistPM involves implementing the distributed power method [22,24] K times, estimating one eigenvector at a time and subtracting its impact on the covariance matrix for the estimation of subsequent eigenvectors. Note that SeqDistPM requires a finite T_c number of consensus iterations per iteration of the power method. Assuming the cost of communicating one $\mathbb{R}^{d \times K}$ matrix across the network from nodes to their neighbors to be one unit, the communication cost of SeqDistPM is T_c/K per iteration of the power method. The error metric used for comparison and reporting of the results is the average of the angles between the estimated and true eigenvectors, i.e., if $\mathbf{x}_{i,k}$ is the estimate of the k th eigenvector at i th node and \mathbf{q}_k is the true k th eigenvector then the average error across all nodes is calculated as follows:

$$E = \frac{1}{MK} \sum_{i=1}^M \sum_{k=1}^K \left(1 - \left(\frac{\mathbf{x}_{i,k}^T \mathbf{q}_k}{\|\mathbf{x}_{i,k}\|} \right)^2 \right). \quad (28)$$

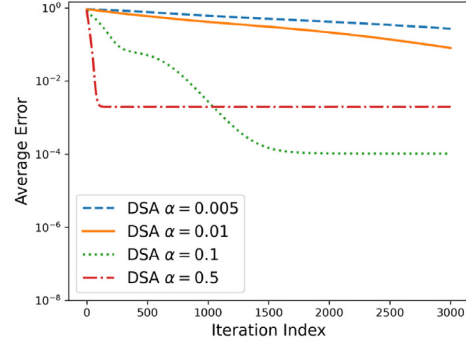
6.1. Synthetic data

We first show results that emphasize on the need for collaboration among the nodes. To that end, we generate $N = 10,000$ independent and identically distributed (i.i.d.) samples drawn from a multivariate Gaussian distribution with an eigengap $\Delta_K = \frac{\lambda_{K+1}}{\lambda_K} = 0.8$ and dimension $d = 10$. These samples are distributed equally among the $M = 10$ nodes of an Erdos-Renyi network (with connectivity probability $p = 0.5$), implying that each node has 1000 samples. The number of eigenvectors estimated is $K = 3$ and a constant step size of $\alpha = 0.1$ is used for this experiment. Fig. 1a shows the effect of using the GHA at a node without collaboration with other nodes versus DSA, which in simple terms embodies GHA + collaboration in the network. The blue line indicating GHA in the figure is the result of using all the data in a centralized manner. It is clear that the lack of any communication between nodes increases the error in estimation of the eigenvectors by a significant factor. In Fig. 1b, we use the same setup and parameters to show the effect of different step sizes on our proposed DSA algorithm. It is evident that if the step size is too low, the convergence becomes significantly slow, while if its high, the final error is larger. Hence, careful choice of the step size is required for DSA, as characterized by its convergence analysis.

Next, we compare DSA with the distributed methods of DPGD and SeqDistPM to demonstrate its communication efficiency. For that purpose, we generate synthetic data with different eigengaps $\Delta_K \in \{0.6, 0.8\}$. We simulate the distributed setup for Erdos-Renyi ($p = 0.5$), star and cycle graph topologies with $M = 10$ nodes. The data is generated so that each node has 1000 i.i.d samples ($N_i = 1000$) drawn from a multivariate Gaussian distribution for $d = 20$, i.e., the total samples generated are 10,000. The dimension of the subspace to be estimated is taken to be $K \in \{1, 5\}$. We use $T_c = 50$ as the number of consensus iterations per power iteration for SeqDistPM throughout our experiments. The results reported are an average of 10 Monte-Carlo trials. Fig. 2 shows the performance of different algorithms for the estimation of the most dominant eigenvector for different network topologies. It is clear that for $K = 1$ SeqDistPM outperforms both DSA and DPGD in terms of communications efficiency because it is basically distributed power method, which is shown in Raja and Bajwa [22], Raja and Bajwa [24] to have good performance for $K = 1$. Even though DSA and DPGD have the same performance in terms of communications cost, it is important to remember that DPGD requires an additional QR normalization step per communications round. Next, Fig. 3 shows a comparison between the three algorithms when the top-5 eigenvectors are estimated i.e., $K = 5$. It is clear that while estimating higher-order eigenvectors, DSA slightly outperforms DPGD without performing explicit QR normalization and it also has much better communications efficiency than SeqDistPM. The error for SeqDistPM is significantly high in the beginning because of the sequential estimation, which means that when the first (higher-order) eigenvector(s) is (are) being estimated, the lower-order estimates are still at their initial values and hence those contribute significant error even when the first or higher order terms have low error. After a sufficiently large number of communications rounds, SeqDistPM eventually does reach a lower final error compared to DSA. But this comes at the expense of slower convergence as a function of communications costs. It should also be noted that SeqDistPM lacks a formal convergence analysis and has two time scales that need to be adjusted as both contribute to the final error. Finally, the benefits of DSA over DPGD are twofold. First, DSA reaches similar or better error floor without explicit QR normalization, thus saving $\mathcal{O}(K^2 d)$ computations per iteration; and second, the convergence guarantees for gradient descent-based algorithms for non-convex problems like the PCA have limitations.

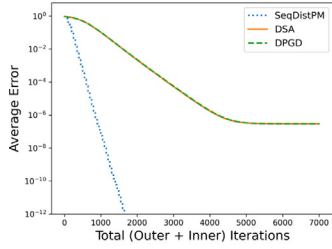


(a) Demonstration of the need for collaboration among the nodes in a network for the PCA problem

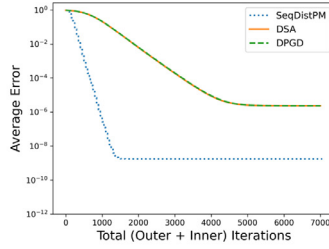


(b) Effect of varying the step size α on the performance of DSA

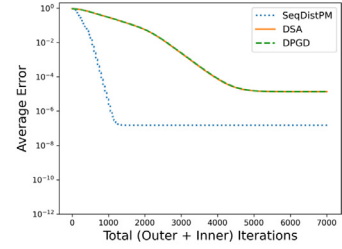
Fig. 1. The role of collaboration in the distributed PCA problem and the effect of changing the step size on the performance of DSA. The distributed setup corresponds to an Erdos-Renyi graph ($p = 0.5$) with $M = 10$ nodes, while the dimension of data is $d = 10$ and the number of estimated eigenvectors is $K = 3$.



(a) Erdos-Renyi network

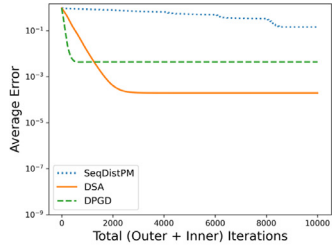


(b) Cyclic network

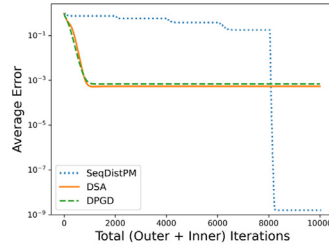


(c) Star network

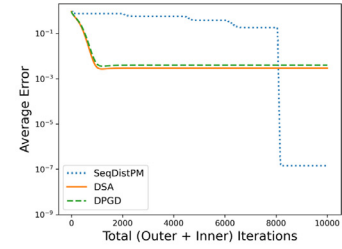
Fig. 2. Comparison between the performances of DSA, DPGD and SeqDistPM for $K = 1$ and $\Delta_K = 0.8$ in terms of communications efficiency, i.e., decrease in average estimation error as a function of the number of data units communicated throughout the network.



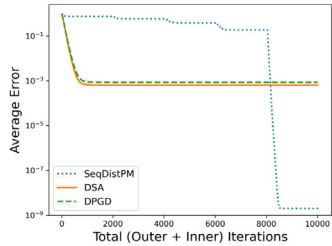
(a) Erdos-Renyi network, $\Delta_K = 0.6$



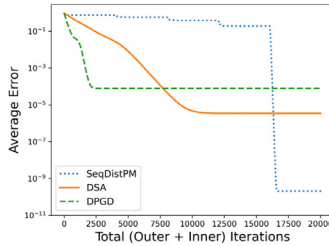
(b) Cyclic network, $\Delta_K = 0.6$



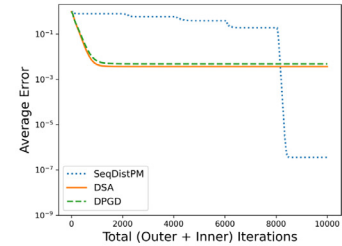
(c) Star network, $\Delta_K = 0.6$



(d) Erdos-Renyi network, $\Delta_K = 0.8$



(e) Cyclic network, $\Delta_K = 0.8$



(f) Star network, $\Delta_K = 0.8$

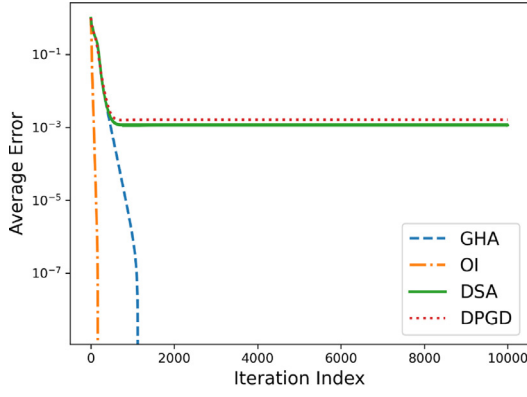
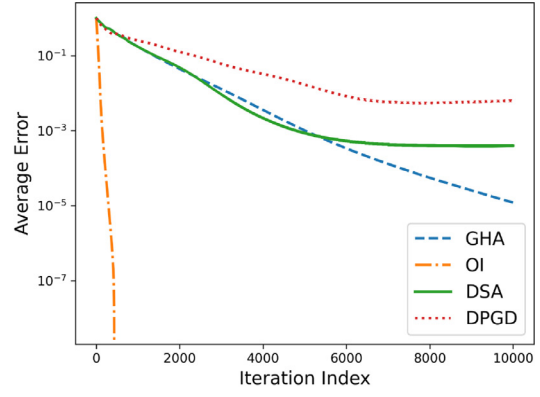
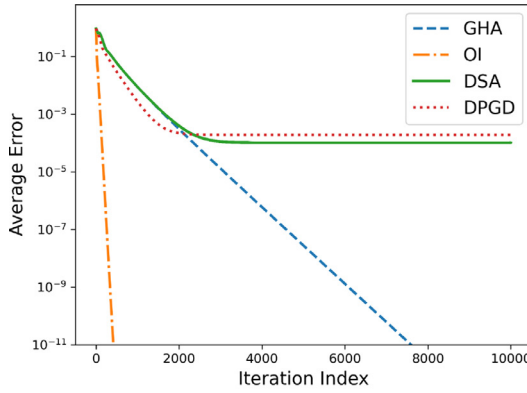
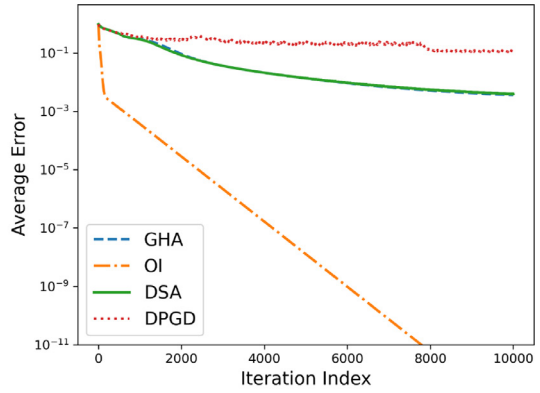
Fig. 3. Comparison between DSA, DPGD, and SeqDistPM for $K = 5$ in terms of communications efficiency.

The guarantees usually exist for convergence to a stationary solution with a sub-linear rate.

6.2. Real-world data

Along with the synthetic data experiments, we provide some experiments with real-world datasets of MNIST [42] and CIFAR-10 [43]. For the distributed setup in this case, we use an Erdos-

Renyi graph with $M = 20$ nodes and $p = 0.5$. Both the datasets have 60,000 samples, thereby making the number of samples per node to be $N_i = 3000$. The data dimension for MNIST is $d = 784$ and a constant step size of $\alpha = 0.1$ was used. The plots in Fig. 4a and b show the results for $K \in \{10, 40\}$ for MNIST. Similar plots are shown for CIFAR-10 in Fig. 5a and b, where the dimension d for CIFAR-10 is 1024, the number of estimated eigenvectors $K \in \{10, 20\}$ and a constant step size of $\alpha = 0.7$ is used. For these real-

(a) MNIST, $K = 10$ (b) MNIST, $K = 40$ **Fig. 4.** Comparison between DSA, OI, GHA, and DPGD for MNIST dataset as a function of the number of algorithmic iterations.(a) CIFAR-10, $K = 10$ (b) CIFAR-10, $K = 20$ **Fig. 5.** Comparison between DSA, OI, GHA, and DPGD for CIFAR-10 dataset as a function of the number of algorithmic iterations.

world data sets, we exclude the comparison with SeqDistPM as it is evident this method requires much higher cost of communications for estimating larger number of eigenvectors.

7. Conclusion

In this paper, we proposed and analyzed a new distributed Principal Component Analysis (PCA) algorithm that, as opposed to distributed subspace learning methods, facilitates both dimensionality reduction and data decorrelation in a distributed setup. Our main contribution in this regard was a detailed convergence analysis to prove that the proposed distributed method linearly converges to a neighborhood of the eigenvectors of the global covariance matrix. We also provided numerical results to demonstrate the communications efficiency and overall effectiveness of the proposed algorithm.

In terms of future work, an obvious extension would be a distributed algorithm that enables *exact* convergence to the PCA solution at a linear rate. Note that the use of a diminishing step size α along with the analysis in this paper already guarantees that DSA can converge exactly to the PCA solution. However, this exact convergence guarantee comes at the expense of a slow convergence rate. We instead expect to combine ideas from this work as well as ideas such as *gradient tracking* from the literature on distributed optimization [31,36,44] to develop a linearly convergent, exact algorithm for distributed PCA in the future. Another possible future direction involves developing an algorithm for distributed PCA that does not require the top K eigenvalues to be distinct. We also

leave the case of multiple eigenvector estimation from distributed, streaming data, as in Raja and Bajwa [26], for future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Statements and proofs of auxiliary lemmas

A1. Statement and proof of Lemma 1

Lemma 1. Assume $\|\mathbf{x}_k^{(0)}\| = 1, \forall k$. If the step size is bounded above as $\alpha \leq \frac{1}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, then

$$\forall t, \quad \|\mathbf{x}_k^{(t)}\| < \sqrt{3} \quad \text{and} \quad (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}. \quad (29)$$

Proof. From (11), we know the iterate for k th eigenvector estimate is

$$\begin{aligned} \mathbf{x}_k^{(t+1)} &= \mathbf{x}_k^{(t)} + \alpha \left(\mathbf{C} \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \mathbf{q}_p \mathbf{q}_p^T \mathbf{C} \mathbf{x}_k^{(t)} \right) \\ &= \mathbf{x}_k^{(t)} + \alpha \left(\mathbf{C} \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T \mathbf{x}_k^{(t)} \right) \\ &= \mathbf{x}_k^{(t)} + \alpha \left(\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)} \right), \end{aligned}$$

where $\tilde{\mathbf{C}}_k = \mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T$. Notice that $\tilde{\mathbf{C}}_k^2 = \mathbf{C}^2 - \sum_{p=1}^{k-1} \lambda_p^2 \mathbf{q}_p \mathbf{q}_p^T$. Hence,

$$\begin{aligned}
 \|\mathbf{x}_k^{(t+1)}\|^2 &= \|\mathbf{x}_k^{(t)} + \alpha(\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)})\|^2 \\
 &= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 \|\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)}\|^2 \\
 &\quad + 2\alpha (\mathbf{x}_k^{(t)})^T (\tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \mathbf{x}_k^{(t)}) \\
 &= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 ((\mathbf{x}_k^{(t)})^T \tilde{\mathbf{C}}_k^2 \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \|\mathbf{x}_k^{(t)}\|^2 \\
 &\quad - 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\mathbf{x}_k^{(t)})^T \tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)}) \\
 &\quad + 2\alpha ((\mathbf{x}_k^{(t)})^T \tilde{\mathbf{C}}_k \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \|\mathbf{x}_k^{(t)}\|^2) \\
 &= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 ((\mathbf{x}_k^{(t)})^T (\mathbf{C}^2 - \sum_{p=1}^{k-1} \lambda_p^2 \mathbf{q}_p \mathbf{q}_p^T) \mathbf{x}_k^{(t)} \\
 &\quad + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \|\mathbf{x}_k^{(t)}\|^2 - 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\mathbf{x}_k^{(t)})^T \\
 &\quad \times (\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T) \mathbf{x}_k^{(t)}) + 2\alpha ((\mathbf{x}_k^{(t)})^T \\
 &\quad \times (\mathbf{C} - \sum_{p=1}^{k-1} \lambda_p \mathbf{q}_p \mathbf{q}_p^T) \mathbf{x}_k^{(t)} - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \|\mathbf{x}_k^{(t)}\|^2) \\
 &= \|\mathbf{x}_k^{(t)}\|^2 + \alpha^2 ((\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 \\
 &\quad + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
 &\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2) \\
 &\quad + 2\alpha ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 - \|\mathbf{x}_k^{(t)}\|^2) - \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2).
 \end{aligned} \tag{30}$$

We now split our analysis into three cases based on the range of values of $\|\mathbf{x}_k^{(t)}\|^2$.

Case I: Let $\|\mathbf{x}_k^{(t)}\|^2 \leq 1$. Then we see from (30) that

$$\begin{aligned}
 \|\mathbf{x}_k^{(t+1)}\|^2 &\leq 1 + \alpha^2 (\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} \lambda_p) + 2\alpha \lambda_1 \\
 &\leq 1 + \alpha^2 (\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} \lambda_1) + 2\alpha \lambda_1 \\
 &\leq 1 + \alpha^2 \lambda_1^2 (2K - 1) + 2\alpha \lambda_1 \sqrt{2K - 1} \\
 &= (1 + \alpha \lambda_1 \sqrt{2K - 1})^2 \\
 &\leq 2(1 + \alpha^2 \lambda_1^2 (2K - 1)) \leq 2(1 + \frac{1}{9(2K - 1)})
 \end{aligned}$$

$$\leq 2(1 + \frac{1}{9}) < 3.$$

Case II: Now suppose $1 < \|\mathbf{x}_k^{(t)}\|^2 \leq 2$. Then from (30) we have

$$\begin{aligned}
 \|\mathbf{x}_k^{(t+1)}\|^2 &\leq 2 + \alpha^2 (2\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} 2\lambda_p) \\
 &\leq 2 + \alpha^2 (2\lambda_1^2 + 2\lambda_1 \sum_{p=1}^{k-1} 2\lambda_1) \\
 &\leq 2(1 + \frac{1}{9(2K - 1)}) \leq 2(1 + \frac{1}{9}) < 3,
 \end{aligned}$$

using similar steps as Case I.

Case III: Finally suppose $2 < \|\mathbf{x}_k^{(t)}\|^2 < 3$. Then from (30) we get

$$\begin{aligned}
 \|\mathbf{x}_k^{(t+1)}\|^2 &< 3 + \alpha^2 ((\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 \\
 &\quad + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
 &\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2) \\
 &\quad + 2\alpha ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 - \|\mathbf{x}_k^{(t)}\|^2) - \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2).
 \end{aligned}$$

To show that $\|\mathbf{x}_k^{(t+1)}\|^2 < 3$, we have to show

$$\begin{aligned}
 &\alpha^2 ((\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
 &\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2) + 2\alpha ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 - \|\mathbf{x}_k^{(t)}\|^2) - \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2) \leq 0 \\
 \Leftrightarrow \alpha &\leq \frac{2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1) + 2 \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2}{(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2}.
 \end{aligned} \tag{31}$$

We now find a lower bound of the right hand side of (31). Note that

$$\begin{aligned}
 &2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1) + 2 \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 \\
 &\geq 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)
 \end{aligned} \tag{32}$$

$$\begin{aligned}
 \text{and } &(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} - \sum_{p=1}^{k-1} \lambda_p^2 (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
 &\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2 \\
 &\leq (\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
 &\quad + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p (\mathbf{q}_p^T \mathbf{x}_k^{(t)})^2.
 \end{aligned} \tag{33}$$

Now, $\frac{(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}}{(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)}}$ is a generalized Rayleigh quotient whose maximum and minimum values are the largest and smallest eigenvalues of the generalized eigenvalue problem $\mathbf{C} \mathbf{y} = \lambda \mathbf{C}^2 \mathbf{y}$. Since the eigenvectors of \mathbf{C} and \mathbf{C}^2 are the same, the largest and smallest eigenvalues of the generalized problems are $\frac{1}{\lambda_d}$ and $\frac{1}{\lambda_1}$, respectively, where λ_1 and λ_d are the largest and smallest eigenvalues of \mathbf{C} . Thus, $(\mathbf{x}_k^{(t)})^T \mathbf{C}^2 \mathbf{x}_k^{(t)} \leq \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}$. Also, since $\mathbf{q}_p^T \mathbf{x}_k^{(t)} \leq \|\mathbf{q}\| \|\mathbf{x}_k^{(t)}\|$, we have the right hand side of (33)

$$\begin{aligned}
& \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} + ((\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) \\
& + 2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \sum_{p=1}^{k-1} \lambda_p \|\mathbf{x}_k^{(t)}\|^2 \\
& = (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\lambda_1 + (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2 \sum_{p=1}^{k-1} \lambda_p \|\mathbf{x}_k^{(t)}\|^2) \\
& \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\lambda_1 + \lambda_1 \|\mathbf{x}_k^{(t)}\|^2 (\|\mathbf{x}_k^{(t)}\|^2 - 2) + 2 \sum_{p=1}^{k-1} \lambda_1 \|\mathbf{x}_k^{(t)}\|^2) \\
& = \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (1 + \|\mathbf{x}_k^{(t)}\|^4 - 2\|\mathbf{x}_k^{(t)}\|^2 + 2(k-1)\|\mathbf{x}_k^{(t)}\|^2) \\
& = \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} ((\|\mathbf{x}_k^{(t)}\|^2 - 1)^2 + 2(k-1)\|\mathbf{x}_k^{(t)}\|^2) \\
& = \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)((\|\mathbf{x}_k^{(t)}\|^2 - 1) + 2(k-1)) \\
& \quad \times \frac{\|\mathbf{x}_k^{(t)}\|^2}{(\|\mathbf{x}_k^{(t)}\|^2 - 1)} \\
& < \lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)((3-1) + 2(k-1)2), \\
& \quad \text{since } \frac{\|\mathbf{x}_k^{(t)}\|^2}{(\|\mathbf{x}_k^{(t)}\|^2 - 1)} < 2 \\
& = 2\lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)(2k-1) \\
& \leq 2\lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)(2K-1).
\end{aligned}$$

Hence, we have that the right hand side of (31) exceeds

$$\frac{2(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)}{2\lambda_1 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} (\|\mathbf{x}_k^{(t)}\|^2 - 1)(2K-1)} = \frac{1}{\lambda_1(2K-1)} > \frac{1}{3\lambda_1(2K-1)}.$$

Thus, if $\alpha \leq \frac{1}{3\lambda_1(2K-1)}$, then $\|\mathbf{x}_k^{(t)}\|^2 < 3$.

Next,

$$0 \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \leq \lambda_1 \|\mathbf{x}_k^{(t)}\|^2 < 3\lambda_1 \leq 3(2K-1)\lambda_1 \leq \frac{1}{\alpha}. \quad (34)$$

Hence, $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$. \square

A2. Statement and proof of Lemma 2

Lemma 2. Suppose $\mathbf{q}_k^T \mathbf{x}_k^{(0)} = z_{k,k}^{(0)} \neq 0$ and $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$, then

$$(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} > \min\{(1 - 3\alpha\lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\}, \quad \forall t.$$

Proof. We know $0 \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \leq \lambda_1 \|\mathbf{x}_k^{(t)}\|^2 < 3\lambda_1$ using Lemma 1. Let $\lambda_m, m > K$ be the smallest non-zero eigenvalue of \mathbf{C} . Now, if $\lambda_m \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < 3\lambda_1$, then

$$\begin{aligned}
(\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)} &= \sum_{l=1}^d \lambda_l (z_{k,l}^{(t+1)})^2 \\
&= \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t+1)})^2 + \sum_{l=k}^d \lambda_l (z_{k,l}^{(t+1)})^2 \\
&= \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2
\end{aligned}$$

$$\begin{aligned}
& + \sum_{l=k}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\
& \geq (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 \\
& + (1 + \alpha(\lambda_m - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 \sum_{l=k}^d \lambda_l (z_{k,l}^{(t)})^2 \\
& > (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=1}^{k-1} \lambda_l (z_{k,l}^{(t)})^2 \\
& + (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=k}^d \lambda_l (z_{k,l}^{(t)})^2 \\
& = (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)})^2 \sum_{l=1}^d \lambda_l (z_{k,l}^{(t)})^2 \\
& > (1 - 3\alpha\lambda_1)^2 (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} \geq (1 - 3\alpha\lambda_1)^2 \lambda_m. \quad (35)
\end{aligned}$$

Also, from (12), we have $\mathbf{x}_k^{(t)} = \sum_{l=1}^d z_{k,l}^{(t)} \mathbf{q}_l = \sum_{l=1}^{k-1} z_{k,l}^{(t)} \mathbf{q}_l + \sum_{l=k}^d z_{k,l}^{(t)} \mathbf{q}_l$. Let $\sum_{l=1}^{k-1} z_{k,l}^{(t)} \mathbf{q}_l = \mathbf{x}'^{(t)}$ and $\sum_{l=k}^d z_{k,l}^{(t)} \mathbf{q}_l = \tilde{\mathbf{x}}_k^{(t)}$. Thus, $(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} = (\tilde{\mathbf{x}}_k^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t)} + (\mathbf{x}'^{(t)})^T \mathbf{C} \mathbf{x}'^{(t)}$.

Now, if $(\tilde{\mathbf{x}}_k^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t)} \leq (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} < \lambda_m$ then

$$\begin{aligned}
(\mathbf{x}_k^{(t+1)})^T \mathbf{C} \mathbf{x}_k^{(t+1)} &\geq (\tilde{\mathbf{x}}_k^{(t+1)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t+1)} \\
&= \sum_{l=k}^d \lambda_l (z_{k,l}^{(t+1)})^2 \\
&= \sum_{l=k}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\
&\geq (1 + \alpha(\lambda_m - (\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)}))^2 \sum_{l=k}^d \lambda_l (z_{k,l}^{(t)})^2 > (\tilde{\mathbf{x}}_k^{(t)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(t)}. \quad (36)
\end{aligned}$$

Combining (35) and (36), we have

$$(\mathbf{x}_k^{(t)})^T \mathbf{C} \mathbf{x}_k^{(t)} > \min\{(1 - 3\alpha\lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C} \tilde{\mathbf{x}}_k^{(0)}\}. \quad (37)$$

\square

A3. Statement and proof of Lemma 3

Lemma 3. Assume $\|\mathbf{x}_{i,k}^{(0)}\| = 1$. If the step size is bounded above as $\alpha \leq \frac{w_{ii}}{3\lambda_1(2K-1)}$, where λ_1 is the largest eigenvalue of \mathbf{C} and K is the number of eigenvectors to be estimated, then

$$\|\mathbf{x}_{i,k}^{(t)}\| < \sqrt{3} \quad \text{and} \quad (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} < \frac{1}{\alpha}, \quad \forall k, t. \quad (38)$$

Proof. We have

$$\begin{aligned}
\mathbf{x}_{i,k}^{(t+1)} &= \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \mathbf{x}_{j,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}) \\
&\quad - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}. \quad (39)
\end{aligned}$$

Hence,

$$\begin{aligned}
\|\mathbf{x}_{i,k}^{(t+1)}\| &\leq \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| \\
&\quad + \alpha \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} \|w_{ij} \mathbf{x}_{j,k}^{(t)}\| \\
&\leq \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\|
\end{aligned}$$

$$\begin{aligned}
& + \alpha \sum_{p=1}^{k-1} \lambda_1 \|(\mathbf{x}_{i,p}^{(t)})\| \|\mathbf{x}_{i,k}^{(t)}\| \|\mathbf{x}_{i,p}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \\
& = \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| \\
& + \alpha \sum_{p=1}^{k-1} \lambda_1 \|(\mathbf{x}_{i,p}^{(t)})\|^2 \|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \\
& \leq \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| \\
& + 3\alpha \lambda_1 \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \\
& = \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| \\
& + 3(k-1)\alpha \lambda_1 \|\mathbf{x}_{i,k}^{(t)}\| + \sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\|.
\end{aligned}$$

Now,

$$\begin{aligned}
& \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}) \mathbf{x}_{i,k}^{(t)})\|^2 \\
& = w_{ii}^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 + \alpha^2 \|\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}) \mathbf{x}_{i,k}^{(t)}\|^2 \\
& + 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}) \\
& = w_{ii}^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 + 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) \\
& + \alpha^2 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + \alpha^2 ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2).
\end{aligned}$$

Case I: Let us assume $\|\mathbf{x}_{i,k}^{(t)}\|^2 \leq 1, \forall i$. Then, we have

$$\begin{aligned}
& \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\|^2 \\
& \leq (w_{ii} + \alpha \lambda_1)^2 \leq \left(w_{ii} + \frac{w_{ii}}{3(2K-1)}\right)^2.
\end{aligned}$$

Thus,

$$\begin{aligned}
\|\mathbf{x}_{i,k}^{t+1}\| & \leq w_{ii} \left(1 + \frac{1}{3(2K-1)}\right) + \frac{3(k-1)}{3(2K-1)} + (1 - w_{ii}) \\
& < \frac{1}{3(2K-1)} + \frac{k-1}{2K-1} + 1 = \frac{k-0.67}{2(K-0.5)} + 1 \\
& \leq \frac{K-0.67}{2(K-0.5)} + 1 < 1.5 < \sqrt{3}.
\end{aligned}$$

Case II: Now, suppose $1 \leq \|\mathbf{x}_{i,k}^{(t)}\|^2 < 2, \forall i$. Then, we get

$$\begin{aligned}
& \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\|^2 \leq 2w_{ii}^2 + 2\alpha^2 \lambda_1^2 \\
& < 2(w_{ii} + \alpha \lambda_1)^2.
\end{aligned}$$

Thus, if we need $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$, the following condition should be met:

$$\begin{aligned}
\|\mathbf{x}_{i,k}^{t+1}\| & \leq \sqrt{2} w_{ii} (1 + \alpha \lambda_1) + 3(k-1)\alpha \lambda_1 \sqrt{2} + (1 - w_{ii}) \sqrt{2} \leq \sqrt{3} \\
& \Leftrightarrow \sqrt{2} + \sqrt{2} w_{ii} \alpha \lambda_1 + 3(k-1)\alpha \lambda_1 \sqrt{2} \leq \sqrt{3} \\
& \Leftrightarrow \sqrt{2} \alpha \lambda_1 + 3(k-1)\alpha \lambda_1 \sqrt{2} \leq \sqrt{3} - \sqrt{2} \\
& \Leftrightarrow \sqrt{2} \alpha \lambda_1 (3k-2) \leq \sqrt{3} - \sqrt{2} \\
& \Leftrightarrow \sqrt{2} \alpha \lambda_1 (3K-2) \leq \sqrt{3} - \sqrt{2} \\
& \Leftrightarrow \alpha \leq \frac{\sqrt{3} - \sqrt{2}}{\sqrt{2} \lambda_1 (3K-2)} = \frac{\sqrt{1.5} - 1}{\lambda_1 (3K-2)} = \frac{0.225}{\lambda_1 (3K-2)}.
\end{aligned}$$

Since $\frac{0.225}{\lambda_1 3(2K-1)} < \frac{0.225}{\lambda_1 3(2K-2)}$, if $\alpha \leq \frac{0.225}{3\lambda_1 (2K-1)}$, then $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$.

Case III: Finally, suppose $2 \leq \|\mathbf{x}_{i,k}^{(t)}\|^2 \leq 3, \forall i$. We then have the following: $\sum_{j \neq i} w_{ij} \|\mathbf{x}_{j,k}^{(t)}\| \leq \sum_{j \neq i} w_{ij} \sqrt{3} = (1 - w_{ii}) \sqrt{3}$.

Now, if we desire $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$, then we need

$$\|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| + 3(k-1)\alpha \lambda_1 \|\mathbf{x}_{i,k}^{(t)}\|$$

$$+ \sum_{j \neq i} w_{ij} \sqrt{3} \leq \sqrt{3}$$

$$\begin{aligned}
& \Leftrightarrow \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| \\
& + 3(k-1)\alpha \lambda_1 \|\mathbf{x}_{i,k}^{(t)}\| + (1 - w_{ii}) \sqrt{3} \leq \sqrt{3} \\
& \Leftrightarrow \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\| \leq \sqrt{3} \\
& - 3(k-1)\alpha \lambda_1 \|\mathbf{x}_{i,k}^{(t)}\| - (1 - w_{ii}) \sqrt{3} \\
& \Leftrightarrow \|w_{ii} \mathbf{x}_{i,k}^{(t)} + \alpha (\mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)})\|^2 \leq 3w_{ii}^2 \\
& - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2.
\end{aligned}$$

Therefore, we need

$$\begin{aligned}
& w_{ii}^2 \|\mathbf{x}_{i,k}^{(t)}\|^2 + 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) + \alpha^2 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} \\
& + \alpha^2 ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
& \leq 3w_{ii}^2 - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \Leftrightarrow 3w_{ii}^2 + 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) \\
& + \alpha^2 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + \alpha^2 ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
& \leq 3w_{ii}^2 - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \Leftrightarrow 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (1 - \|\mathbf{x}_{i,k}^{(t)}\|^2) \\
& + \alpha^2 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + \alpha^2 ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
& \leq -6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| + 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \Leftrightarrow \alpha^2 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + \alpha^2 ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
& - 9\alpha^2\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \leq 2\alpha w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 6\sqrt{3}(k-1)w_{ii}\alpha\lambda_1\|\mathbf{x}_{i,k}^{(t)}\| \\
& \Leftrightarrow \alpha \leq \frac{2w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 6\sqrt{3}(k-1)w_{ii}\lambda_1\|\mathbf{x}_{i,k}^{(t)}\|}{(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2}. \quad (40)
\end{aligned}$$

We now find the lower bound of the right-hand side of (40). Note that

$$\begin{aligned}
& (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \leq \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
& - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2, \\
& \text{since } (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i^2 \mathbf{x}_{i,k}^{(t)} \leq \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \\
& \leq \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \|\mathbf{x}_{i,k}^{(t)}\|^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\
& - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& = \lambda_1 (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1)^2 - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& \leq \lambda_1 (k-1) (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1)^2 - 9\lambda_1^2(k-1)^2\|\mathbf{x}_{i,k}^{(t)}\|^2 \\
& < \lambda_1 (k-1) (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) \left((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9\lambda_1 (k-1) \right)
\end{aligned}$$

$$\text{since } \frac{\|\mathbf{x}_{i,k}^{(t)}\|^2}{\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1} > 1$$

and,

$$2w_{ii} (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 6\sqrt{3}(k-1)w_{ii}\lambda_1\|\mathbf{x}_{i,k}^{(t)}\|$$

$$\begin{aligned} &\geq 2w_{ii}(\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 18(k-1)w_{ii}\lambda_1 \\ &= 2w_{ii}((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9(k-1)\lambda_1). \end{aligned}$$

Thus, we have that the right hand side of (40) exceeds

$$\begin{aligned} &\frac{2w_{ii}((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9(k-1)\lambda_1)}{\lambda_1(k-1)(\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) \left((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 1) - 9\lambda_1(k-1) \right)} \\ &= \frac{2w_{ii}}{\lambda_1(k-1)} > \frac{w_{ii}}{3\lambda_1(2K-1)}. \end{aligned}$$

This proves if $\alpha \leq \min\{\frac{w_{ii}}{3\lambda_1(2K-1)}, \frac{0.225}{3\lambda_1(2K-1)}\}$ then $\|\mathbf{x}_{i,k}^{(t+1)}\| \leq \sqrt{3}$. \square

A4. Statement and proof of Lemma 4

Lemma 4. The norm of Sanger's direction $\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})$ is bounded as

$$\|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 \leq 3\lambda_{i,1}^2(3k-2)(3k+1), \forall k = 1, \dots, K. \quad (41)$$

Proof. We know

$$\begin{aligned} \mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) &= \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \\ &= (\mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T) \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} \\ &= \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} \\ \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 &= \|\tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)}\|^2 \\ &= (\mathbf{x}_{i,k}^{(t)})^T (\tilde{\mathbf{C}}_i^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\ &\quad + (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)}. \end{aligned}$$

Next, notice that $\|\tilde{\mathbf{C}}_i^{(t)}\| = \|\mathbf{C}_i - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\|$. Thus,

$$\begin{aligned} \|\tilde{\mathbf{C}}_i^{(t)}\| &\leq \|\mathbf{C}_i\| + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T\| \|\mathbf{C}_i\| \leq \lambda_{i,1} + \sum_{p=1}^{k-1} 3\lambda_{i,1} \\ &= \lambda_{i,1} + 3(k-1)\lambda_{i,1} = \lambda_{i,1}(3k-2). \end{aligned}$$

We, therefore get,

$$\begin{aligned} (\mathbf{x}_{i,k}^{(t)})^T (\tilde{\mathbf{C}}_i^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} &\leq \lambda_{\max}((\tilde{\mathbf{C}}_i^{(t)})^T) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} \\ &= \|(\tilde{\mathbf{C}}_i^{(t)})\| \|(\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)}\| \leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)}. \end{aligned}$$

Thus,

$$\begin{aligned} \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} \\ &\quad + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\ &\quad + (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \\ &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \tilde{\mathbf{C}}_i^{(t)} \mathbf{x}_{i,k}^{(t)} + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) \\ &\quad + (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} \end{aligned}$$

$$\begin{aligned} \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})\|^2 &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T (\mathbf{I} - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T) \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \\ &\quad + ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) + \\ &\quad (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} \\ &\leq \lambda_{i,1}(3k-2) (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} + \lambda_{i,1}(3k-2) \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} + \\ &\quad ((\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)})^2 (\|\mathbf{x}_{i,k}^{(t)}\|^2 - 2) + (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \sum_{p=1}^{k-1} \|\mathbf{x}_{i,k}^{(t)}\|^2 \|\mathbf{x}_{i,p}^{(t)}\|^2 \lambda_{i,1} \\ &\leq \lambda_{i,1}(3k-2) 3\lambda_{i,1} + \lambda_{i,1}(3k-2) \sum_{p=1}^{k-1} 9\lambda_{i,1} + 9\lambda_{i,1}^2 + \lambda_{i,1} 3 \sum_{p=1}^{k-1} 9\lambda_{i,1} \\ &= 3\lambda_{i,1}^2(3k-2)(3k+1). \end{aligned}$$

\square

Appendix B. Statement and proof of Lemma 5

Lemma 5. Let $\eta = \min\{(1 - 3\alpha\lambda_1)^2\lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C}_k \tilde{\mathbf{x}}_k^{(0)}\}$. Now, suppose $\eta < (\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$, then the following is true for $\gamma = 1 - \alpha\eta$, and some constant $a_1 > 0$:

$$\sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 < a_1 \gamma^{t+1}. \quad (42)$$

Proof. For $l = 1, \dots, k-1$, we know from (13)

$$z_{k,l}^{(t+1)} = (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)}) z_{k,l}^{(t)}$$

$$\text{or, } (z_{k,l}^{(t+1)})^2 = (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2.$$

Let $\min\{(1 - 3\alpha\lambda_1)^2\lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C}_k \tilde{\mathbf{x}}_k^{(0)}\} = \eta$. Since $0 < \eta < (\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$ (from (34) and (37)), we have $0 < 1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)} < 1 - \alpha\eta < 1$. Therefore,

$$\begin{aligned} \sum_{l=1}^{k-1} (z_{k,l}^{(t+1)})^2 &< \sum_{l=1}^{k-1} \gamma (z_{k,l}^{(t)})^2 < \gamma^{t+1} \sum_{l=1}^{k-1} (z_{k,l}^{(0)})^2 \\ &= a_1 \gamma^{t+1}, \text{ where } \gamma = (1 - \alpha\eta)^2. \end{aligned} \quad (43)$$

\square

Appendix C. Statement and proof of Lemma 6

Lemma 6. Suppose $z_{k,k}^{(0)} \neq 0$ and $(\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$. Then the following is true for $\rho_k = \left(\frac{1+\alpha\lambda_{k+1}}{1+\alpha\lambda_k}\right)^2 < 1$ and some constant $a_2 > 0$:

$$\sum_{l=k+1}^d (z_{k,l}^{(t+1)})^2 \leq a_2 \rho_k^{t+1}. \quad (44)$$

Proof. For $l = k, \dots, d$ we know from (14) that $z_{k,l}^{(t+1)} = (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)})) z_{k,l}^{(t)}$. If $(\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)} < \frac{1}{\alpha}$, we have $1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}_k \mathbf{x}_k^{(t)}) > \alpha\lambda_l \geq 0, \forall l = k, \dots, d$.

Thus, we have for $l = k+1, \dots, d$,

$$\begin{aligned} \left(\frac{z_{k,l}^{(t+1)}}{z_{k,k}^{(t+1)}} \right)^2 &= \left(\frac{1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})}{1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})} \right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}} \right)^2 \\ &= \left(1 - \frac{\alpha(\lambda_k - \lambda_l)}{1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})} \right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}} \right)^2 \\ &\leq \left(1 - \frac{\alpha(\lambda_k - \lambda_l)}{1 + \alpha\lambda_k} \right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}} \right)^2 \\ &= \left(\frac{1 + \alpha\lambda_l}{1 + \alpha\lambda_k} \right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}} \right)^2 \leq \left(\frac{1 + \alpha\lambda_{k+1}}{1 + \alpha\lambda_k} \right)^2 \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}} \right)^2 \\ &= \rho_k \left(\frac{z_{k,l}^{(t)}}{z_{k,k}^{(t)}} \right)^2, \quad \rho_k = \left(\frac{1 + \alpha\lambda_{k+1}}{1 + \alpha\lambda_k} \right)^2 < 1. \end{aligned}$$

Therefore, for $l = k+1, \dots, d$, $(z_{k,l}^{(t+1)})^2 \leq \rho_k^{t+1} \left(\frac{z_{k,l}^{(0)}}{z_{k,k}^{(0)}} \right)^2 (z_{k,k}^{(t+1)})^2$.

Since $\|\mathbf{x}_k^{(t+1)}\|^2 \leq 3$ and $\|\mathbf{x}_k^{(0)}\| = 1$, hence $(z_{k,k}^{(t+1)})^2 \leq 3$ and $z_{k,l}^{(0)} \leq 1$. Also, because of the assumption $z_{k,k}^{(0)} \neq 0$, let us assume $(z_{k,k}^{(0)})^2 > \tilde{\eta}$. Thus, we can write

$$\sum_{l=k+1}^d (z_{k,l}^{(t+1)})^2 \leq \rho_k^{t+1} \sum_{l=k+1}^d \frac{3}{\tilde{\eta}} = a_2 \rho_k^{t+1}. \quad (45)$$

□

Appendix D. Statement and proof of Lemma 7

Lemma 7. Suppose $(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} < \frac{1}{\alpha}$ and $(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} > \min\{(1 - 3\alpha\lambda_1)^2 \lambda_m, (\tilde{\mathbf{x}}_k^{(0)})^T \mathbf{C}\mathbf{x}_k^{(0)}\}$. Then there exists constants $0 < \delta, \gamma_1 < 1, a_4 > 0$ such that

$$|\lambda_k - (\mathbf{x}_k^{(t+1)})^T \mathbf{C}\mathbf{x}_k^{(t+1)}| \leq ta_4(\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}). \quad (46)$$

Proof.

$$\begin{aligned} (\mathbf{x}_k^{(t+1)})^T \mathbf{C}\mathbf{x}_k^{(t+1)} &= \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=k}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &= \sum_{l=1}^{k-1} \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=k}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 \\ &\quad - \sum_{l=1}^{k-1} \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad - \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &= \sum_{l=1}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 + P^{(t)} \\ &= (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} + P^{(t)}, \end{aligned}$$

where

$$\begin{aligned} P^{(t)} &= \sum_{l=1}^{k-1} \lambda_l (1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 (z_{k,l}^{(t)})^2 \\ &\quad - \sum_{l=1}^{k-1} \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad + \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2 \\ &\quad - \sum_{l=k+1}^d \lambda_l (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (z_{k,l}^{(t)})^2. \end{aligned}$$

Now,

$$\begin{aligned} \lambda_k - (\mathbf{x}_k^{(t+1)})^T \mathbf{C}\mathbf{x}_k^{(t+1)} &= \lambda_k - (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - P^{(t)} \\ &= \lambda_k - (1 + \alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 \\ &\quad + 2\alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - P^{(t)} \\ &= \lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - (\alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 \\ &\quad + 2\alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - P^{(t)} \\ &= \lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - (\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(\alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})) \\ &\quad + 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - P^{(t)} \\ &= (\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(1 - \alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}) \\ &\quad - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} - P^{(t)}. \end{aligned}$$

Let us denote $V^{(t)} = |\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}|$. Then,

$$\begin{aligned} V^{(t+1)} &\leq V^{(t)} |1 - \alpha^2(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} \\ &\quad - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}| + |P^{(t)}| \\ &\leq V^{(t)} \max\{(1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2, \alpha^2\lambda_k(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}\} + |P^{(t)}|. \end{aligned}$$

Also from (34) and (37), $0 < \alpha\eta < \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} < 1$ and $\alpha^2\lambda_k(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)} < \alpha\lambda_k$. Denote $\delta = \max\{(1 - \alpha\eta)^2, \alpha\lambda_k\}$. Since $\alpha\lambda_k < \alpha\lambda_1 < 1$, hence $0 < \delta < 1$. Thus,

$$V^{(t+1)} \leq \delta V^{(t)} + |P^{(t)}|. \quad (47)$$

Next, we bound $|P^{(t)}|$ as follows:

$$\begin{aligned} |P^{(t)}| &= \left| \sum_{l=1}^{k-1} \lambda_l ((1 - \alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})^2 \right. \\ &\quad \left. - (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2) (z_{k,l}^{(t)})^2 \right. \\ &\quad \left. + \sum_{l=k+1}^d \lambda_l ((1 + \alpha(\lambda_l - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2 \right. \\ &\quad \left. - (1 + \alpha(\lambda_k - (\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)}))^2) (z_{k,l}^{(t)})^2 \right| \\ &= \left| \sum_{l=1}^{k-1} \lambda_l (-\alpha\lambda_k)(2 + \alpha\lambda_k - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(z_{k,l}^{(t)})^2 \right. \\ &\quad \left. + \sum_{l=k+1}^d \lambda_l \alpha(\lambda_l - \lambda_k)(2 + \alpha(\lambda_k + \lambda_l) - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(z_{k,l}^{(t)})^2 \right| \\ &\leq \sum_{l=1}^{k-1} \lambda_l |(-\alpha\lambda_k)(2 + \alpha\lambda_k - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(z_{k,l}^{(t)})^2| \\ &\quad + \sum_{l=k+1}^d \lambda_l |\alpha(\lambda_l - \lambda_k)(2 + \alpha(\lambda_k + \lambda_l) - 2\alpha(\mathbf{x}_k^{(t)})^T \mathbf{C}\mathbf{x}_k^{(t)})(z_{k,l}^{(t)})^2| \\ &\leq \sum_{l=1}^{k-1} \lambda_l \alpha\lambda_k (2 + \alpha\lambda_k) (z_{k,l}^{(t)})^2 \end{aligned}$$

$$\begin{aligned}
& + \sum_{l=k+1}^d \lambda_l \alpha (\lambda_k - \lambda_l) (2 + \alpha (\lambda_k + \lambda_l)) (z_{k,l}^{(t)})^2 \\
& < \sum_{l=1}^{k-1} \lambda_l \alpha \lambda_k (2 + \alpha \lambda_k) (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d \lambda_l (2\alpha \lambda_k + \alpha^2 \lambda_k^2) (z_{k,l}^{(t)})^2 \\
& < \sum_{l=1}^{k-1} 3\lambda_1 (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d 3\lambda_1 (z_{k,l}^{(t)})^2,
\end{aligned}$$

since $\alpha \lambda_k < 1$ and $\lambda_l < \lambda_1$

$$= 3\lambda_1 \left(\sum_{l=1}^{k-1} (z_{k,l}^{(t)})^2 + \sum_{l=k+1}^d (z_{k,l}^{(t)})^2 \right) < 3\lambda_1 (a_1 \gamma^t + a_2 \rho_k^t)$$

using Lemma 5 and 6

$$\leq a_3 \gamma_1^t, \quad \text{where } a_3 = \max\{3\lambda_1 a_1, 3\lambda_1 a_2\}$$

and $\gamma_1 = \max\{\gamma, \rho_k\}$.

So from (47), we have $V^{(t+1)} \leq \delta V^{(t)} + a_3 \gamma_1^t \leq \delta^{t+1} V^{(0)} + a_3 \sum_{r=0}^t (\delta \gamma_1^{-1})^r \gamma_1^t$. Since $\gamma_1, \delta < 1$, we have the following two cases:

1. $\delta \leq \gamma_1 \Rightarrow \delta \gamma_1^{-1} \leq 1$. Then, $\sum_{r=0}^t (\delta \gamma_1^{-1})^r \gamma_1^t \leq \sum_{r=0}^t \gamma_1^t = t \gamma_1^t$.
2. $\delta > \gamma_1$. Then $\sum_{r=0}^t (\delta \gamma_1^{-1})^r \gamma_1^t = \gamma_1^t + \delta \gamma_1^{t-1} + \dots + \delta^t < \delta^t + \dots + \delta^t = t \delta^t$.

Thus,

$$V^{(t+1)} \leq \delta^{t+1} V^{(0)} + t a_3 \max\{\delta^t, \gamma_1^t\} \leq t a_4 (\delta^{t+1} + \max\{\delta^t, \gamma_1^t\}),$$

where $a_4 = \max\{V^{(0)}, a_3\}$. \square

Appendix E. Statement and proof of Lemma 8

Lemma 8. The deviation of an iterate at a node from the average is bounded from above as

$$\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq b_k (\beta^t + \frac{\alpha}{1-\beta}), \quad \forall k = 1, \dots, K, \quad (48)$$

where β is the second largest magnitude of the eigenvalues of \mathbf{W} given as $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_M(\mathbf{W})|\} < 1$ and $b_k > 0$ is some constant.

Proof. We stack the iterates $\mathbf{x}_{i,k}^{(t)}$ and $\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)})$ as

$$\begin{aligned}
\mathbf{x}_k^{(t)} &= \begin{bmatrix} \mathbf{x}_{1,k}^{(t)} \\ \mathbf{x}_{2,k}^{(t)} \\ \vdots \\ \mathbf{x}_{M,k}^{(t)} \end{bmatrix} \in \mathbb{R}^{Md} & \mathcal{H}(\mathbf{x}_k^{(t)}) &= \begin{bmatrix} \mathcal{H}_1(\mathbf{x}_{1,k}^{(t)}) \\ \mathcal{H}_2(\mathbf{x}_{2,k}^{(t)}) \\ \vdots \\ \mathcal{H}_M(\mathbf{x}_{M,k}^{(t)}) \end{bmatrix} \in \mathbb{R}^{Md} \\
\mathbf{x}_{\text{avg},k}^{(t)} &= \begin{bmatrix} \bar{\mathbf{x}}_k^{(t)} \\ \bar{\mathbf{x}}_k^{(t)} \\ \vdots \\ \bar{\mathbf{x}}_k^{(t)} \end{bmatrix} \in \mathbb{R}^{Md}.
\end{aligned}$$

The next network-wide iterate (as a stacked vector) can then be written as $\mathbf{x}_k^{(t)} = (\mathbf{W} \otimes \mathbf{I}) \mathbf{x}_k^{(t-1)} + \alpha \mathcal{H}(\mathbf{x}_k^{(t-1)})$, where \otimes denotes the Kronecker product. The t th iterate can thus be written as

$$\mathbf{x}_k^{(t)} = (\mathbf{W}^t \otimes \mathbf{I}) \mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)}).$$

Since $\mathbf{W} = [w_{ij}]$ is a symmetric and doubly stochastic mixing matrix, its largest eigenvalue is 1 corresponding to the eigenvector $\mathbf{1}_M$, a column vector of all 1's. It is also the left eigenvector of \mathbf{W} . That is, $\mathbf{W} \mathbf{1}_M = \mathbf{1}_M$ and $\mathbf{1}_M^T \mathbf{W} = \mathbf{1}_M^T$. Also, since the squared norm

of Sanger's direction at every node is bounded, it is easy to see $\|\mathcal{H}(\mathbf{x}_k^{(t)})\|^2 = 3M\lambda_1^2(3k-2)(3k+1)$. Now,

$$\begin{aligned}
\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| &\leq \|\mathbf{x}_k^{(t)} - \mathbf{x}_{\text{avg},k}^{(t)}\| = \|\mathbf{x}_k^{(t)} - \frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I}) \mathbf{x}_k^{(t)}\| \\
&= \|(\mathbf{W}^t \otimes \mathbf{I}) \mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)}) \\
&\quad - \frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I})((\mathbf{W}^t \otimes \mathbf{I}) \mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)}))\| \\
&= \|(\mathbf{W}^t \otimes \mathbf{I}) \mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} (\mathbf{W}^{t-1-s} \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)}) \\
&\quad - \frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I}) \mathbf{x}_k^{(0)} - \alpha \sum_{s=0}^{t-1} (\frac{1}{M}((\mathbf{1}_M \mathbf{1}_M^T) \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)}))\| \\
&= \|((\mathbf{W}^t - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I}) \mathbf{x}_k^{(0)} + \alpha \sum_{s=0}^{t-1} ((\mathbf{W}^{t-1-s} \\
&\quad - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&\leq \|((\mathbf{W}^t - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I}) \mathbf{x}_k^{(0)}\| + \alpha \sum_{s=0}^{t-1} \|((\mathbf{W}^{t-1-s} \\
&\quad - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I}) \mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&\leq \|((\mathbf{W}^t - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I})\| \|\mathbf{x}_k^{(0)}\| + \alpha \sum_{s=0}^{t-1} \|((\mathbf{W}^{t-1-s} \\
&\quad - \frac{1}{M}(\mathbf{1}_M \mathbf{1}_M^T)) \otimes \mathbf{I})\| \|\mathcal{H}(\mathbf{x}_k^{(s)})\| \\
&= \beta^t \|\mathbf{x}_k^{(0)}\| + \alpha \sum_{s=0}^{t-1} \beta^{t-1-s} \|\mathcal{H}(\mathbf{x}_k^{(s)})\| \leq \beta^t \sqrt{3M} \\
&\quad + \alpha \sqrt{3M\lambda_1^2(3k-2)(3k+1)} \sum_{s=0}^{t-1} \beta^{t-1-s} \\
&\leq \beta^t \sqrt{3M} + \frac{\alpha \sqrt{3M\lambda_1^2(3k-2)(3k+1)}}{1-\beta} \\
&\leq \sqrt{3M}\lambda_1 \sqrt{(3k-2)(3k+1)} (\beta^t + \frac{\alpha}{1-\beta}) = b_k (\beta^t + \frac{\alpha}{1-\beta}),
\end{aligned}$$

where $b_k = \lambda_1 \sqrt{3M} \sqrt{(3k-2)(3k+1)}$. \square

Appendix F. Statement and proof of Lemma 9

Lemma 9. Suppose $\|\mathbf{x}_{i,k}^{(t)}\|^2 \leq 3$ and $\|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq b_k (\beta^t + \frac{\alpha}{1-\beta})$, then the following is true $\forall k = 1, \dots, K$:

$$\|\mathbf{h}_k^{(t)}\| \leq 3(k+2)\lambda_1 b_k (\beta^t + \frac{\alpha}{1-\beta}). \quad (49)$$

Proof. We have

$$\begin{aligned}
&\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)}) \\
&= \mathbf{C}_i(\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}) - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{x}_{i,k}^{(t)} + (\bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)} \bar{\mathbf{x}}_k^{(t)} \\
&\quad - \sum_{p=1}^{k-1} (\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} - \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i \bar{\mathbf{x}}_k^{(t)}) \\
&= (\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I})(\mathbf{x}_{i,k}^{(t)} \\
&\quad - \bar{\mathbf{x}}_k^{(t)}) - ((\mathbf{x}_{i,k}^{(t)} + \bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})) \bar{\mathbf{x}}_k^{(t)}
\end{aligned}$$

$$\begin{aligned}
& - \sum_{p=1}^{k-1} \mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}) \\
& \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)})\| \\
& \leq \|\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I}\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|(\mathbf{x}_{i,k}^{(t)} \\
& \quad + \bar{\mathbf{x}}_k^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})\| \|\bar{\mathbf{x}}_k^{(t)}\| + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i (\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)})\| \\
& \leq \|\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I}\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|\mathbf{x}_{i,k}^{(t)} \\
& \quad + \bar{\mathbf{x}}_k^{(t)}\| \|\mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \|\bar{\mathbf{x}}_k^{(t)}\| + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T \mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
& \leq \|\mathbf{C}_i - (\mathbf{x}_{i,k}^{(t)})^T \mathbf{C}_i \mathbf{x}_{i,k}^{(t)} \mathbf{I}\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \|\bar{\mathbf{x}}_k^{(t)}\| (\|\mathbf{x}_{i,k}^{(t)}\| \\
& \quad + \|\bar{\mathbf{x}}_k^{(t)}\|) \|\mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \sum_{p=1}^{k-1} \|\mathbf{x}_{i,p}^{(t)} (\mathbf{x}_{i,p}^{(t)})^T\| \|\mathbf{C}_i\| \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
& \leq 3\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| + \sqrt{3}(2\sqrt{3})\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
& \quad + \sum_{p=1}^{k-1} 3\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \\
& = 3(k+2)\lambda_1 \|\mathbf{x}_{i,k}^{(t)} - \bar{\mathbf{x}}_k^{(t)}\| \leq 3(k+2)\lambda_1 b_k (\beta^t + \frac{\alpha}{1-\beta}).
\end{aligned}$$

Thus,

$$\begin{aligned}
\|\mathbf{h}_k^{(t)}\| & \leq \frac{1}{M} \sum_{i=1}^M \|\mathcal{H}_i(\mathbf{x}_{i,k}^{(t)}) - \mathcal{H}_i(\bar{\mathbf{x}}_k^{(t)})\| \\
& \leq 3(k+2)\lambda_1 b_k \left(\beta^t + \frac{\alpha}{1-\beta} \right). \quad (50)
\end{aligned}$$

□

References

- [1] A. Gang, H. Raja, W.U. Bajwa, Fast and communication-efficient distributed PCA, in: Proc. IEEE International Conf. Acoustics, Speech and Signal Process. (ICASSP), 2019, pp. 7450–7454, doi:10.1109/ICASSP.2019.8683095.
- [2] H. Hotelling, Analysis of a complex of statistical variables into principal components, *J. Educ. Psychol.* 24 (6) (1933) 417–441, doi:10.1037/h0071325.
- [3] M. Nokleby, H. Raja, W.U. Bajwa, Scaling-up distributed processing of data streams for machine learning, *Proc. IEEE* 108 (11) (2020) 1984–2012, doi:10.1109/JPROC.2020.3021381.
- [4] W.U. Bajwa, V. Cevher, D. Papailiopoulos, A. Scaglione, Machine learning from distributed, streaming data, *IEEE Signal Process. Mag.* 37 (3) (2020) 11–13, doi:10.1109/MSP.2020.2972654.
- [5] M. Loève, *Probability Theory*, third ed., Springer, New York, 1963.
- [6] R. Dixit, W.U. Bajwa, Exit time analysis for approximations of gradient descent trajectories around saddle points, *ArXiv Prepr. arXiv:2006.01106* (2020).
- [7] R. Dixit, W.U. Bajwa, Boundary conditions for linear exit time gradient trajectories around saddle points: analysis and algorithm, *ArXiv Prepr. arXiv:2101.02625* (2021).
- [8] P. Baldi, K. Hornik, Neural networks and principal component analysis: learning from examples without local minima, *Neural Netw.* 2 (1) (1989) 53–58, doi:10.1016/0893-6080(89)90014-2.
- [9] E. Oja, J. Karhunen, On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix, *J. Math. Anal. Appl.* 106 (1) (1985) 69–84.
- [10] K.I. Diamantaras, S.Y. Kung, *Principal Component Neural Networks: Theory and Applications*, John Wiley & Sons, Inc., USA, 1996.
- [11] T.D. Sanger, Optimal unsupervised learning in a single-layer linear feedforward neural network, *Neural Netw.* 2 (6) (1989) 459–473.
- [12] Z. Yang, A. Gang, W.U. Bajwa, Adversary-resilient distributed and decentralized statistical inference and machine learning: an overview of recent advances under the Byzantine threat model, *IEEE Signal Process. Mag.* 37 (3) (2020) 146–159, doi:10.1109/MSP.2020.2973345.
- [13] K. Pearson, On lines and planes of closest fit to systems of points in space, *Philos. Mag.* 2 (1901) 559–572.
- [14] G.H. Golub, C.F. Van Loan, *Matrix Computations*, third ed., Johns Hopkins University Press, Baltimore, MD, USA, 1996.
- [15] D.O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*, Wiley New York, 1949.
- [16] Z. Yi, M. Ye, J.C. Lv, K.K. Tan, Convergence analysis of a deterministic discrete time system of Oja's PCA learning algorithm, *IEEE Trans. Neural Netw.* 16 (6) (2005) 1318–1328, doi:10.1109/TNN.2005.852236.
- [17] J.C. Lv, Z. Yi, K.K. Tan, Global convergence of GHA learning algorithm with nonzero-approaching adaptive learning rates, *IEEE Trans. Neural Netw.* 18 (6) (2007) 1557–1571, doi:10.1109/TNN.2007.895824.
- [18] D. Kempe, F. McSherry, A decentralized algorithm for spectral analysis, *J. Comput. Syst. Sci.* 74 (1) (2008) 70–83.
- [19] A. Scaglione, R. Pagliari, H. Krim, The decentralized estimation of the sample covariance, in: Proc. 42nd Asilomar Conf. on Signals, Syst. and Comput., 2008, pp. 1722–1726, doi:10.1109/ACSSC.2008.5074720.
- [20] L. Li, A. Scaglione, J.H. Manton, Distributed principal subspace estimation in wireless sensor networks, *IEEE J. Sel. Top. Signal Process.* 5 (4) (2011) 725–738.
- [21] M. Nokleby, W.U. Bajwa, Resource tradeoffs in distributed subspace tracking over the wireless medium, in: Proc. 1st IEEE Global Conf. Signal and Information Processing (GlobalSIP'13), Symposium on Network Theory, Austin, TX, 2013, pp. 823–826, doi:10.1109/globalsip.2013.6737018.
- [22] H. Raja, W.U. Bajwa, Cloud K-SVD: computing data-adaptive representations in the cloud, in: Proc. 51st Annual Allerton Conf. Commun., Control and Computing, 2013, pp. 1474–1481, doi:10.1109/Allerton.2013.6736701.
- [23] H. Wai, A. Scaglione, J. Lafond, E. Moulines, Fast and privacy preserving distributed low-rank regression, in: Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process., (ICASSP), 2017, pp. 4451–4455, doi:10.1109/ICASSP.2017.7952998.
- [24] H. Raja, W.U. Bajwa, Cloud-K-SVD: a collaborative dictionary learning algorithm for big, distributed data, *IEEE Trans. Signal Process.* 64 (1) (2016) 173–188, doi:10.1109/TSP.2015.2472372.
- [25] L. Xiao, S. Boyd, Fast linear iterations for distributed averaging, *Syst. Control Lett.* 53 (1) (2004) 65–78.
- [26] H. Raja, W.U. Bajwa, Distributed stochastic algorithms for high-rate streaming principal component analysis, *CoRR abs/2001.01017* (2020).
- [27] S.X. Wu, H.-T. Wai, L. Li, A. Scaglione, A review of distributed algorithms for principal component analysis, *Proc. IEEE* 106 (8) (2018) 1321–1340.
- [28] M. Hong, D. Hajinezhad, M.-M. Zhao, Prox-PDA: the proximal primal-dual algorithm for fast distributed nonconvex optimization and learning over networks, in: Proc. 34th Int. Conf. Mach. Learning, vol. 70, PMLR, 2017, pp. 1529–1538.
- [29] P. Bianchi, J. Jakubowicz, Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization, *IEEE Trans. Autom. Control* 58 (2) (2013) 391–405, doi:10.1109/TAC.2012.2209984.
- [30] H. Wai, A. Scaglione, J. Lafond, E. Moulines, A projection-free decentralized algorithm for non-convex optimization, in: Proc. 2016 IEEE Global Conf. Signal and Inform. Process. (GlobalSIP), 2016, pp. 475–479, doi:10.1109/GlobalSIP.2016.7905887.
- [31] P.D. Lorenzo, G. Scutari, NEXT: in-network nonconvex optimization, *IEEE Trans. Signal Inf. Process. Netw.* 2 (2) (2016) 120–136, doi:10.1109/TSIPN.2016.2524588.
- [32] A. Nedic, A. Ozdaglar, Distributed subgradient methods for multi-agent optimization, *IEEE Trans. Autom. Control* 54 (1) (2009) 48–61.
- [33] R. Vershynin, How close is the sample covariance matrix to the actual covariance matrix? *J. Theor. Probab.* 25 (2010), doi:10.1007/s10959-010-0338-z.
- [34] R. Arora, A. Cotter, N. Srebro, Stochastic optimization of PCA with capped MSG, in: Proc. Advances Neural Inform. Process. Syst., 2013, pp. 1815–1823.
- [35] M.K. Warmuth, D. Kuzmin, Randomized PCA algorithms with regret bounds that are logarithmic in the dimension, in: Proc. Advances Neural Inform. Process. Syst., 2007, pp. 1481–1488.
- [36] W. Shi, Q. Ling, G. Wu, W. Yin, EXTRA: an exact first-order algorithm for decentralized consensus optimization, *SIAM J. Optim.* 25 (2) (2015) 944–966, doi:10.1137/14096668X.
- [37] F.S. Cattivelli, A.H. Sayed, Diffusion LMS strategies for distributed estimation, *IEEE Trans. Signal Process.* 58 (3) (2010) 1035–1048.
- [38] S. Kar, J.M. Moura, Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems, *IEEE Signal Process. Mag.* 30 (3) (2013) 99–109.
- [39] K. Yuan, Q. Ling, W. Yin, On the convergence of decentralized gradient descent, *SIAM J. Optim.* 26 (3) (2016) 1835–1854, doi:10.1137/130943170.
- [40] S. Boyd, P. Diaconis, L. Xiao, Fastest mixing Markov chain on a graph, *SIAM Rev.* 46 (2003) 667–689.
- [41] P.-A. Absil, R. Mahony, R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, USA, 2007.
- [42] Y. LeCun, C. Cortes, C. Burges, *MNIST Handwritten Digit Database*, 2, ATT Labs, 2010.
- [43] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, Technical Report, 2009.
- [44] G. Qu, N. Li, Harnessing smoothness to accelerate distributed optimization, *IEEE Trans. Control Netw. Syst.* 5 (3) (2018) 1245–1260, doi:10.1109/TCNS.2017.2698261.