Distributed algorithms to determine eigenvectors of matrices on spatially distributed networks

Nazar Emirov, Cheng Cheng, Qiyu Sun and Zhihua Qu

Abstract—Eigenvectors of matrices on a network have been used for understanding spectral clustering and influence of a vertex. For matrices with small geodesic-width, we propose a distributed iterative algorithm in this letter to find eigenvectors associated with their given eigenvalues. We also consider the implementation of the proposed algorithm at the vertex/agent level in a spatially distributed network.

Keywords: Eigenvector, preconditioned gradient descent algorithm, spatially distributed network.

I. INTRODUCTION

Spatially distributed networks (SDNs) consist of a large amount of agents, and each agent is equipped with subsystems for limited data processing and direct communication link to its “neighboring” agents within communication range. SDNs appear in (wireless) sensor networks, smart grids, social networks and many real world applications [1]–[3]. In this letter, we describe the topological structure of an SDN by a finite graph \( G \). We consider its “neighboring” agents within communication range. SDNs for limited data processing and direct communication link to a large amount of agents, and each agent is equipped with subsystems associated with their given eigenvalues. We also consider the implementation of the proposed algorithm at the vertex/agent level in a spatially distributed network.

II. A DISTRIBUTED ITERATIVE ALGORITHM FOR DETERMINING EIGENVECTORS

Let \( G = (V, E) \) be a connected, undirected and unweighted graph of order \( N \). Denote the set of all s-hop neighbors of a vertex \( i \in V \) by \( B(i, s) = \{ j \in V, \rho(j, i) \leq s \}, s \geq 0 \). For a complex-valued matrix \( A = (A(i, j))_{i,j \in V} \) with small geodesic-width \( \omega(A) \), we denote its Hermitian transpose by \( A^{*} \) and define the diagonal preconditioning matrix \( P_{A} \) with diagonal elements

\[
P_{A}(i, i) := \max_{k \in B(i, n)(A)} \left\{ \max_{j \in B(k, n)(A)} |A(j, k)|, \sum_{j \in B(k, n)(A)} |A(k, j)| \right\}, \quad i \in V \tag{II.1}
\]

In this section, we introduce a distributed iterative algorithm to find eigenvectors of a complex-valued matrix.

Theorem II.1. Let \( A \) be a complex-valued matrix on the graph \( G \) of order \( N \), \( P_{A} \) be the diagonal matrix in (II.1), and \( Q \) be a nonsingular diagonal matrix such that

\[
Q - P_{A} \text{ is positive semidefinite}. \tag{II.2}
\]

Then for any initial \( x_{0} \in \mathbb{C}^{N} \), the sequence \( x_{n}, n \geq 0 \), defined inductively by

\[
x_{n+1} = (I - Q^{-2}A^{*}A)x_{n}, \tag{II.3}
\]

converges exponentially to either the zero vector or an eigenvector associated with the zero eigenvalue of the matrix \( A \).

The proof of Theorem II.1 will be given in Appendix A. Take a positive constant \( c \) and define a diagonal matrix \( Q_c = \text{diag}(Q_c(i, i))_{i \in V} \) by

\[
Q_c(i, i) = \max(P_{A}(i, i), c), \quad i \in V. \tag{II.4}
\]

Then \( Q_c \) is a nonsingular diagonal matrix satisfying (II.2) and it can be constructed at the vertex level, since the preconditioning matrix \( P_{A} \) can, see [15] Algorithm II.1.

Let \( H = (H(i, j))_{i,j \in V} \) be a matrix with small geodesic-width \( \omega(H) \) and \( \lambda \) be its eigenvalue. By selecting a random initial \( x_0 \) with entries i.i.d. on \([0,1] \), and applying the iterative algorithm (II.3) to the matrix \( A = H - \lambda I \) or \( \lambda I - H \), we obtain...
Algorithm II.1 Realization of the PGDA at a vertex $i \in V$.

Inputs: The total iteration number $M$, the geodesic-width $\omega(W)$ of the matrix $W = (W(i,j))_{i,j \in V}$, the set $B(i, \omega(W))$ of $\omega(W)$-hop neighbors of the vertex $i$, the eigenvalue $\lambda$ of the matrix $W$, entries $H(i,j)$ and $H(j,i)$, $j \in B(i, \omega(W))$ in the $i$-th row and column of the matrix $W$, and the $i$-th diagonal entry $Q(i,i)$ of the matrix $Q$.

Pre-iteration: Compute $A(i,j) = H(i,j) - \lambda \delta(i,j)$ and $\hat{A}(j,i) = (Q(i,i))^{-1/2} (H(j,i) - \lambda \delta(j,i))$ for $j \in B(i, \omega(W))$, where $\delta$ is the Kronecker delta.

Initial: Select the $i$-th component $x_0(i) \in [0,1]$ of the initial vector $x_0$ randomly, and set $n = 0$.

Iteration:
1. Send $x_n(i)$ to all neighbors $k \in B(i, \omega(W)) \setminus \{i\}$ and receive $x_n(k)$ from neighbors $k \in B(i, \omega(W)) \setminus \{i\}$.
2. Evaluate $\tilde{x}_n(i) = \sum_{j \in B(i, \omega(W))} A(i,j) x_n(j)$.
3. Send $\tilde{x}_n(i)$ to all neighbors $k \in B(i, \omega(W)) \setminus \{i\}$ and receive $\tilde{x}_n(k)$ from neighbors $k \in B(i, \omega(W)) \setminus \{i\}$.
4. Evaluate $\tilde{x}_n(i) = \tilde{x}_n(i) - \tilde{x}_n(i)$ and $n = n + 1$.
5. If $n \leq M$, go to Output. Otherwise, go to 1.

Output: $u(i) = x_M(i)$, where $u = (u(i))_{i \in V}$ is the eigenvector.

III. PRINCIPAL EIGENVECTORS OF HERMITIAN MATRICES

In this section, we consider finding eigenvectors associated with the minimal/maximal eigenvalue of a Hermitian matrix on a graph $G = (V,E)$ of order $N$ in a distributed manner.

Theorem III.1. Let $A = (A(i,j))_{i,j \in V}$ be a positive semidefinite matrix on the graph $G$ with its geodesic-width $\omega(A)$, and $Q^sym = \text{diag}(Q^sym(i,i))_{i \in V}$ be a nonsingular diagonal matrix satisfying

$$Q^sym(i,i) \geq \sum_{j \in B(i, \omega(A))} |A(i,j)|, \quad i \in V.$$ (III.1)

Then for any $x_0 \in \mathbb{C}^N$, the sequence $x_n, n \geq 1$, defined by $x_{n+1} = (I - (Q^sym)^{-1} A) x_n$, (III.2) converges exponentially to either the zero vector or an eigenvector associated with the zero eigenvalue of the matrix $A$.

Proof: Following the argument used in [15, Theorem III.1] and applying (III.1) and (III.2), we obtain that $Q^sym - A$ is a positive semidefinite. This together with the positive semidefiniteness of the matrix $A$ implies that all eigenvalues of the Hermitian matrix $B^sym := I - (Q^sym)^{-1/2} A (Q^sym)^{-1/2}$ are in the unit interval $[0,1]$, cf. [A3] in Appendix A. Applying similar arguments in the proof of Theorem II.1 with $Q$ and $A^* A$ replaced by $(Q^sym)^{-1/2}$ and $A$ respectively, we obtain

$$||Q^sym^{1/2} (x_n - u)||_2 \leq ||Q^sym^{1/2} x_0||_2 r^n, \quad n \geq 0 \quad (III.3)$$

for some vector $u \in \mathbb{C}^N$, where $r$ is the largest eigenvalue of $B^sym$ in $[0,1]$. This together with the nonsingularity of the matrix $Q^sym$ proves the exponential convergence of $x_n, n \geq 0$.

Taking limit in (III.2) proves $A u = 0$, and hence completes the proof.

Let $H$ be a Hermitian matrix with minimal eigenvalue $\lambda_{\min}$ and maximal eigenvalue $\lambda_{\max}$. Then $A_1 = H - \lambda_{\min} I$ and $A_2 = \lambda_{\max} I - H$ have eigenvalue zero and they are positive semidefinite. Then applying the iterative algorithm (III.2) to $A_1$ (resp. $A_2$) with a random initial $x_0$ having entries i.i.d. on $[0,1]$, we obtain the principal eigenvectors associated with minimal (resp. maximal) eigenvalues of the Hermitian matrix $H$ by Theorem III.1.

For a positive semidefinite matrix $A = (A(i,j))_{i,j \in V}$ with geodesic-width $\omega(A)$, a nonsingular diagonal matrix $Q^sym = \text{diag}(Q^sym(i,i))_{i \in V}$ satisfying (III.1) can be constructed at the vertex level by setting

$$Q^sym(i,i) = \max_{j \in B(i, \omega(A))} |A(i,j)|, \quad i \in V.$$ (III.4)

where $c$ is a positive constant, cf. (II.4). With the above selection of the preconditioning matrix as in (III.2), we can find eigenvectors associated with minimal/maximal eigenvalues of a Hermitian matrix by the distributed iterative algorithm (III.2), implementable at the vertex level, see Algorithm III.2. Following the terminology in [13], we call the algorithm (III.2) with a random initial having entries i.i.d. on $[0,1]$ as a symmetric preconditioned gradient descent algorithm, SPGDA for abbreviation. Comparing with Algorithm II.1 to find eigenvectors of the hyperlink matrix $W$, we can locally evaluate principal eigenvectors of the hyperlink matrix and hence identify the local influence of a vertex on its neighborhood.
Algorithm III.1 Realization of the SPGDA at a vertex \( i \in V \).

**Inputs:** The total iteration number \( M \), the geodesic-width \( \omega(A) \) of the positive semidefinite matrix \( A \), the set \( B(i, \omega(A)) \) of \( \omega(A) \)-hop neighbors of the vertex \( i \), entries \( A(i, j), j \in B(i, \omega(A)) \) in the \( i \)-th row of the matrix \( A \) and the \( i \)-th entry \( Q^y(i, i) \) of the diagonal matrix \( Q^y \).

**Pre-iteration:** Evaluate \( \tilde{A}(i, j) = (Q^y(i, i))^{-1}A(i, j), j \in B(i, \omega(A)) \).

**Initial:** Select \( x_0(i) \) randomly in \([0, 1]\), and set \( n = 0 \).

**Iteration:**
1. Send \( x_n(i) \) to all neighbors \( k \in B(i, \omega(A)) \setminus \{i\} \) and receive \( x_n(k) \) from neighbors \( k \in B(i, \omega(A)) \setminus \{i\} \).
2. Evaluate \( x_{n+1}(i) = x_n(i) - \sum_{j \in B(i, \omega(A))} \tilde{A}(i, j)x_n(j) \) and set \( n = n + 1 \).
3. Go to step 1 if \( n \leq M \), otherwise output.

**Output:** \( u(i) = y_M(i) \), where \( u = (u(i))_{i \in V} \).

of an arbitrary matrix, the Algorithm III.1 to find principal eigenvectors of a Hermitian matrix has less computational cost and communication expense in each iteration. Our numerical simulations in Section V also indicate that it may have faster convergence.

IV. EIGENVECTORS OF POLYNOMIAL FILTERS

Graph filter is a fundamental concept in graph signal processing and it has been used in many applications such as denoising and consensus of multi-agent systems [2, 7, 11, 12, 13, 20–26]. An elementary graph filter is a graph shift, which has 1 as its geodesic-width. Graph filters in most of literature are designed to be polynomials

\[
A = h(S_1, \ldots, S_d) = \sum_{l_1=0}^{L_1} \cdots \sum_{l_d=0}^{L_d} h_{l_1, \ldots, l_d} S_1^{l_1} \cdots S_d^{l_d}
\]

of commutative graph shifts \( S_1, \ldots, S_d \), i.e., \( S_k S_{k'} = S_{k'} S_k \) for all \( 1 \leq k, k' \leq d \), where the multivariate polynomial \( h(t_1, \ldots, t_d) = \sum_{l_1=0}^{L_1} \cdots \sum_{l_d=0}^{L_d} h_{l_1, \ldots, l_d} t_1^{l_1} \cdots t_d^{l_d} \) has polynomial coefficients \( h_{l_1, \ldots, l_d} \), \( 0 \leq l_1 \leq L_k \), \( 1 \leq k \leq d \) [29–31]. On the graph \( G = (V, E) \), a polynomial filter \( A \) in (IV.1) can be represented by a matrix \( A = (A(i, j))_{i,j \in V} \), which has geodesic-width no more than the degree of the polynomial \( h \), i.e., \( \omega(A) \leq L \).

Then we can apply the PGDA (resp. the SPGDA if \( A \) is Hermitian) to find eigenvectors associated with any given eigenvalue (resp. the minimal/maximal eigenvalues) on SDNs with communication range \( L \geq \sum_k L_k \). In this section, we propose iterative algorithms to determine eigenvectors associated with a polynomial graph filter, which can be implemented on an SDN with 1 as its communication range, i.e., direct communication exists between all adjacent vertices.

Observe that

\[
A^\ast = \sum_{l_1=0}^{L_1} \cdots \sum_{l_d=0}^{L_d} h_{l_1, \ldots, l_d} (S_1)^{l_1} \cdots (S_d)^{l_d}
\]

is a polynomial graph filter of commutative shifts \( S_1^*, \ldots, S_d^* \). Then applying Algorithm II.2 in [14] to implement the filtering procedure associated with polynomial graph filters \( A \) and \( A^\ast \), we can implement each iteration in the PGDA (II.3) and the SPGDA (II.2) in finite steps with each step including data exchanging between adjacent vertices only, see Algorithm IV.1 to determine eigenvectors associated with eigenvalue zero.

This concludes that eigenvectors for a polynomial graph filter on SDNs with communication range 1 can be obtained by applying Algorithm IV.1 in each iteration.

Now it remains to construct diagonal matrices satisfying (II.2) and (III.1) on SDNs with communication range 1. For the polynomial graph filter \( A \) in (IV.1), define diagonal matrices \( Q_c = \text{diag}(Q_c(i, i))_{i \in V} \) and \( Q^{y}(i, i) = \text{diag}(Q^{y}(i, i))_{i \in V} \) by

\[
Q_c(i, i) = \max_{\rho(j,i) \leq L} \{ \sum_{k \in V} \hat{A}(j, k), \sum_{k \in V} \hat{A}(k, j), c \}
\]

and

\[
Q^{y}(i, i) = \max_{k \leq L_c} \{ \sum_{k \in V} \hat{A}(j, k), c \}, i \in V,
\]

where \( c \) is a positive number, \( |S_k| = (|S_k(i,j)|)_{i,j \in V} \), \( 1 \leq k \leq d \), and

\[
(\hat{A}(i, j))_{i,j \in V} =: \hat{A} := \sum_{l_1=0}^{L_1} \cdots \sum_{l_d=0}^{L_d} h_{l_1, \ldots, l_d}|S_1|^{l_1} \cdots |S_d|^{l_d}.
\]

One may verify that \( |A(i, j)| \leq \hat{A}(i, j) \) for all \( i, j \in V \). Therefore the matrices \( Q_c \) in (IV.3) and \( Q^{y}(i, i) \) in (IV.4) satisfy (II.2) and (III.1) respectively. Moreover, as shown in Algorithm IV.2, they can be constructed at the vertex level in finite steps such that in each such step, each vertex needs to exchange data with adjacent vertices only.

V. NUMERICAL SIMULATIONS

Let \( G_N = (V_N, E_N), N \geq 2 \), be random geometric graphs with \( N \) vertices deployed on \([0, 1]^2 \) and an undirected edge between two vertices in \( V_N \) existing if their physical distance is not larger than \( \sqrt{2/N} \) [26, 29]. In this section, we consider finding eigenvectors associated with maximal eigenvalue 1 of lowpass spline filters \( H^{\text{pol}}_{0,m} = (I - L^{\text{sym}}/2)^m, m \geq 1 \),
we can use the conventional power iteration method with SGDASchur respectively, see Figure 1 for their performance. The above algorithms with $c$ are given by (II.1) [8], [15]. For the case that the constant in Figure 1 are the average of convergence errors $\|x\|_2$ in the logarithmic scale, where from left to right are lowpass spline filters $H_{0,m}$ of orders $m = 2, 3, 4$ on the random geometric graph $\mathcal{G}_{512}$.

**Algorithm IV.2** Construction of diagonal entries $\hat{Q}_c(i, i)$ and $\hat{Q}^{\text{sym}}_c(i, i)$ at a vertex $i \in V$ for a polynomial filter $\Lambda$.

**Inputs:** The positive constant $c$, polynomial coefficients $b_1, \ldots, b_d$, $0 \leq l_1 \leq L_1, \ldots, 0 \leq l_d \leq L_d$, of the polynomial filter $\Lambda$, entries $S_k(i, j)$ and $S_k(j, i)$ for all $1 \leq k \leq d$ and $j \in N_i$, the set of all adjacent vertices of the vertex $i$.

1. Apply Algorithm II.2 in [14] to implement the polynomial filter procedure $1 \rightarrow \Lambda_1$ at the vertex $i$. The input is the $i$-th entry 1 of the all-one vector $\mathbf{1}$ and the output is the $i$-th entry $a_1(i)$ of the vector $\mathbf{A}_1 = (a_1(k))_{k \in V}$.

2. Apply Step 1 with the same input but the filter $\hat{\Lambda}$ replaced by $\hat{\Lambda}_1$. The output is the $i$-th entry $a_2(i)$ of the vector $\hat{\Lambda}_1 \mathbf{1}$.

3. Evaluate $q_0(i) = \max(a_1(i), a_2(i), c)$ and set $l = 0$.

4. **Finite-step iteration:**

   4a) Send $q_l(i)$ to all adjacent vertices $k \in N_i$ and receive $q_l(k)$ from all adjacent vertices $k \in N_i$.

   4b) Compare $q_l(i)$ with $q_l(k), k \in N_i$ and define $q_{l+1}(i) = \max(q_l(i), \max_{k \in N_i} q_l(k))$ and set $l := l + 1$.

4c) Return to step 1 if $l \leq L_1 + \ldots + L_d$, go to Outputs otherwise.

**Outputs:** $\hat{Q}_c(i, i) = q_l(i)$ and $\hat{Q}^{\text{sym}}_c(i, i) = \max(a_1(i), c)$.

where $L^{\text{sym}}$ is the symmetric normalized Laplacian matrix on the graph $\mathcal{G}_N$ [29, 30]. In the simulations, we take $c = 0.01$ and use PGDA and PAGDAh to denote the PGDA with $\Lambda$ replaced by $I - H_{0,m}^\text{split}$ and $Q$ by $Q_c$ in (IV.1) and $Q_c$ in (IV.2) respectively, and similarly we use SPGDA and SPGDA1h to denote the SPGDA with $\Lambda$ replaced by $I - H_{0,m}^\text{split}$ and $Q$ by $Q_c^{\text{sym}}$ in (IV.3) and $Q_c^{\text{sym}}$ in (IV.4) respectively. For the sequences $x_n, n \geq 0$, in the PGDA, SPGDA, PAGDA1h and SPGDA1h and their limits $u$, define convergence errors $CE(n) = \log_{10} (x_n - u)_2$ and normalized residues $NR(n) = \log_{10} (1 - H_{0,m}^\text{split} x_n)_2, n \geq 0$, in the logarithmic scale, where $x_n = x_n/\|x_n\|_2$, $u = |u|/\|u\|_2$, and $\|x\|_2 = (\sum_{j \in V} |x(j)|^2)^{1/2}$ for $x = (x_j)_{j \in V}$. Shown in Figure 1 are the average of convergence errors $CE(n)$ and normalized residues $RE(n), n \geq 0$, over 500 trials. This demonstrates the exponential of the sequence $x_n, n \geq 0$, in the proposed distributed iterative algorithms to eigenvectors associated with eigenvalues 1 of lowpass spline filters, which is proved in Theorems II.1 and III.1

For a matrix $\Lambda$ on a graph $\mathcal{G} = (V, E)$, define its Schur norm by $\|A\|_S = \max_{e \in E} |P_A(i, i)|$ for $P_A(i, i), i \in V$, are given by (II.1) [3]. For the case that the constant $c$ in (II.4) and (III.4) is so chosen that $c \geq \|A\|_S$, the preconditioning matrices $Q_c$ and $Q_c^{\text{sym}}$ become a multiple of the identity matrix $I$ and the corresponding PGDA and SPGDA are the conventional gradient descent algorithm and its symmetric version respectively [9, 12, 14, 15, 31]. We denote the above algorithms with $c = \|A\|_S$ by GDASchur and SGDASchur respectively, see Figure 1 for their performance. Since 1 is the maximal eigenvalue of matrices $H_{0,m}^{\text{split}}, m \geq 1$, we can use the conventional power iteration method with entries of the initial $\mathbf{x}_0$ randomly selected in $[0, 1]$, POWER for abbreviation, to find principal eigenvectors [32]. Presented in Figure 1 is its performance. From Figure 1 we observe that the centralized algorithm POWER has fastest convergence to find eigenvectors of matrices $H_{0,m}^{\text{split}}, 2 \leq m \leq 4$, as followed are the distributed algorithm SPGDA, the centralized algorithm SPGDASchur and the distributed algorithm SPGDA1h, the next are the distributed algorithm PGDA and the centralized algorithm GDASchur, and the distributed algorithm PGDA1h has slowest convergence.

**APPENDIX A:** PROOF OF THEOREM II.1

By nonsingularity of the matrix $Q$, it suffices to prove

$$\|Q(x_n - u)\|_2 \leq \|Qx_0\|_2 r^n, \quad n \geq 0$$

(A.1)

for some $u$ satisfying $Au = 0$, where $r \in (0, 1)$.

Set $B = I - Q^{-1}A^*AQ^{-1}$ and let $u_i$ be orthonormal eigenvectors associated with eigenvalues $\gamma_i$ of the Hermitian matrix $B$ that satisfy

$$Bu_i = \gamma_i u_i, \quad 1 \leq i \leq N.$$  

(A.2)

Following the argument in [15] Theorem II.1 and applying (II.2), we obtain that $Q_2 - A^*A$ is positive semidefinite. This together with nonsingularity of the matrix $Q$ implies that

$$0 \leq \gamma_i \leq 1, \quad 1 \leq i \leq N.$$  

(A.3)

Write $Qx_0 = \sum_{i=1}^N \langle Qx_0, u_i \rangle u_i$, where $\langle \cdot, \cdot \rangle$ is the standard inner product on $\mathbb{C}^N$. By (II.1), we have that $Qx_n = BQx_{n-1}, n \geq 1$. Therefore

$$Qx_n = B^n Qx_0 = \sum_{i=1}^N \langle Qx_0, u_i \rangle u_i, \quad n \geq 0.$$  

(A.4)

Define $u = \sum_{i=1}^N \langle Qx_0, u_i \rangle Q^{-1}u_i$. Then by (A.3), (A.4) and the orthonormality of $u_i, 1 \leq i \leq N$, we obtain

$$\|Q(x_n - u)\|_2 = \left( \sum_{0 \leq \gamma_i < 1} \|\langle Qx_0, u_i \rangle\|_2^{2\gamma_i} \right)^{1/2} \leq r^n \|Qx_0 - Qu\|_2 \leq r^n \|Qx_0\|_2,$$

(A.5)

where $r = \max_{0 \leq \gamma_i < 1} \gamma_i$. This proves (A.1) and the desired exponential convergence of the sequence $x_n, n \geq 0$.

Taking the limit in (II.3) and applying the convergence in (A.1) yields $Q^{-2}A^*Au = 0$. This proves that $Au = 0$ and completes the proof.
