Linear Algebra and its Applications 611 (2021) 334–355

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Accelerating the distributed Kaczmarz algorithm by strong over-relaxation

Riley Borgard[d], Steven N. Harding[c], Haley Duba[e], Chloe Makdad[b], Jay Mayfield[c], Randal Tuggle[a], Eric S. Weber[c,*]

[a] Berry College, United States of America
[b] Butler University, United States of America
[c] Iowa State University, United States of America
[d] Purdue University, United States of America
[e] Wheaton College, United States of America

ARTICLE INFO

Article history:
Received 30 January 2020
Accepted 26 October 2020
Available online 31 October 2020
Submitted by A. Frommer

MSC:
15A06
15A24

Keywords:
Kaczmarz algorithm

ABSTRACT

The distributed Kaczmarz algorithm is an adaptation of the standard Kaczmarz algorithm to the situation in which data is distributed throughout a network represented by a tree. We isolate substructures of the network and study convergence of the distributed Kaczmarz algorithm for relatively large relaxation parameters associated to these substructures. If the system is consistent, then the algorithm converges to the solution of minimal norm; however, if the system is inconsistent, then the algorithm converges to an approximated least-squares solution that is dependent on the parameters and the network topology. We show that the relaxation parameters may be larger than the standard upper-bound in literature in this context and provide numerical experiments to support our results.

* Corresponding author.
E-mail address: esweber@iastate.edu (E.S. Weber).

https://doi.org/10.1016/j.laa.2020.10.035

1. Introduction

The Kaczmarz algorithm, introduced in [9], is a classic row-action projection method for solving a system of linear equations $A\vec{x} = \vec{b}$ where A is a complex-valued $k \times d$ matrix. We denote row i of the matrix A by $\vec{a}_i^*$ so that the corresponding equation in the system is $\langle \vec{x}, \vec{a}_i \rangle = \vec{a}_i^* \vec{x} = b_i$. Herein, we provide a self-contained description of the Kaczmarz algorithm for completeness. Given an initial vector $\vec{x}^{(0)}$, we find the orthogonal projection of $\vec{x}^{(0)}$ onto the hyperplane $\vec{a}_1^* \vec{x} = b_1$ to obtain the estimate $\vec{x}^{(1)}$. We repeat this procedure, iterating through the rows of A; once we obtain $\vec{x}^{(k)}$, we return to the first equation to obtain $\vec{x}^{(k+1)}$ and continue through the matrix as before. More precisely, for $i = n \pmod{k} + 1$, we have

$$\vec{x}^{(n+1)} = \vec{x}^{(n)} + \omega_i \frac{b_i - \vec{a}_i^* \vec{x}^{(n)}}{\|\vec{a}_i\|^2} \vec{a}_i, \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm, and ω_i is a relaxation parameter. Stefan Kaczmarz showed in [9] that if the system is consistent and the solution is unique, then the sequence $\{\vec{x}^{(n)}\}$ converges to the solution with no relaxation parameter. Later, several authors considered a uniform relaxation parameter $\omega_i = \omega$. In [20], Tanabe showed that the sequence $\{\vec{x}^{(n)}\}$ converges to the solution of minimal norm when the system is consistent for any $\omega \in (0, 2)$. When the system is inconsistent, it was shown in [4] (see also [13]) that for every $\omega \in (0, 2)$, the sequence $\{\vec{x}^{(n)}\}$ converges, and for ω small, the limit is an approximation of a weighted least-squares solution.

Since each estimate is obtained by projecting the previous estimate onto the appropriate hyperplane, the Kaczmarz algorithm is well-suited for an adaptation to a network structure where each equation in the system corresponds to a node in a tree, an undirected graph excluding cycles. This was formalized in [6]. Such a system is said to be distributed, as any node is uninformed of the equation of another node. A distributed system has many benefits in practical applications, e.g. data that is too large to store on a single server or cannot be explicitly shared for privacy reasons. Further, for large distributed systems, we can exploit parallelism to speed up the real time of iterations within the algorithm.

1.1. Related work

The Kaczmarz method was originally introduced in [9]. Variations on the Kaczmarz method allowed for relaxation parameters [20], re-ordering equations to speed up convergence [5], or considering block versions of the Kaczmarz method with relaxation matrices Ω_i ([4], see also [3]). Block versions of the method allow for over-relaxation parameters of greater than 2 as demonstrated in [1,14], in similar fashion to our results in Section 5. The Kaczmarz method is also known for memory efficiency [7].

Relatively recently, choosing the next equation randomly has been shown to dramatically improve the rate of convergence of the algorithm [19,23,16,17,2]. Moreover, this randomized version of the Kaczmarz algorithm has been shown to be comparable to the gradient descent method [15]. In our situation, the equations are *a priori* distributed across a network with a fixed topology; this determines the next equation to use to update the estimate and does not allow a choice. Instead, we demonstrate that the convergence rate can be improved by relaxation parameters greater than 2 in Section 6.

A distributed version of the Kaczmarz algorithm was introduced in [10]. The main ideas presented there are very similar to ours: updated estimates are obtained from prior estimates using the Kaczmarz update with the equations that are available at the node, and distributed estimates are averaged together at a single node (which the authors refer to as a fusion center, for us it is the root of the tree). Another distributed version was proposed in [11], which has a shared memory architecture.

Our distributed Kaczmarz algorithm for solving systems of linear equations is similar to gossip or consensus algorithms [18,12] and distributed optimization [21,8,22]. See [6] for a fuller discussion of these similarities.

1.2. Main results

Our main focus in the present paper is to consider an extension of the Kaczmarz algorithm that can solve a system of linear equations when the equations are distributed across a network. This extension was introduced in [6], where it was shown that the distributed form of the Kaczmarz algorithm converges for any uniform relaxation parameter $\omega \in (0, 2)$. It was also shown that, as is the case with the classical Kaczmarz algorithm, the convergence rate can be accelerated by choosing $\omega > 1$. Moreover, it was observed that convergence can occur with $\omega > 2$, which cannot happen in the classical case.

Our main results concern proving convergence for relaxation parameters that are equation dependent, as well as determining what the algorithm converges to. Additionally, we demonstrate that some relaxation parameters can exceed the upper bound of 2 that exists for uniform relaxation parameters, and doing so accelerates the convergence of the distributed algorithm. First, we prove that with large relaxation parameters that satisfy a certain admissibility condition (Definition 1), when the system is consistent, the distributed Kaczmarz algorithm converges to the solution of minimal norm independent of the relaxation parameters (Theorem 3.8). Second, we prove that under the same admissibility conditions, when the system is inconsistent, the distributed Kaczmarz algorithm will yield approximations of a weighted least-squares solution as the parameters tend to 0 (Theorem 4.4).

We then consider possible values for the relaxation parameters that satisfy the admissibility condition. We prove an estimate on the sizes of the relaxation parameters at nodes that are near the leaves of the tree (Corollary 5.1.1). Our estimate allows for relaxation parameters that are larger than 2. In Section 6, we present numerical examples that illustrate convergence with relaxation parameters greater than 2 that is faster than with parameters less than 2.

1.3. Notation

We define the network for a distributed system as a tree in graph theory parlance–that is, a connected graph consisting of k vertices, each corresponding to one equation in the system, with edges that connect particular pairs of vertices in such a way that there are no cycles. Herein, we only consider trees which are rooted, having a single vertex r designated as the root. We denote arbitrary vertices of the tree by either u or v. We write $u \preceq v$ when either $u = v$ or u is on a path from r to v. We further write $u \rightarrow v$ or $v \leftarrow u$ when $u \neq v$ and $u \preceq x \preceq v$ implies either $u = x$ or $x = v$. From this partial ordering on the set of vertices, we define a leaf of the tree as a vertex ℓ satisfying $\ell \preceq u$ implies $u = \ell$, and we denote the collection of all of the leaves by $\mathcal{L}$. Whenever necessary, we enumerate the leaves as $\ell_1, \ell_2, ..., \ell_t$.

A weight w is a positive function on the paths of the tree, which we denote by $w(u, v)$ where $u \preceq v$, that satisfies the following three conditions:

(1) For every vertex $u \notin \mathcal{L}$,

$$\sum_{v\,:\,u\rightarrow v} w(u, v) = 1 \tag{2}$$

(2) if $u = u_1 \rightarrow u_2 \rightarrow \cdots \rightarrow u_J = v$, then

$$w(u, v) = \prod_{j=1}^{J-1} w(u_j, u_{j+1}), \tag{3}$$

(3) $w(u, u) = 1$.

When working with a distributed network represented by a rooted tree, it is convenient to index each equation by the corresponding vertex, and we proceed with this convention throughout the remainder of the paper. We recall, for a linear transformation T on $\mathcal{H}$, the kernel (null space) $\mathcal{N}(T) = \{\vec{x} \in \mathcal{H} \,:\, T\vec{x} = \vec{0}\}$ and the range $\mathcal{R}(T) = \{T\vec{x} \,:\, \vec{x} \in \mathcal{H}\}$. We define $S_v\vec{x} := \vec{a}_v^*\vec{x}$, and let P_v be the orthogonal projection onto $\mathcal{N}(S_v)$,

$$P_v\vec{x} = (I - S_v^*(S_vS_v^*)^{-1}S_v)\vec{x} = \vec{x} - \frac{\vec{a}_v^*\vec{x}}{\|\vec{a}_v\|^2}\vec{a}_v. \tag{4}$$

Then, let Q_v be the affine projection onto the hyperplane $S_v\vec{x} = b_v$,

$$Q_v\vec{x} = \vec{x} + \frac{b_v - \vec{a}_v^*\vec{x}}{\|\vec{a}_v\|^2}\vec{a}_v. \tag{5}$$

The relationship between P_v and Q_v is then

$$Q_v\vec{x} = P_v\vec{x} + \vec{h}_v, \tag{6}$$

where $\vec{h}_v$ is the vector that satisfies $S_v\vec{h}_v = b_v$ and is orthogonal to $\mathcal{N}(S_v)$. The vector $\vec{\omega}$ refers to the entire collection of relaxation parameters, and notation associated with $\vec{\omega}$ implies a dependence on the relaxation parameters. Specifically, the component ω_v in $\vec{\omega}$ is the relaxation parameter associated with vertex v. We further define the associated operators $P_v^{\vec{\omega}}$ and $Q_v^{\vec{\omega}}$ by

$$P_v^{\vec{\omega}} = (1-\omega_v)I + \omega_v P_v, \tag{7}$$

$$Q_v^{\vec{\omega}} = (1-\omega_v)I + \omega_v Q_v. \tag{8}$$

The relationship between $P_v^{\vec{\omega}}$ and $Q_v^{\vec{\omega}}$ is then

$$Q_v^{\vec{\omega}}\vec{x} = P_v^{\vec{\omega}}\vec{x} + \omega_v\vec{h}_v. \tag{9}$$

Lemma 1.1. *Let $\omega_v \in (0,2)$. Then $P_v^{\vec{\omega}}$ is a contraction (i.e., $\|P_v^{\vec{\omega}}\| \leq 1$). Moreover, $\|P_v^{\vec{\omega}}\vec{x}\| \leq \|\vec{x}\|$ with equality if and only if $\vec{x} \in \mathcal{N}(S_v)$.*

Proof. The argument is fairly straightforward, yet it illustrates the sufficient condition that $\omega_v \in (0,2)$.

$$\begin{aligned}\|P_v^{\vec{\omega}}\vec{x}\|^2 &= \|P_v(\vec{x}) + (1-\omega_v)(I-P_v)(\vec{x})\|^2 \\ &= \|P_v\vec{x}\|^2 + |1-\omega_v|^2\|(I-P_v)(\vec{x})\|^2 \\ &\leq \|P_v\vec{x}\|^2 + \|(I-P_v)(\vec{x})\|^2 = \|\vec{x}\|^2\end{aligned}$$

with equality if and only if $\vec{x} = P_v\vec{x}$. □

2. The distributed Kaczmarz algorithm with relaxation

Each iteration of the distributed Kaczmarz algorithm begins with an estimate $\vec{x}^{(n)}$ at the root of the tree; the superscript indicates the number of times that we iterated through the tree to obtain the estimate for some given initial estimate $\vec{x}^{(0)}$. An iteration of the algorithm occurs in two stages: *dispersion* followed by *pooling*. In the dispersion stage, a new estimate is first calculated at the root using the Kaczmarz update with the relaxation parameter ω_r,

$$\vec{x}_r^{(n)} = \vec{x}^{(n)} + \omega_r\frac{b_r - \vec{a}_r^*\vec{x}^{(n)}}{\|\vec{a}_r\|^2}\vec{a}_r =: Q_r^{\vec{\omega}}\vec{x}^{(n)}.$$

Each subsequent vertex $v \neq r$ receives an input estimate $\vec{x}_u^{(n)}$ from its parent u (i.e., $u \to v$), and a new estimate is calculated at the vertex v using the Kaczmarz update with relaxation parameter ω_v,

$$\vec{x}_v^{(n)} = \vec{x}_u^{(n)} + \omega_v\frac{b_v - \vec{a}_v^*\vec{x}_u^{(n)}}{\|\vec{a}_v\|^2}\vec{a}_v =: Q_v^{\vec{\omega}}\vec{x}_u^{(n)}.$$

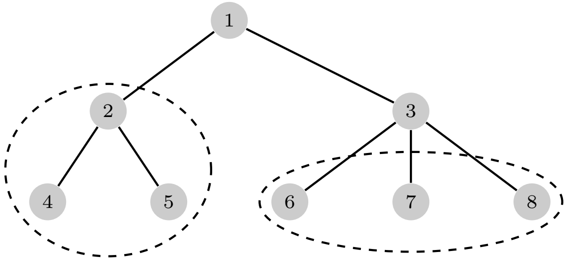


Fig. 1. A network with the two types of subnetworks.

Each leaf ℓ then has its own estimate $\vec{x}_\ell^{(n)}$ at the end of the dispersion stage.

In the pooling stage, we back-propagate the leaf estimates, weighting along the edges, to obtain the next iterate in the algorithm,

$$\vec{x}^{(n+1)} = \sum_{\ell \in \mathcal{L}} w(r,\ell)\vec{x}_\ell^{(n)}.$$

It was shown in [6] that the distributed Kaczmarz algorithm with uniform relaxation parameter $\omega_v = \omega \in (0,2)$ converges to the solution of minimal norm when the system is consistent and converges to an approximate solution related to some weighted least-squares solution, dependent on the parameters and the network topology, when the system is inconsistent.

2.1. Substructures of a network

A *subnetwork* G of a network is a subset of vertices and edges satisfying the following conditions:

(1) If $u \in G$ and $u \to v$, then G contains v and the edge between u and v.
(2) If $u, v \in G$, $x \to u$ and $y \to v$, and $x, y \notin G$, then $x = y$.
(3) Let $u, v \in G$. The path from u to v does not include the root.

The topology of a subnetwork can thus be characterized as follows: It is either a network itself or a *leaf subnetwork* (a set containing only leaves). Fig. 1 illustrates a network with both types of subnetworks.

Throughout the paper, we assume that every leaf is included in a subnetwork. The purpose of each subnetwork is to isolate a substructure of the network, so we assume that the subnetworks are pairwise disjoint. Moreover, we will show in Section 5 that the subnetworks can have relaxation parameters that exceed the classical upper bound of 2, and that doing so accelerates the convergence of the algorithm.

We denote the subnetworks by G_1, G_2, ..., G_c and denote the vertex that immediately precedes G_i by g_i. We further denote the leaves in G_i by $\ell_{i,1}$, $\ell_{i,2}$, ..., ℓ_{i,t_i}. We last denote the root of the largest tree in G_i with the leaf $\ell_{i,j}$ by $r_{i,j}$. For example, in Fig. 1, we have the following:

- $G_1 = \{2, 4, 5\}$, $G_2 = \{6, 7, 8\}$
- $g_1 = 1$, $g_2 = 3$
- $\ell_{1,1} = 4$, $\ell_{1,2} = 5$, $\ell_{2,1} = 6$, $\ell_{2,2} = 7$, $\ell_{2,3} = 8$
- $r_{1,1} = 2$, $r_{1,2} = 2$, $r_{2,1} = 6$, $r_{2,2} = 7$, $r_{2,3} = 8$

As each subnetwork is a forest of trees, we may interpret an iteration of $\vec{x}^{(n)}_{g_i}$ through the subnetwork G_i as a weighted average of the iterations through the corresponding trees. We therefore define the following operators:

$$P^{\vec{\omega}}_{G_i} = \sum_{j=1}^{t_i} w(g_i, \ell_{i,j}) P^{\vec{\omega}}_{\ell_{i,j}} \dots P^{\vec{\omega}}_{r_{i,j}}, \tag{10}$$

$$P^{\vec{\omega}}_{G_i,r} = P^{\vec{\omega}}_{G_i} P^{\vec{\omega}}_{g_i} \dots P^{\vec{\omega}}_{r}, \tag{11}$$

$$P^{\vec{\omega}} = \sum_{i=1}^{c} w(r, g_i) P^{\vec{\omega}}_{G_i,r}, \tag{12}$$

where $P^{\vec{\omega}}_{v} \dots P^{\vec{\omega}}_{u}$ with $u \preceq v$ is the composition of those operators $P^{\vec{\omega}}_{x}$ where $u \preceq x \preceq v$ in the appropriate order designated by the path from u to v. We define analogous operators in Q. Doing so yields the full distributed Kaczmarz iteration as:

$$\vec{x}^{n+1} = Q^{\vec{\omega}} \vec{x}^{n}.$$

We will show in Section 5 that the substructures in a network generally admit large relaxation parameters for convergence. We require that the relaxation parameters satisfy certain admissibility conditions.

Definition 1. We say that the relaxation parameters ω_v are *admissible* provided that:

(1) If $v \notin G_i$ for every i, then $\omega_v \in (0, 2)$.
(2) For each $i \in \{1, 2, ..., c\}$, there exists a constant $\alpha_i < 1$ such that

$$\|P^{\vec{\omega}}_{G_i} \vec{x}\| \leq \alpha_i \|\vec{x}\|$$

for all $\vec{x} \in \text{span}\{\vec{a}_u : u \in G_i\}$.

Lemma 2.1. *If the relaxation parameters are admissible, then $P^{\vec{\omega}}_{G_i}$, $P^{\vec{\omega}}_{G_i,r}$ and $P^{\vec{\omega}}$ are contractions.*

Proof. Suppose that $\vec{x} \in \{\vec{a}_u : u \in G_i\}^{\perp}$, the subspace orthogonal to the vectors in the set $\{\vec{a}_u : u \in G_i\}$. Then $P^{\vec{\omega}}_{u} \vec{x} = \vec{x}$ for every $u \in G_i$, and we have

$$P_{G_i}^{\vec{\omega}}\vec{x} = \sum_{j=1}^{t_i} w(g_i, \ell_{i,j}) P_{\ell_{i,j}}^{\vec{\omega}} \cdots P_{r_{i,j}}^{\vec{\omega}} \vec{x} = \vec{x}.$$

Since $\operatorname{span}\{\vec{a}_u \ : \ u \in G_i\}$ is an invariant subspace for $P_{G_i}^{\vec{\omega}}$, the operator $P_{G_i}^{\vec{\omega}}$ is a contraction. Then, from Lemma 1.1, it follows that $P_{G_i,r}^{\vec{\omega}}$ and, subsequently, $P^{\vec{\omega}}$ are contractions. □

3. Consistent systems

We prove Theorem 3.8, the main result of this section, using a sequence of lemmas. We follow the argument presented in [6], adapting those lemmas for our assumptions on the relaxation parameters. We also direct the reader to the original source [13].

Lemma 3.1. *Let $\mathcal{H}$ be a Hilbert space and $\mathcal{K}$ be a closed subspace of $\mathcal{H}$. Let U be a linear operator on $\mathcal{H}$ with the following properties:*

(1) $U\vec{x} = \vec{x}$ for every $\vec{x} \in \mathcal{K}$,
(2) $\mathcal{K}^\perp$ is an invariant subspace for U (i.e., $U(\mathcal{K}^\perp) \subseteq \mathcal{K}^\perp$),
(3) $\|U|_{\mathcal{K}^\perp}\| < 1$.

Given a sequence $\{\vec{x}_k\}$ in $\mathcal{H}$ such that

$$\|\vec{x}_k\| \leq 1 \text{ and } \lim_{k\to\infty} \|U\vec{x}_k\| = 1,$$

it follows that

$$\lim_{k\to\infty} (I - U)\vec{x}_k = \vec{0}.$$

Proof. For convenience, we denote $\alpha = \|U|_{\mathcal{K}^\perp}\|$, and let P be the orthogonal projection onto $\mathcal{K}^\perp$. We claim that $\|P\vec{x}_k\| \to 0$. Indeed, we have

$$\begin{aligned}
1 &= \lim_{k\to\infty} \|U\vec{x}_k\|^2 \\
&= \lim_{k\to\infty} \|U(I-P)\vec{x}_k + UP\vec{x}_k\|^2 \\
&= \lim_{k\to\infty} \left(\|(I-P)\vec{x}_k\|^2 + \|UP\vec{x}_k\|^2\right) \\
&\leq \liminf \left(\|(I-P)\vec{x}_k\|^2 + \alpha^2\|P\vec{x}_k\|^2\right) \\
&= \liminf \left(\|\vec{x}_k\|^2 - (1-\alpha^2)\|P\vec{x}_k\|^2\right) \\
&\leq \liminf \left(1 - (1-\alpha^2)\|P\vec{x}_k\|^2\right) \\
&= 1 - (1-\alpha^2) \limsup \|P\vec{x}_k\|^2 \leq 1.
\end{aligned}$$

We therefore observe that $1-(1-\alpha^2)\limsup\|P\vec{x}_k\|^2=1$ so that $\limsup\|P\vec{x}_k\|=0$, as desired. Hence

$$\lim_{k\to\infty}(I-U)\vec{x}_k=\lim_{k\to\infty}(I-U)(P\vec{x}_k)=\vec{0}. \quad \square$$

Lemma 3.2. *Suppose $i\in\{1,2,...,c\}$, an enumeration of the subnetworks, and that the relaxation parameters are admissible. Suppose that $\{\vec{x}_k\}$ is a sequence in $\mathbb{C}^d$ such that*

$$\|\vec{x}_k\|\le 1 \textit{ and } \lim_{k\to\infty}\|P_{G_i}^{\vec{\omega}}\vec{x}_k\|=1.$$

It follows that

$$\lim_{k\to\infty}(I-P_{G_i}^{\vec{\omega}})\vec{x}_k=\vec{0}.$$

Proof. Let $\mathcal{K}=\{\vec{a}_u\ :\ u\in G_i\}^\perp$. The proof consists of simply verifying that $P_{G_i}^{\vec{\omega}}$ satisfies the conditions of Lemma 3.1.

As observed in Lemma 2.1, we have that $P_{G_i}^{\vec{\omega}}\vec{x}=\vec{x}$ for every $\vec{x}\in\mathcal{K}$ and that $\mathcal{K}^\perp$ is an invariant subspace for $P_{G_i}^{\vec{\omega}}$. Condition (3) of Lemma 3.1 follows from the assumptions on the relaxation parameters, specifically $\|P_{G_i}^{\vec{\omega}}\vec{x}\|\le\alpha_i\|\vec{x}\|$ for every $\vec{x}\in\mathcal{K}^\perp$. □

Lemma 3.3. *Suppose $i\in\{1,2,...,c\}$, an enumeration of the subnetworks, and that the relaxation parameters are admissible. Suppose that $\{\vec{x}_k\}$ is a sequence in $\mathbb{C}^d$ such that*

$$\|\vec{x}_k\|\le 1 \textit{ and } \lim_{k\to\infty}\|P_{G_i,r}^{\vec{\omega}}\vec{x}_k\|=1.$$

It follows that

$$\lim_{k\to\infty}(I-P_{G_i,r}^{\vec{\omega}})\vec{x}_k=\vec{0}.$$

Proof. Note that

$$(I-P_{G_i}^{\vec{\omega}}P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}})\vec{x}_k=(I-P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}})\vec{x}_k+(I-P_{G_i}^{\vec{\omega}})P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}}\vec{x}_k.$$

Since $\|P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}}\vec{x}_k\|\le 1$, we have $(I-P_{G_i}^{\vec{\omega}})P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}}\vec{x}_k\to\vec{0}$ from Lemma 3.2. Hence it suffices to show $(I-P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}})\vec{x}_k\to\vec{0}$. Consider the path from r to g_i, say $r=u_1\to u_2\to...\to u_n=g_i$, and let $\mathcal{K}=\{\vec{a}_{u_j}\ :\ 1\le j\le n\}^\perp$. We check Lemma 3.1. Conditions (1) and (2) are straightforward to check, so we only show condition (3). Assume by way of contradiction that $\|P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}}|_{\mathcal{K}^\perp}\|=1$. By continuity and compactness, there then exists a unit vector $\vec{x}\in\mathcal{K}^\perp$ such that $\|P_{g_i}^{\vec{\omega}}\cdots P_r^{\vec{\omega}}\vec{x}\|=1$. From this observation and Lemma 1.1, it follows that $\vec{x}\in\mathcal{K}$ so that $\vec{x}=\vec{0}$, which is a contradiction. □

Lemma 3.4. *Suppose that $\{\vec{x}_k\}$ is a sequence in $\mathbb{C}^d$ such that*

$$\|\vec{x}_k\| \leq 1 \text{ and } \lim_{k\to\infty} \|P^{\vec{\omega}}\vec{x}_k\| = 1,$$

and that the relaxation parameters are admissible. Then

$$\lim_{k\to\infty}(I - P^{\vec{\omega}})\vec{x}_k = \vec{0}.$$

Proof. Recalling Equation (12), we note that

$$(I - P^{\vec{\omega}})\vec{x}_k = \sum_{i=1}^{c} w(r, g_i)(I - P^{\vec{\omega}}_{G_i,r})\vec{x}_k.$$

Therefore it suffices to show that the hypotheses of Lemma 3.3 are satisfied. From Lemma 2.1, we have $\|P^{\vec{\omega}}_{G_i,r}\vec{x}_k\| \leq 1$ and, thus,

$$1 = \lim_{k\to\infty}\|P^{\vec{\omega}}\vec{x}_k\| \leq \liminf \sum_{i=1}^{c} w(r, g_i)\|P^{\vec{\omega}}_{G_i,r}\vec{x}_k\| \leq 1.$$

It follows that, for each $i \in \{1, 2, ..., c\}$,

$$\lim_{k\to\infty}\|P^{\vec{\omega}}_{G_i,r}\vec{x}_k\| = 1. \quad \square$$

Proposition 3.5. *If the relaxation parameters are admissible and $\|P^{\vec{\omega}}\vec{x}\| = \|\vec{x}\|$, then $\vec{x} \in \mathcal{R}(A^*)^{\perp}$.*

Proof. Note that

$$\|\vec{x}\| = \left\|\sum_i w(r, g_i)P^{\vec{\omega}}_{G_i}P^{\vec{\omega}}_{g_i}...P^{\vec{\omega}}_{r}\vec{x}\right\| \leq \sum_i w(r, g_i)\|P^{\vec{\omega}}_{G_i}P^{\vec{\omega}}_{g_i}...P^{\vec{\omega}}_{r}\vec{x}\| \leq \|\vec{x}\|.$$

Therefore it follows that $\|P^{\vec{\omega}}_{G_i}P^{\vec{\omega}}_{g_i}...P^{\vec{\omega}}_{r}\vec{x}\| = \|\vec{x}\|$ for all i. Hence $\|P^{\vec{\omega}}_{r}\vec{x}\| = \|\vec{x}\|$ which, by Lemma 1.1, implies that $\vec{x} \in \mathcal{N}(S_r)$ and $P^{\vec{\omega}}_{r}\vec{x} = \vec{x}$. We then inductively find

$$\vec{x} \in \mathcal{N}(S_{g_i}) \cap ... \cap \mathcal{N}(S_r), \tag{13}$$

$P^{\vec{\omega}}_{g_i}\vec{x} = ... = P^{\vec{\omega}}_{r}\vec{x} = \vec{x}$, and $\|P^{\vec{\omega}}_{G_i}\vec{x}\| = \|\vec{x}\|$. Now let P be the orthogonal projection onto $\{\vec{a}_u : u \in G_i\}^{\perp}$. Then, as argued in Lemma 2.1, we find

$$\begin{aligned}\|\vec{x}\|^2 &= \|P^{\vec{\omega}}_{G_i}\vec{x}\|^2 \\ &= \|P^{\vec{\omega}}_{G_i}P\vec{x} + P^{\vec{\omega}}_{G_i}(I - P)\vec{x}\|^2 \\ &= \|P\vec{x} + P^{\vec{\omega}}_{G_i}(I - P)\vec{x}\|^2\end{aligned}$$

$$
\begin{aligned}
&= \|P\vec{x}\|^2 + \|P^{\vec{\omega}}_{G_i}(I-P)\vec{x}\|^2 \\
&\le \|P\vec{x}\|^2 + \alpha_i^2\|(I-P)\vec{x}\|^2 \\
&\le \|\vec{x}\|^2.
\end{aligned}
$$

Therefore $P\vec{x} = \vec{x}$ so that $\vec{x} \in \mathcal{N}(S_u)$ for every $u \in G_i$. Since every vertex is either in a subnetwork G_i or is a predecessor of some G_i, combining Equation (13) with the prior conclusion completes the proof. □

The next lemma is an immediate consequence of Proposition 3.5.

Lemma 3.6. *Let $\mathcal{V}$ be the collection of all of the vertices in the network. Then*

$$\mathcal{N}(I - P^{\vec{\omega}}) = \bigcap_{v\in\mathcal{V}} \mathcal{N}(I - P_v).$$

Lemma 3.7. *Suppose the relaxation parameters are admissible. Let $\mathcal{V}$ be the collection of all of the vertices in the network. As $k \to \infty$, $(P^{\vec{\omega}})^k$ converges strongly to the orthogonal projection onto*

$$\bigcap_{v\in\mathcal{V}} \mathcal{N}(I - P_v) = \mathcal{N}(A).$$

Proof. Using Lemmas 3.4 and 3.6 with the observation that $\mathcal{N}(S_v) = \mathcal{N}(I - P_v)$, the proof is identical to the proof of Lemma 3.5 in [13]. □

Theorem 3.8. *If the system of equations $A\vec{x} = \vec{b}$ is consistent, then the sequence of estimates $\{\vec{x}^{(n)}\}$ from the distributed Kaczmarz algorithm given by the recursion*

$$\vec{x}^{(n+1)} = Q^{\vec{\omega}}\vec{x}^{(n)} = \sum_{\ell\in\mathcal{L}} w(r,\ell) Q^{\vec{\omega}}_{\ell} \cdots Q^{\vec{\omega}}_{r} \vec{x}^{(n)},$$

with admissible relaxation parameters, converges to the solution of minimal norm provided that the initial estimate $\vec{x}^{(0)} \in \mathcal{R}(A^*)$.

Proof. Let $\vec{x}$ be a solution to the system of equations, and let v be any vertex in the network. Then, from Equation (9), we have

$$\vec{x} = Q^{\vec{\omega}}_v \vec{x} = P^{\vec{\omega}}_v \vec{x} + \omega_v \vec{h}_v.$$

Let $\vec{y}$ be an arbitrary vector. From Equation (9), again, we find

$$Q^{\vec{\omega}}_v \vec{y} = P^{\vec{\omega}}_v \vec{y} + \omega_v \vec{h}_v = P^{\vec{\omega}}_v (\vec{y} - \vec{x}) + \vec{x}.$$

It then immediately follows from this last identity that

$$\begin{aligned}
Q^{\vec{\omega}}\vec{y} &= \sum_{\ell\in\mathcal{L}} w(r,\ell)Q_\ell^{\vec{\omega}}\cdots Q_r^{\vec{\omega}}\vec{y} \\
&= \left(\sum_{\ell\in\mathcal{L}} w(r,\ell)P_\ell^{\vec{\omega}}\cdots P_r^{\vec{\omega}}(\vec{y}-\vec{x})\right)+\vec{x} \\
&= P^{\vec{\omega}}(\vec{y}-\vec{x})+\vec{x}.
\end{aligned}$$

Further, for every positive integer k,

$$(Q^{\vec{\omega}})^k\vec{y} = (P^{\vec{\omega}})^k(\vec{y}-\vec{x})+\vec{x}.$$

Then, from Lemma 3.7, we have that

$$(Q^{\vec{\omega}})^k\vec{y} \to T(\vec{y}-\vec{x})+\vec{x}$$

where T is the orthogonal projection onto $\mathcal{N}(A)$. Now, if $\vec{y}=\vec{x}^{(0)}\in\mathcal{R}(A^*)$, then $T(\vec{y}-\vec{x})+\vec{x}=(I-T)\vec{x}$ is the solution of minimal norm, which concludes the proof. □

4. Inconsistent systems

In this section, we show that the distributed Kaczmarz algorithm with admissible relaxation parameters converges regardless of the consistency of the system and that the limit point is an approximation of a weighted least-squares solution when the system of equations is inconsistent. We first develop the relevant theory by following Successive Over-Relaxation (SOR) analysis of the Kaczmarz algorithm as developed in [13].

Let $\ell\in\mathcal{L}$, and suppose $r=u_1\to u_2\to\cdots\to u_{p-1}\to u_p=\ell$, the path from r to ℓ. We denote the initial estimate at r by $\vec{x}_{u_0}$. Then, from the Kaczmarz update, we recursively attain $\vec{x}_{u_j}$, the relaxed projection of $\vec{x}_{u_{j-1}}$ onto the hyperplane given by $\vec{a}_{u_j}^*\vec{x}=b_{u_j}$,

$$\vec{x}_{u_j} = Q_{u_j}^{\vec{\omega}}\vec{x}_{u_{j-1}} = \vec{x}_{u_{j-1}} + \omega_{u_j}\frac{b_{u_j}-\vec{a}_{u_j}^*\vec{x}_{u_{j-1}}}{\|\vec{a}_{u_j}\|^2}\vec{a}_{u_j}. \tag{14}$$

Hence, there exist complex scalars $\{c_k\}_{k=1}^p$ such that, for all j,

$$\vec{x}_{u_j} = \vec{x}_{u_0} + \sum_{k=1}^{j} c_k\vec{a}_{u_k}. \tag{15}$$

Substituting Equation (15) into Equation (14),

$$c_j = \omega_{u_j}\frac{b_{u_j}-\vec{a}_{u_j}^*\vec{x}_{u_0}-\sum\limits_{k=1}^{j-1}c_k\vec{a}_{u_j}^*\vec{a}_{u_k}}{\|\vec{a}_{u_j}\|^2}. \tag{16}$$

We can then consolidate Equation (16) for all j into the matrix equation

$$D_\ell \vec{c} = \Omega_\ell(\vec{b}_\ell - A_\ell \vec{x}_{u_0} - L_\ell \vec{c}) \tag{17}$$

where $\vec{c} = (c_1, c_2, ..., c_p)^T$ and D_ℓ, Ω_ℓ, $\vec{b}_\ell$, L_ℓ and A_ℓ are as follows:

$$D_\ell = \begin{pmatrix} \|\vec{a}_{u_1}\|^2 & 0 & \dots & 0 \\ 0 & \|\vec{a}_{u_2}\|^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \|\vec{a}_{u_p}\|^2 \end{pmatrix}, \Omega_\ell = \begin{pmatrix} \omega_{u_1} & 0 & \dots & 0 \\ 0 & \omega_{u_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega_{u_p} \end{pmatrix},$$

$$\vec{b}_\ell = \begin{pmatrix} b_{u_1} \\ b_{u_2} \\ \vdots \\ b_{u_p} \end{pmatrix}, L_\ell = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ \vec{a}^*_{u_2}\vec{a}_{u_1} & 0 & 0 & \dots & 0 & 0 \\ \vec{a}^*_{u_3}\vec{a}_{u_1} & \vec{a}^*_{u_3}\vec{a}_{u_2} & 0 & \dots & 0 & 0 \\ \vec{a}^*_{u_4}\vec{a}_{u_1} & \vec{a}^*_{u_4}\vec{a}_{u_2} & \vec{a}^*_{u_4}\vec{a}_{u_3} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vec{a}^*_{u_p}\vec{a}_{u_1} & \vec{a}^*_{u_p}\vec{a}_{u_2} & \vec{a}^*_{u_p}\vec{a}_{u_3} & \dots & \vec{a}^*_{u_p}\vec{a}_{u_{p-1}} & 0 \end{pmatrix}, A_\ell = \begin{pmatrix} \vec{a}^*_{u_1} \\ \vec{a}^*_{u_2} \\ \vdots \\ \vec{a}^*_{u_p} \end{pmatrix}.$$

Altogether, from Equations (15) and (17), respectively, we may express the iterate $\vec{x}_\ell^{(n)}$ at the leaf ℓ given the initial vector $\vec{x}^{(n)}$ at the root in terms of the scalar vector $\vec{c}$,

$$\begin{aligned} \vec{x}_\ell^{(n)} &= \vec{x}^{(n)} + A_\ell^* \vec{c}, \\ \vec{c} &= (D_\ell + \Omega_\ell L_\ell)^{-1}\Omega_\ell \left(\vec{b}_\ell - A_\ell \vec{x}^{(n)}\right). \end{aligned}$$

We eliminate the scalar vector and attain

$$\vec{x}_\ell^{(n)} = (I - A_\ell^*(D_\ell + \Omega_\ell L_\ell)^{-1}\Omega_\ell A_\ell)\vec{x}^{(n)} + A_\ell^*(D_\ell + \Omega_\ell L_\ell)^{-1}\Omega_\ell \vec{b}_\ell.$$

We then aggregate the leaf operators as follows:

$$D = \begin{pmatrix} D_{\ell_1} & 0 & ... & 0 \\ 0 & D_{\ell_2} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & D_{\ell_t} \end{pmatrix}, \Omega = \begin{pmatrix} \Omega_{\ell_1} & 0 & ... & 0 \\ 0 & \Omega_{\ell_2} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \Omega_{\ell_t} \end{pmatrix},$$

$$\vec{\mathfrak{b}} = \begin{pmatrix} \vec{b}_{\ell_1} \\ \vec{b}_{\ell_2} \\ \vdots \\ \vec{b}_{\ell_t} \end{pmatrix}, L = \begin{pmatrix} L_{\ell_1} & 0 & ... & 0 \\ 0 & L_{\ell_2} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & L_{\ell_t} \end{pmatrix}, \mathcal{A} = \begin{pmatrix} A_{\ell_1} \\ A_{\ell_2} \\ \vdots \\ A_{\ell_t} \end{pmatrix},$$

$$W = \begin{pmatrix} w(r,\ell_1)I_{\dim(\Omega_{\ell_1})} & 0 & ... & 0 \\ 0 & w(r,\ell_2)I_{\dim(\Omega_{\ell_2})} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & w(r,\ell_t)I_{\dim(\Omega_{\ell_t})} \end{pmatrix}.$$

The estimate obtained from the pooling stage of the nth iteration can be expressed in terms of these matrices,

$$\vec{x}^{(n+1)} = \sum_{\ell \in \mathcal{L}} w(r,\ell)\vec{x}_{\ell}^{(n)} = B^{\vec{\omega}}\vec{x}^{(n)} + \vec{b}^{\vec{\omega}} \tag{18}$$

where

$$B^{\vec{\omega}} = I - \mathcal{A}^*(D + \Omega L)^{-1}W\Omega\mathcal{A}, \tag{19}$$

$$\vec{b}^{\vec{\omega}} = \mathcal{A}^*(D + \Omega L)^{-1}W\Omega\vec{\mathfrak{b}}. \tag{20}$$

Note that there exists a vector $\vec{h}$ such that $Q^{\vec{\omega}}\vec{x} = P^{\vec{\omega}}\vec{x} + \vec{h}$ for every $\vec{x}$. Then, from Equation (18) and the linearity of $P^{\vec{\omega}}$ and $B^{\vec{\omega}}$, we have $B^{\vec{\omega}} = P^{\vec{\omega}}$ and $\vec{b}^{\vec{\omega}} = \vec{h}$.

Proposition 4.1. *Suppose the relaxation parameters are admissible and $B^{\vec{\omega}}\vec{x} = \lambda\vec{x}$ for some $\vec{x} \neq \vec{0}$. Then $\lambda = 1$ or $|\lambda| < 1$, and*

(1) $\lambda = 1$ if and only if $\vec{x} \in \mathcal{R}(A^*)^{\perp}$,
(2) $|\lambda| < 1$ if and only if $\vec{x} \in \mathcal{R}(A^*)$.

Proof. Suppose $P^{\vec{\omega}}\vec{x} = \lambda\vec{x}$ for some $\vec{x} \neq \vec{0}$. By Lemma 2.1, we note that $|\lambda| \leq 1$. Let P be the orthogonal projection onto $\mathcal{R}(A^*)^{\perp}$. Then we find

$$\lambda P\vec{x} + \lambda(I - P)\vec{x} = \lambda\vec{x} = P^{\vec{\omega}}\vec{x} = P^{\vec{\omega}}P\vec{x} + P^{\vec{\omega}}(I - P)\vec{x} = P\vec{x} + P^{\vec{\omega}}(I - P)\vec{x}.$$

By uniqueness of the decomposition in $\mathcal{R}(A^*) \oplus \mathcal{R}(A^*)^{\perp}$, we have

$$P\vec{x} = \lambda P\vec{x},$$
$$P^{\vec{\omega}}(I - P)\vec{x} = \lambda(I - P)\vec{x}.$$

If $\lambda \neq 1$, then $P\vec{x} = \vec{0}$ so that $\vec{x} = (I - P)\vec{x} \in \mathcal{R}(A^*)$. From this observation and Proposition 3.5, we find that $|\lambda| < 1$. Now suppose $\lambda = 1$. Then, by Proposition 3.5, $(I - P)\vec{x} \in \mathcal{R}(A^*)^{\perp}$ so that $\vec{x} = P\vec{x} \in \mathcal{R}(A^*)^{\perp}$. The sufficient statement of (1) is straightforward, and (2) follows. □

Lemma 4.2. *Suppose the relaxation parameters are admissible. Let $\vec{x}^{(0)} \in \mathcal{R}(A^*)$. The sequence $\{\vec{x}^{(n)}\}$ converges to the fixed point of the mapping $\vec{x} \in \mathcal{R}(A^*) \mapsto B^{\vec{\omega}}\vec{x} + \vec{b}^{\vec{\omega}}$. Precisely, the sequence converges to*

$$(I - B^{\vec{\omega}})|_{\mathcal{R}(A^*)}^{-1}\vec{b}^{\vec{\omega}} = \sum_{j=0}^{\infty}(B^{\vec{\omega}})^j\vec{b}^{\vec{\omega}}.$$

Proof. Throughout the proof, we assume that every operator is restricted to $\mathcal{R}(A^*)$. From Proposition 4.1, there exists an induced matrix norm $\|\cdot\|$ such that $\|B^{\vec{\omega}}\| < 1$. Note that, with respect to this norm, $(B^{\vec{\omega}})^n$ converges to the zero matrix and $(B^{\vec{\omega}})^{n-1} + \ldots + B^{\vec{\omega}} + I$ converges to the matrix $(I - B^{\vec{\omega}})^{-1}$. Then

$$\vec{x}^{(n)} = (B^{\vec{\omega}})^n \vec{x}^{(0)} + ((B^{\vec{\omega}})^{n-1} + \ldots + B^{\vec{\omega}} + I)\vec{b}^{\vec{\omega}} \to (I - B^{\vec{\omega}})^{-1}\vec{b}^{\vec{\omega}} =: \vec{z}.$$

Note that $\vec{z} \in \mathcal{R}(A^*)$ and that $\vec{z} = B^{\vec{\omega}}\vec{z} + \vec{b}^{\vec{\omega}}$, as desired. □

Remark 4.3. We observe that, in general, the sequence $\{\vec{x}^{(n)}\}$ converges to

$$\vec{y} = \sum_{j=0}^{\infty} (B^{\vec{\omega}})^j \vec{b}^{\vec{\omega}} + P\vec{x}^{(0)} \tag{21}$$

where P is the orthogonal projection onto $\mathcal{N}(A)$. Hence, it is preferable to choose $\vec{x}^{(0)} \in \mathcal{R}(A^*)$ (e.g., $\vec{x} = \vec{0}$) so that the norm of the vector in Equation (21) is minimized.

Theorem 4.4. *Let $\vec{x}^{(0)} \in \mathcal{R}(A^*)$. The distributed Kaczmarz algorithm with admissible relaxation parameters converges to the vector $\vec{y}$ in Equation (21). If the system is inconsistent and $\Omega = s\widetilde{\Omega}$ where $s \in (0, 1]$ and $\widetilde{\Omega}$ is fixed, then $\vec{y} = \vec{y}_M + o(s)$ where $\vec{y}_M$ minimizes the functional*

$$\vec{x} \in \mathcal{R}(A^*) \mapsto \langle D^{-1}W\widetilde{\Omega}(\vec{\mathfrak{b}} - \mathcal{A}\vec{x}), \vec{\mathfrak{b}} - \mathcal{A}\vec{x}\rangle.$$

Proof. With Lemma 4.2, the proof is similar to the proof of Theorem V.3.9. in [13]. Nonetheless, we provide a self-contained proof for clarification of our adaptation. First, by Lemma 4.2, we have that the sequence $\{\vec{x}^{(n)}\}$ converges to the vector $\vec{y}$ satisfying $\vec{y} = B^{\vec{\omega}}\vec{y} + \vec{b}^{\vec{\omega}}$, that is

$$\mathcal{A}^*(D + \Omega L)^{-1}W\Omega\mathcal{A}\vec{y} = \mathcal{A}^*(D + \Omega L)^{-1}W\Omega\vec{\mathfrak{b}}. \tag{22}$$

Note that $\vec{y}_M$ minimizes $\|D^{-1/2}W^{1/2}\widetilde{\Omega}^{1/2}(\vec{\mathfrak{b}} - \mathcal{A}\vec{x})\|$ if and only if

$$(D^{-1/2}W^{1/2}\widetilde{\Omega}^{1/2}\mathcal{A})^*(D^{-1/2}W^{1/2}\widetilde{\Omega}^{1/2}\mathcal{A})\vec{y}_M = (D^{-1/2}W^{1/2}\widetilde{\Omega}^{1/2}\mathcal{A})^*D^{-1/2}W^{1/2}\widetilde{\Omega}^{1/2}\vec{\mathfrak{b}}$$

(see Theorem 1.1 of IV.1 in [13]), that is

$$\mathcal{A}^*D^{-1}W\widetilde{\Omega}\mathcal{A}\vec{y}_M = \mathcal{A}^*D^{-1}W\widetilde{\Omega}\vec{\mathfrak{b}}. \tag{23}$$

Substituting $\Omega = s\widetilde{\Omega}$ into Equation (22), we have

$$\mathcal{A}^*(D + s\widetilde{\Omega}L)^{-1}W\widetilde{\Omega}\mathcal{A}\vec{y} = \mathcal{A}^*(D + s\widetilde{\Omega}L)^{-1}W\widetilde{\Omega}\vec{\mathfrak{b}}. \tag{24}$$

From Equations (23) and (24), we observe that $\vec{y} = \vec{y}_M + o(s)$, as desired. □

Remark 4.5. The minimizer of the functional in Theorem 4.4 is the weighted least-squares solution of

$$\vec{x} \in \mathcal{R}(A^*) \mapsto \sum_{v \in \mathcal{V}} (\widetilde{\Omega})_v \left(\sum_{\ell \,:\, v \preceq \ell} w(r, \ell) \right) \frac{|b_v - \vec{a}_v^* \vec{x}|^2}{\|\vec{a}_v\|^2}.$$

We note that there is a trade-off between the convergence rate of the algorithm and the approximation error; that is, the algorithm converges more slowly as s approaches zero. Indeed, as is known [4,13,6], the convergence rate decreases as the relaxation parameters decrease in the classical and (uniform) distributed cases. The same phenomenon occurs in the case that the relaxation parameters are equation dependent as illustrated for a particular example in Fig. 5.

5. Leaf subnetworks

In this section, we consider the particular situation in which the subnetworks consist of leaves. We derive a concise expression for the norm of $P^{\vec{\omega}}_{G_i}$ restricted to the subspace $\mathcal{H}_i := \operatorname{span}\{\vec{a}_u \,:\, u \in G_i\}$ and provide sufficient upper-bounds on the relaxation parameters for the vertices in G_i to guarantee admissibility. We recall that the Gram matrix $\mathcal{G}(\vec{x}_1, \vec{x}_2, ..., \vec{x}_t)$ is the $t \times t$ matrix of inner-products,

$$\mathcal{G}(\vec{x}_1, \vec{x}_2, ..., \vec{x}_t) = \begin{pmatrix} \langle \vec{x}_1, \vec{x}_1 \rangle & \langle \vec{x}_1, \vec{x}_2 \rangle & ... & \langle \vec{x}_1, \vec{x}_t \rangle \\ \langle \vec{x}_2, \vec{x}_1 \rangle & \langle \vec{x}_2, \vec{x}_2 \rangle & ... & \langle \vec{x}_2, \vec{x}_t \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \vec{x}_t, \vec{x}_1 \rangle & \langle \vec{x}_t, \vec{x}_2 \rangle & ... & \langle \vec{x}_t, \vec{x}_t \rangle \end{pmatrix}.$$

We further denote the diagonal matrix $\mathcal{D}_i$ associated with the leaf subnetwork $G_i = \{\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,t_i}\}$ by

$$\mathcal{D}_i = \begin{pmatrix} \frac{w(g_i, \ell_{i,1})\omega_{\ell_{i,1}}}{\|\vec{a}_{\ell_{i,1}}\|^2} & 0 & \cdots & 0 \\ 0 & \frac{w(g_i, \ell_{i,2})\omega_{\ell_{i,2}}}{\|\vec{a}_{\ell_{i,2}}\|^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{w(g_i, \ell_{i,t_i})\omega_{\ell_{i,t_i}}}{\|\vec{a}_{\ell_{i,t_i}}\|^2} \end{pmatrix}.$$

We denote the spectrum (collection of eigenvalues) of a matrix A by $\sigma(A)$, and we denote its spectral radius by $\rho(A) = \max\{|\lambda| \,:\, \lambda \in \sigma(A)\}$.

Theorem 5.1. *Suppose $G_i = \{\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,t_i}\}$. Then*

$$\|P^{\vec{\omega}}_{G_i}|_{\mathcal{H}_i}\| = \max\{|1 - \lambda| \,:\, \lambda \in \sigma(\mathcal{D}_i \mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})) \setminus \{0\}\}.$$

Proof. From Equations (4), (7) and (10), we have

$$P_{G_i}^{\vec{\omega}} = I - \sum_{j=1}^{t_i} \frac{w(g_i, \ell_{i,j})\omega_{\ell_{i,j}}}{\|\vec{a}_{\ell_{i,j}}\|^2} \vec{a}_{\ell_{i,j}} \vec{a}_{\ell_{i,j}}^*. \tag{25}$$

Now let $K_{G_i} := \sqrt{\mathcal{D}_i}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})^*$. Then, Equation (25) may be expressed as $P_{G_i}^{\vec{\omega}} = I - K_{G_i}^* K_{G_i}$. Note that $\mathcal{H}_i$ is an invariant subspace for $K_{G_i}^* K_{G_i}$. Hence, from the spectral mapping theorem, we find

$$\sigma(P_{G_i}^{\vec{\omega}}|_{\mathcal{H}_i}) = 1 - \sigma(K_{G_i}^* K_{G_i}|_{\mathcal{H}_i}).$$

We claim that $\sigma(K_{G_i}^* K_{G_i}|_{\mathcal{H}_i})$ is precisely the collection of all of the nonzero eigenvalues of $K_{G_i}^* K_{G_i}$. Suppose, to the contrary, that there exists a nonzero vector $\vec{x} \in \mathcal{H}_i$ such that $K_{G_i}^* K_{G_i}\vec{x} = 0$. Then $K_{G_i}\vec{x} \in R(K_{G_i}) \cap \mathcal{N}(K_{G_i}^*)$ implying $K_{G_i}\vec{x} = 0$, yet this leads to the contradiction that $\vec{x} \in \mathcal{H}_i \cap \mathcal{H}_i^\perp$ or $\vec{x} = 0$. It is well-known that $K_{G_i}^* K_{G_i}$ and $K_{G_i} K_{G_i}^*$ have the same nonzero eigenvalues and

$$\begin{aligned}\sigma(K_{G_i} K_{G_i}^*) &= \sigma\left(\sqrt{\mathcal{D}_i}\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})^T \sqrt{\mathcal{D}_i}\right) \\ &= \sigma(\mathcal{D}_i \mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})),\end{aligned}$$

concluding the proof. □

Corollary 5.1.1. *Suppose $G_i = \{\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,t_i}\}$. If*

$$0 < \omega_{\ell_{i,j}} < \frac{2\|\vec{a}_{\ell_{i,j}}\|^2}{w(g_i, \ell_{i,j})\rho(\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}}))} \qquad \textit{for all} \qquad 1 \le j \le t_i,$$

then $\|P_{G_i}^{\vec{\omega}}|_{\mathcal{H}_i}\| < 1$.

Proof. Since $\mathcal{D}_i$ and $\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})$ are positive semi-definite matrices, the eigenvalues of $\mathcal{D}_i\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})$ are nonnegative. Therefore, by Theorem 5.1, it suffices to show $\lambda < 2$ for $\lambda \in \sigma(\mathcal{D}_i\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}}))$. Let j be the index for the largest diagonal entry in $\mathcal{D}_i$. By Theorem 8.12 in [22], we have

$$\rho(\mathcal{D}_i\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})) \le \frac{w(g_i, \ell_{i,j})\omega_{\ell_{i,j}}}{\|\vec{a}_{\ell_{i,j}}\|^2} \rho(\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})) < 2,$$

as desired. □

Remark 5.2. It is not unusual to require that the rows of A are normalized (i.e., $\|\vec{a}_u\| = 1$ for all u). Further, for the case that $\rho(\mathcal{G}(\vec{a}_{\ell_{i,1}}, \vec{a}_{\ell_{i,2}}, ..., \vec{a}_{\ell_{i,t_i}})) \approx 1$, the relaxation parameters for the vertices in G_i are admissible if

$$\omega_{\ell_{i,j}} \lesssim \frac{2}{w(g_i, \ell_{i,j})} \qquad \text{for all} \qquad 1 \leq j \leq t_i.$$

This upper-bound is greater than the usual bound in literature and can be drastically larger than 2, depending on the associated weights. For example, if the weights are uniformly distributed, then the upper-bound is $2t_i \geq 2$.

We end this section by observing that it is necessary and sufficient to check that Ω_1 satisfies the admissibility conditions in Theorem 4.4 when the subnetwork consists of only leaves. We note that this need not hold for other subnetworks.

Theorem 5.3. *Suppose $G_i = \{\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,t_i}\}$. Let $\Omega = s\widetilde{\Omega}$ for some $s \in (0, 1]$ as in Theorem 4.4. If $\widetilde{\Omega}$ satisfies the admissibility conditions, then Ω satisfies the admissibility conditions.*

Proof. We check condition (2) in Definition 1. Let $\vec{x} \in \mathcal{H}_i$. Then

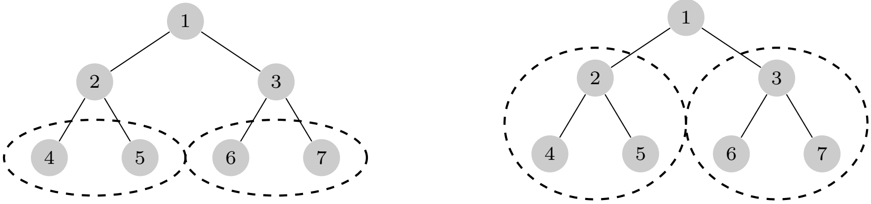
$$\begin{aligned} \|P^{\Omega}_{G_i}\vec{x}\| &= \left\| \sum_{j=1}^{t_i} w(g_i, \ell_{i,j}) P^{\Omega}_{\ell_{i,j}} \vec{x} \right\| \\ &= \left\| \sum_{j=1}^{t_i} w(g_i, \ell_{i,j}) \left[(1-s)I + sP^{\widetilde{\Omega}}_{\ell_{i,j}}\right] \vec{x} \right\| \\ &= \left\| (1-s)\vec{x} + sP^{\widetilde{\Omega}}_{G_i}\vec{x} \right\| \\ &\leq (1-s)\|\vec{x}\| + s\alpha_i\|\vec{x}\| \\ &= [(1-s)1 + s\alpha_i]\|\vec{x}\|, \end{aligned}$$

where the coefficient is strictly less than one as it is a convex sum of 1 and α_i. □

6. Experiments

In this section we implement our algorithm on various kinds of distributed networks corresponding to randomly generated systems of equations and systems perturbed from an orthogonal coefficient matrix. The latter illustrates the point of Remark 5.2. Specifically, we analyze two scenarios: (1) comparing different subnetwork structures for a given network and (2) comparing different network structures for a given system of equations.

For the first experiment, we consider a 7-node binary network and compare leaf subnetworks to extended subnetworks as depicted in Fig. 2. We assign the relaxation parameters as follows: set $\omega_v = 1.5$ if the node v is not associated with a subnetwork; set $\omega_v = \omega$ if the node v belongs to a subnetwork. Then we calculate the spectral radius of the operator $P^{\vec{\omega}}$ as a function of ω. For a baseline, we include the spectral radius of the network with no subnetwork structures in this set-up, which we label uniform. See Fig. 3.



(A) Leaf subnetworks

(B) Extended subnetworks

Fig. 2. The 7-node binary network with its subnetworks.

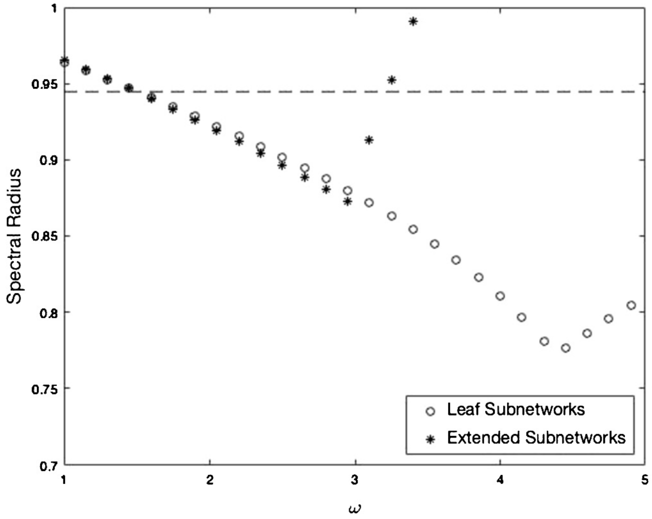
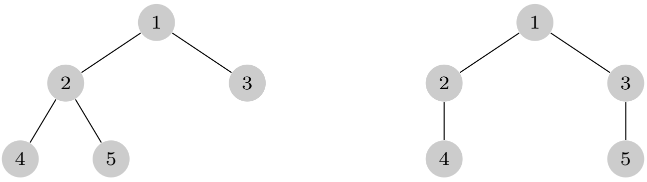


Fig. 3. Spectral radius of $P^{\vec{\omega}}$ for two subnetwork structures. The dashed line represents a network with uniformly distributed relaxation parameters $\omega = 1.5$.

The numerical experiments suggest that the leaf subnetwork structures are more practical than the extended subnetwork structures for two reasons. In general, the spectral radius of $P^{\vec{\omega}}$ is decreasing for ω slightly larger than 1.5 and is, therefore, comparatively smaller than the baseline established by the uniform case in which all of the parameters are set to 1.5. In this situation, we find that the spectral radius tends to be smaller than the baseline for relatively large relaxation parameters in the case of the leaf subnetwork structures and less so in the case of the extended subnetwork structures. This implies that parameter selection is more reliable for leaf subnetworks than for their extended counterparts. Second, the spectral radius is often smaller for leaf subnetworks when the parameters are large. We believe that these observations are a consequence of the pooling stage which is a poor method of producing the next iterate in the distributed Kaczmarz algorithm from the leaf estimates. The depth of the extended network increases the number of overrelaxed projections, often leading to adverse results in the pooling stage.

For the second experiment, we consider the different network structures given in Fig. 4 for a system of five equations. We compare the network structures for two kinds of systems: (1) entries of A are randomly selected from a uniform distribution over $[0, 1]$



(A) Network I (B) Network II

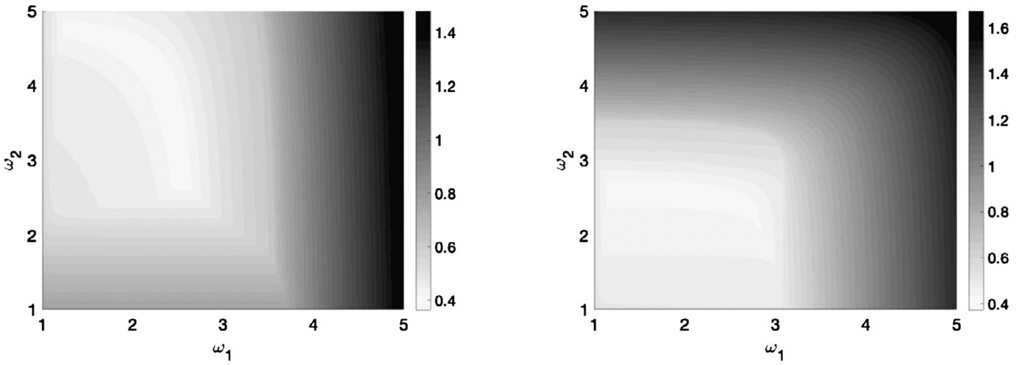
Fig. 4. Two networks for a system of five equations.

Table 1
Comparing networks I and II for a nearly orthogonal system.

Network Type	Leaf subnetworks			Uniform	
	$(\omega_1, \omega_2)_{\text{opt}}$	$\rho(P^{\vec{\omega}})$	$\lVert A\vec{x}^{(10)} - \vec{b}\rVert$	$\rho(P^{\vec{\omega}})$	$\lVert A\vec{x}^{(10)} - \vec{b}\rVert$
I	$(2.27, 3.93)$	0.36532	3.479e-4	0.66617	3.6441e-3
II	$(1.49, 2.52)$	0.37492	3.4554e-4	0.47598	5.7009e-4

Table 2
Comparing networks I and II for a random system.

Network Type	Leaf subnetworks			Uniform	
	$(\omega_1, \omega_2)_{\text{opt}}$	$\rho(P^{\vec{\omega}})$	$\lVert A\vec{x}^{(1500)} - \vec{b}\rVert$	$\rho(P^{\vec{\omega}})$	$\lVert A\vec{x}^{(1500)} - \vec{b}\rVert$
I	$(7.92, 8.06)$	0.98844	1.5743e-8	0.99626	1.7049e-3
II	$(4.57, 3.90)$	0.99512	8.3191e-4	0.99619	1.3288e-3



(A) Network I (B) Network II

Fig. 5. Spectral radii for a nearly orthogonal system.

and (2) A is nearly orthogonal by perturbing the identity. Further, the entries of $\vec{b}$ are also randomly selected from a uniform distribution over $[0, 1]$. We present results of numerical experiments for the nearly orthogonal system in Table 1 and for the random system in Table 2. We include the optimal relaxation parameters that yield the minimum spectral radius along with an error estimate of an iterate using the optimal parameters. Fig. 5 shows how the spectral radius varies with respect to the relaxation parameter for networks I and II with leaf subnetworks. For network I, ω_1 is on node 3, and ω_2 is on the leaf subnetwork composed of nodes 4 and 5. For network II, ω_1 is on node 5, and ω_2 is on node 4.

For both the nearly orthogonal and random systems, we see that the relaxation parameter is allowed to be larger than 2 to achieve convergence. Note also that the spectral radius $\rho(P^{\vec{\omega}})$ for systems with leaf subnetworks is smaller than the uniform system; hence we see better performance. For the nearly orthogonal systems with leaf subnetworks, we do not need many iterations of the algorithm to achieve a smaller error than the uniform system. However, for the random system, we need many more iterations to achieve this smaller error.

Declaration of competing interest

The authors declare that they have no competing interests.

Acknowledgements

Riley Borgard, Haley Duba, Chloe Makdad, Jay Mayfield, and Randal Tuggle were supported by the National Science Foundation through the REU award #1457443. Steven Harding and Eric Weber were supported by the National Science Foundation and the National Geospatial-Intelligence Agency under award #1830254. Eric Weber was also supported under award #1934884.

References

[1] Yair Censor, Dan Gordon, Rachel Gordon, Component averaging: an efficient iterative parallel algorithm for large and sparse unstructured problems, Parallel Comput. 27 (6) (2001) 777–808. MR 1823354.

[2] Xuemei Chen, The Kaczmarz Algorithm, Row Action Methods, and Statistical Learning Algorithms, Frames and Harmonic Analysis, Contemp. Math., vol. 706, Amer. Math. Soc., Providence, RI, 2018, pp. 115–127. MR 3796634.

[3] G. Cimmino, Calcolo approssimato per soluzioni dei sistemi di equazioni lineari, Ric. Sci. XVI, Ser. II, Anno IX 1 (1938) 326–333.

[4] P.P.B. Eggermont, G.T. Herman, A. Lent, Iterative algorithms for large partitioned linear systems, with applications to image reconstruction, Linear Algebra Appl. 40 (1981) 37–67.

[5] C. Hamaker, D.C. Solmon, The angles between the null spaces of X rays, J. Math. Anal. Appl. 62 (1) (1978) 1–23.

[6] Chinmay Hegde, Fritz Keinert, Eric S. Weber, A Kaczmarz algorithm for solving tree based distributed systems of equations, Appl. Numer. Harmon. Anal. (2020), in press, http://arxiv.org/abs/1904.05732, 2019.

[7] G.T. Herman, A. Lent, H. Hurwitz, A storage-efficient algorithm for finding the regularized solution of a large, inconsistent system of equations, J. Inst. Math. Appl. 25 (4) (1980) 361–366. MR 578083.

[8] Björn Johansson, Maben Rabi, Mikael Johansson, A randomized incremental subgradient method for distributed optimization in networked systems, SIAM J. Optim. 20 (3) (2009) 1157–1170.

[9] Stefan Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, Bull. Int. Acad. Po. Sci. Lett. (1937) 355–357.

[10] Goutham Kamath, Paritosh Ramanan, Wen-Zhan Song, Distributed randomized Kaczmarz and applications to seismic imaging in sensor network, in: 2015 International Conference on Distributed Computing in Sensor Systems, 06 2015, pp. 169–178.

[11] Ji Liu, Stephen J. Wright, Srikrishna Sridhar, An asynchronous parallel randomized Kaczmarz algorithm, arXiv preprint, arXiv:1401.4780, 2014.

[12] Nicolas Loizou, Peter Richtárik, Revisiting randomized gossip algorithms: general framework, convergence rates and novel block and accelerated protocols, arXiv:1905.08645, 2019.

[13] F. Natterer, The Mathematics of Computerized Tomography, Classics in Applied Mathematics, vol. 32, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001, reprint of the 1986 original. MR 1847845.

[14] Ion Necoara, Faster randomized block Kaczmarz algorithms, SIAM J. Matrix Anal. Appl. 40 (4) (2019) 1425–1452. MR 4036092.

[15] Deanna Needell, Nathan Srebro, Rachel Ward, Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm, Math. Program., Ser. A 155 (1–2) (2016) 549–573. MR 3439812.

[16] Deanna Needell, Joel A. Tropp, Paved with good intentions: analysis of a randomized block Kaczmarz method, Linear Algebra Appl. 441 (2014) 199–221. MR 3134343.

[17] Deanna Needell, Ran Zhao, Anastasios Zouzias, Randomized block Kaczmarz method with projection for solving least squares, Linear Algebra Appl. 484 (2015) 322–343. MR 3385065.

[18] Devavrat Shah, Gossip algorithms, Found. Trends Netw. 3 (1) (2008) 1–125.

[19] Thomas Strohmer, Roman Vershynin, A randomized Kaczmarz algorithm with exponential convergence, J. Fourier Anal. Appl. 15 (2) (2009) 262–278.

[20] Kunio Tanabe, Projection method for solving a singular system of linear equations and its applications, Numer. Math. 17 (1971) 203–214. MR 293824.

[21] John Tsitsiklis, Dimitri Bertsekas, Michael Athans, Distributed asynchronous deterministic and stochastic gradient optimization algorithms, IEEE Trans. Autom. Control 31 (9) (1986) 803–812.

[22] F. Zhang, Matrix Theory, Universitext, vol. 32, Springer-Verlag, New York, 2011, reprint of the 1986 original.

[23] Anastasios Zouzias, Nikolaos M. Freris, Randomized extended Kaczmarz for solving least squares, SIAM J. Matrix Anal. Appl. 34 (2) (2013) 773–793. MR 3069089.