Distributed Estimation of the Laplacian Spectrum via Wave Equation and Distributed Optimization

Diego Deplano*, **Claudia Congiu***, **Alessandro Giua***, **Mauro Franceschelli***

*Department of Electrical and Electronic Engineering, University of Cagliari, Italy

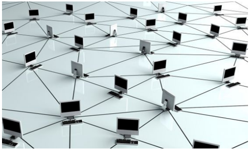
22nd IFAC World Congress, Yokohama, Japan, 9-14 July 2023.

Outline

1. Problem statement and motivation
2. Proposed protocol based on the Wave Equation and Distributed Optimization
3. Numerical simulations
4. Conclusions and future perspectives

Outline

Scenarios



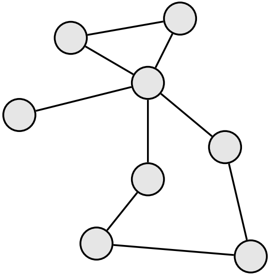
Peer-to-peer Networks

Multi-robot Systems

Smart Grids

Problem set-up

Undirected network → $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Set of agents → $\mathcal{V} = \{1, \ldots, n\}$
Set of interactions → $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$
State of agent i → $s_i \in \mathbb{R}^m$
Neighbors of agent i → $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$
Framework → Discrete-time $k \in \mathbb{N}$



$$s_i(k+1) = f_i\left(s_i(k), s_j(k) : j \in \mathcal{N}_i\right), \quad i \in \mathcal{V} \tag{1}$$

Objective

Make each agent $i \in \mathcal{V}$ estimate all the distinct eigenvalues $\lambda_1, \lambda_2, \ldots$ of the normalized[1] Laplacian matrix $L = \{\ell_{i,j}\}$ defined by

$$\ell_{i,j} = \begin{cases} 1 & \text{if } i = j \\ -1/|\mathcal{N}_i| & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}.$$

Problem set-up

Undirected network $\rightarrow \mathcal{G} = (\mathcal{V}, \mathcal{E})$

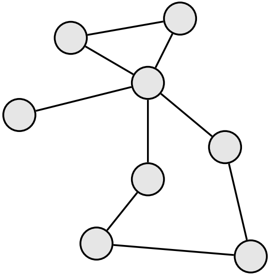
Set of agents $\rightarrow \mathcal{V} = \{1, \dots, n\}$

Set of interactions $\rightarrow \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$

State of agent i $\rightarrow s_i \in \mathbb{R}^m$

Neighbors of agent i $\rightarrow \mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$

Framework $\rightarrow$ Discrete-time $k \in \mathbb{N}$



$$s_i(k+1) = f_i\left(s_i(k), s_j(k) : j \in \mathcal{N}_i\right), \quad i \in \mathcal{V} \tag{1}$$

Objective

Make each agent $i \in \mathcal{V}$ estimate all the distinct eigenvalues $\lambda_1, \lambda_2, \dots$ of the normalized[1] Laplacian matrix $L = \{\ell_{i,j}\}$ defined by

$$\ell_{i,j} = \begin{cases} 1 & \text{if } i = j \\ -1/|\mathcal{N}_i| & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}.$$

Literature

Power Iteration methods:

- D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis", in *Journal of Computer and System Sciences* (2008).
- P. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks", in *Automatica* (2010).
- L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance", in *IEEE Transactions on Robotics* (2013).

Wave equation and Fourier Transform strategies:

- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized laplacian eigenvalues estimation for networked multi-agent systems", in "48th IEEE Conference on Decision and Control (CDC) held jointly with 28th IEEE Chinese Control Conference" (2009).
- T. Sahai, A. Speranzon, A. BanaszukT, "Hearing the clusters of a graph: A distributed algorithm", in *Automatica* (2012).
- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized estimation of Laplacian eigenvalues in multi-agent systems", in *Automatica* (2013).

Consensus-based strategies:

- A.Y. Kibangou and C. Commault, "Decentralized Laplacian eigenvalues estimation and collaborative network topology identification", in *IFAC Proceedings Volumes* (2012).
- T. Charalambous, M.G. Rabbat, M. Johansson, C.N. Hadjicostis, "Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum" in *IEEE Transactions on Control of Network Systems* (2015).

Distributed Optimization techniques:

- T.M.D. Tran and A.Y. Kibangou., "Distributed estimation of Laplacian eigenvalues via constrained consensus optimization problems", in "Systems & Control Letters" (2015).
- M. Zareh, L. Sabattini, and C. Secchi, "Distributed Laplacian eigenvalue and eigenvector estimation in multi-robot systems", in "Distributed autonomous robotic systems" (2018).

Literature

Power Iteration methods:

- D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis", in *Journal of Computer and System Sciences* (2008).
- P. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks", in *Automatica* (2010).
- L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance", in *IEEE Transactions on Robotics* (2013).

Wave equation and Fourier Transform strategies:

- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized laplacian eigenvalues estimation for networked multi-agent systems", in "48th IEEE Conference on Decision and Control (CDC) held jointly with 28th IEEE Chinese Control Conference" (2009).
- T. Sahai, A. Speranzon, A. BanaszukT, "Hearing the clusters of a graph: A distributed algorithm", in *Automatica* (2012).
- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized estimation of Laplacian eigenvalues in multi-agent systems", in *Automatica* (2013).

Consensus-based strategies:

- A.Y. Kibangou and C. Commault, "Decentralized Laplacian eigenvalues estimation and collaborative network topology identification", in *IFAC Proceedings Volumes* (2012).
- T. Charalambous, M.G. Rabbat, M. Johansson, C.N. Hadjicostis, "Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum" in *IEEE Transactions on Control of Network Systems* (2015).

Distributed Optimization techniques:

- T.M.D. Tran and A.Y. Kibangou., "Distributed estimation of Laplacian eigenvalues via constrained consensus optimization problems", in "Systems & Control Letters" (2015).
- M. Zareh, L. Sabattini, and C. Secchi, "Distributed Laplacian eigenvalue and eigenvector estimation in multi-robot systems", in "Distributed autonomous robotic systems" (2018).

Literature

Power Iteration methods:

- D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis", in *Journal of Computer and System Sciences* (2008).
- P. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks", in *Automatica* (2010).
- L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance", in *IEEE Transactions on Robotics* (2013).

Wave equation and Fourier Transform strategies:

- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized laplacian eigenvalues estimation for networked multi-agent systems", in "48th IEEE Conference on Decision and Control (CDC) held jointly with 28th IEEE Chinese Control Conference" (2009).
- T. Sahai, A. Speranzon, A. BanaszukT, "Hearing the clusters of a graph: A distributed algorithm", in *Automatica* (2012).
- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized estimation of Laplacian eigenvalues in multi-agent systems", in *Automatica* (2013).

Consensus-based strategies:

- A.Y. Kibangou and C. Commault, "Decentralized Laplacian eigenvalues estimation and collaborative network topology identification", in *IFAC Proceedings Volumes* (2012).
- T. Charalambous, M.G. Rabbat, M. Johansson, C.N. Hadjicostis, "Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum" in *IEEE Transactions on Control of Network Systems* (2015).

Distributed Optimization techniques:

- T.M.D. Tran and A.Y. Kibangou., "Distributed estimation of Laplacian eigenvalues via constrained consensus optimization problems", in "Systems & Control Letters" (2015).
- M. Zareh, L. Sabattini, and C. Secchi, "Distributed Laplacian eigenvalue and eigenvector estimation in multi-robot systems", in "Distributed autonomous robotic systems" (2018).

Literature

Power Iteration methods:

- D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis", in *Journal of Computer and System Sciences* (2008).
- P. Yang, R.A. Freeman, G.J. Gordon, K.M. Lynch, S.S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks", in *Automatica* (2010).
- L. Sabattini, C. Secchi, N. Chopra, A. Gasparri, "Distributed control of multirobot systems with global connectivity maintenance", in *IEEE Transactions on Robotics* (2013).

Wave equation and Fourier Transform strategies:

- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized laplacian eigenvalues estimation for networked multi-agent systems", in "48th IEEE Conference on Decision and Control (CDC) held jointly with 28th IEEE Chinese Control Conference" (2009).
- T. Sahai, A. Speranzon, A. BanaszukT, "Hearing the clusters of a graph: A distributed algorithm", in *Automatica* (2012).
- M. Franceschelli, A. Gasparri, A. Giua, C. Seatzu, "Decentralized estimation of Laplacian eigenvalues in multi-agent systems", in *Automatica* (2013).

Consensus-based strategies:

- A.Y. Kibangou and C. Commault, "Decentralized Laplacian eigenvalues estimation and collaborative network topology identification", in *IFAC Proceedings Volumes* (2012).
- T. Charalambous, M.G. Rabbat, M. Johansson, C.N. Hadjicostis, "Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum" in *IEEE Transactions on Control of Network Systems* (2015).

Distributed Optimization techniques:

- T.M.D. Tran and A.Y. Kibangou., "Distributed estimation of Laplacian eigenvalues via constrained consensus optimization problems", in "Systems & Control Letters" (2015).
- M. Zareh, L. Sabattini, and C. Secchi, "Distributed Laplacian eigenvalue and eigenvector estimation in multi-robot systems", in "Distributed autonomous robotic systems" (2018).

Main contribution

Motivation

In the current state-of-the-art methods, significant numerical errors occur in large networks due to the ill-conditioning of the problem.

Contribution

A novel protocol to compute the eigenvalues of the Laplacian matrix with the following features:

- High accuracy due to improved numerical conditioning of the problem (numerical simulations);
- Each agent of the system is able to estimate the entire Laplacian spectrum;
- The number of locally exchanged messages grows linearly with the size of the network;
- Scalability in large networks;

Main contribution

Motivation

In the current state-of-the-art methods, significant numerical errors occur in large networks due to the ill-conditioning of the problem.

Contribution

A novel protocol to compute the eigenvalues of the Laplacian matrix with the following features:

- High accuracy due to improved numerical conditioning of the problem (numerical simulations);
- Each agent of the system is able to estimate the entire Laplacian spectrum;
- The number of locally exchanged messages grows linearly with the size of the network;
- Scalability in large networks;

Outline

The proposed protocol: overview

The proposed protocol envisages three main steps to be performed by each agent in the network:

1. Update each state $x_i \in \mathbb{R}$ according to the discretized wave equation,

$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k) \tag{2}$$

and keep the memory of past iterations of the local state x_i.

2. Derive a data-driven model of the whole interconnected system by solving a distributed optimization problem

$$\begin{aligned} \boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n} \quad & \sum_{i \in \mathcal{V}} \|A_i \boldsymbol{\theta}_i - b_i\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad \forall (i,j) \in \mathcal{E} \end{aligned}, \tag{3}$$

where matrix A_i and vector b_i are constructed exploiting only the history of the local state x_i.

3. Compute the eigenvalues of L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$.

The proposed protocol: overview

The proposed protocol envisages three main steps to be performed by each agent in the network:

1. Update each state $x_i \in \mathbb{R}$ according to the discretized wave equation,

$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k) \tag{2}$$

   and keep the memory of past iterations of the local state x_i.

2. Derive a data-driven model of the whole interconnected system by solving a distributed optimization problem

$$\begin{aligned} \boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n} \quad & \sum_{i \in \mathcal{V}} \|A_i \boldsymbol{\theta}_i - b_i\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad \forall (i,j) \in \mathcal{E} \end{aligned}, \tag{3}$$

   where matrix A_i and vector b_i are constructed exploiting only the history of the local state x_i.

3. Compute the eigenvalues of L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$.

The proposed protocol: overview

The proposed protocol envisages three main steps to be performed by each agent in the network:

1. Update each state $x_i \in \mathbb{R}$ according to the discretized wave equation,

$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k) \tag{2}$$

   and keep the memory of past iterations of the local state x_i.

2. Derive a data-driven model of the whole interconnected system by solving a distributed optimization problem

$$\begin{aligned} \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_n}{\operatorname{argmin}} \quad & \sum_{i \in \mathcal{V}} \|A_i \boldsymbol{\theta}_i - b_i\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad \forall (i,j) \in \mathcal{E} \end{aligned}, \tag{3}$$

   where matrix A_i and vector b_i are constructed exploiting only the history of the local state x_i.

3. Compute the eigenvalues of L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$.

Step 1: Local state updates based on the discretized wave equation

The agents update their state according to the discretized wave equation

$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k), \qquad c^2 \in (0, 2].$$

The dynamics of the Multi-Agent System can be written in a compact form

$$\begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} = \underbrace{\begin{bmatrix} 2I_n - c^2 L & -I_n \\ I & 0_{n\times n} \end{bmatrix}}_{R} \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix}. \qquad (4)$$

Proposition 1

If the graph is connected, the eigenvalues $r_i \in \mathbb{C}$ of the transition matrix $R \in \mathbb{R}^{2n\times 2n}$ in eq. (4) are related to the eigenvalues $\lambda_i \in \mathbb{R}$ of the normalized Laplacian matrix $L \in \mathbb{R}^{n\times n}$ by

$$\Re\{r_i\} = \frac{2 - c^2\lambda_i}{2}, \quad \Im\{r_i\} = \pm\frac{c}{2}\sqrt{(4 - c^2\lambda_i)\lambda_i}.$$

Remark

Step 1: Local state updates based on the discretized wave equation

The agents update their state according to the discretized wave equation

$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k), \qquad c^2 \in (0,2].$$

The dynamics of the Multi-Agent System can be written in a compact form

$$\begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} = \underbrace{\begin{bmatrix} 2I_n - c^2 L & -I_n \\ I & 0_{n\times n} \end{bmatrix}}_{R} \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix}. \tag{4}$$

Proposition 1

If the graph is connected, the eigenvalues $r_i \in \mathbb{C}$ of the transition matrix $R \in \mathbb{R}^{2n\times 2n}$ in eq. (4) are related to the eigenvalues $\lambda_i \in \mathbb{R}$ of the normalized Laplacian matrix $L \in \mathbb{R}^{n\times n}$ by

$$\Re\{r_i\} = \frac{2 - c^2\lambda_i}{2}, \quad \Im\{r_i\} = \pm\frac{c}{2}\sqrt{(4 - c^2\lambda_i)\lambda_i}.$$

Remark

The problem of estimating the eigenvalues of L reduces to estimating the

Step 1: Local state updates based on the discretized wave equation

The agents update their state according to the discretized wave equation

$$x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_j(k), \qquad c^2 \in (0,2].$$

The dynamics of the Multi-Agent System can be written in a compact form

$$\begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} = \underbrace{\begin{bmatrix} 2I_n - c^2 L & -I_n \\ I & 0_{n\times n} \end{bmatrix}}_{R} \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix}. \tag{4}$$

Proposition 1

If the graph is connected, the eigenvalues $r_i \in \mathbb{C}$ of the transition matrix $R \in \mathbb{R}^{2n\times 2n}$ in eq. (4) are related to the eigenvalues $\lambda_i \in \mathbb{R}$ of the normalized Laplacian matrix $L \in \mathbb{R}^{n\times n}$ by

$$\Re\{r_i\} = \frac{2 - c^2\lambda_i}{2}, \quad \Im\{r_i\} = \pm\frac{c}{2}\sqrt{(4 - c^2\lambda_i)\lambda_i}.$$

Remark

The problem of estimating the eigenvalues of L reduces to estimating the

Step 2: Distributed Data-Driven Model Identification

An equivalent representation of the MAS is given by the following Auto-Regressive (AR) model

$$\begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} = R \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{x}(k+1) = \theta_m^* \boldsymbol{x}(k) + \theta_{m-1}^* \boldsymbol{x}(k-1) + \cdots + \theta_1^* \boldsymbol{x}(k-m+1), \tag{5}$$

for $k \geq m$, where $m \in \mathbb{N}$ denotes the number of past values.

Proposition 2

If $m \geq 2n$, the eigenvalues of the transition matrix $R \in \mathbb{R}^{2n \times 2n}$ are a subset of the roots of the polynomial with coefficients $\boldsymbol{\theta}^* = [\theta_1^*, \cdots, \theta_m^*]^\top$ in eq. (5).

Remark

The problem of estimating the eigenvalues of R reduces to estimating the coefficients $\boldsymbol{\theta}^*$.

Step 2: Distributed Data-Driven Model Identification

An equivalent representation of the MAS is given by the following Auto-Regressive (AR) model

$$\begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} = R \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{x}(k+1) = \theta_m^* \boldsymbol{x}(k) + \theta_{m-1}^* \boldsymbol{x}(k-1) + \cdots + \theta_1^* \boldsymbol{x}(k-m+1), \tag{5}$$

for $k \geq m$, where $m \in \mathbb{N}$ denotes the number of past values.

Proposition 2

If $m \geq 2n$, the eigenvalues of the transition matrix $R \in \mathbb{R}^{2n \times 2n}$ are a subset of the roots of the polynomial with coefficients $\boldsymbol{\theta}^* = [\theta_1^*, \cdots, \theta_m^*]^\top$ in eq. (5).

Remark

The problem of estimating the eigenvalues of R reduces to estimating the coefficients $\boldsymbol{\theta}^*$.

Step 2: Distributed Data-Driven Model Identification

An equivalent representation of the MAS is given by the following Auto-Regressive (AR) model

$$\begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} = R \begin{bmatrix} x(k) \\ x(k-1) \end{bmatrix} \quad \Leftrightarrow \quad \boldsymbol{x}(k+1) = \theta_m^* \boldsymbol{x}(k) + \theta_{m-1}^* \boldsymbol{x}(k-1) + \cdots + \theta_1^* \boldsymbol{x}(k-m+1), \tag{5}$$

for $k \geq m$, where $m \in \mathbb{N}$ denotes the number of past values.

Proposition 2

If $m \geq 2n$, the eigenvalues of the transition matrix $R \in \mathbb{R}^{2n \times 2n}$ are a subset of the roots of the polynomial with coefficients $\boldsymbol{\theta}^* = [\theta_1^*, \cdots, \theta_m^*]^\top$ in eq. (5).

Remark

The problem of estimating the eigenvalues of R reduces to estimating the coefficients $\boldsymbol{\theta}^*$.

Step 2: Distributed Data-Driven Model Identification

Lemma 1

If $m \geq 2n$, the vector of coefficients $\boldsymbol{\theta}^*$ is the solution of the optimization problem

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^m}{\operatorname{argmin}} \quad \| [X]_m^1 \boldsymbol{\theta} - [\boldsymbol{x}]_{2m}^{m+1} \|_2^2,$$

where

$$[X]_m^1 = \begin{bmatrix} \boldsymbol{x}(1) & \cdots & \boldsymbol{x}(m) \\ \vdots & . & \vdots \\ \boldsymbol{x}(m) & \cdots & \boldsymbol{x}(2m-1) \end{bmatrix}, \qquad [\boldsymbol{x}]_{2m}^{m+1} = \begin{bmatrix} \boldsymbol{x}(m+1) \\ \vdots \\ \boldsymbol{x}(2m) \end{bmatrix}.$$

Moreover, if the graph is connected, a distributed formulation of this optimization problem is given by

$$\begin{aligned} \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n \in \mathbb{R}^m}{\operatorname{argmin}} \quad & \sum_{i \in \mathcal{V}} \| [X_i]_m^1 \boldsymbol{\theta}_i - [\boldsymbol{x}_i]_{2m}^{m+1} \|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad \forall (i,j) \in \mathcal{E} \end{aligned}. \qquad (6)$$

Step 2: Data-driven model identification via distributed optimization

The agents can derive a data-driven model of the system by computing the coefficients $\boldsymbol{\theta}^*$ by solving the distributed optimization problem in eq. (6).

Lemma 2

The closed-form updates of the R-ADMM applied to the distributed optimization problem in eq. (6) over a connected graph $\mathcal{G}$ are given by,

$$\boldsymbol{\theta}_i(k+1) = N_i\Big(\boldsymbol{v}_i + \sum_{j\in\mathcal{N}_i} \boldsymbol{y}_{ij}(k)\Big)$$

$$\boldsymbol{y}_{ij}(k+1) = (1-\alpha)\boldsymbol{y}_{ij}(k) + \alpha\boldsymbol{p}_{ji}$$

where $\boldsymbol{p}_{ij} = -\boldsymbol{y}_{ij}(k) + 2\rho\boldsymbol{\theta}_i(k+1)$ are messages sent from agent i to agent j if $(i,j)\in\mathcal{E}$, $\boldsymbol{y}_{ij}$ are auxiliary local variables, $\alpha\in(0,1)$, $\rho>0$ are design parameters, and

$$N_i = (2[X_i]_m^1{}^\top [X_i]_m^1 + \rho|\mathcal{N}_i|I_m)^{-1},$$

$$\boldsymbol{v}_i = 2[X_i]_m^1{}^\top [\boldsymbol{x}_i]_{2m}^{m+1}.$$

[R0] N. Bastianello, R. Carli, L. Schenato, and M. Todescato. Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence. IEEE Transactions on Automatic Control, 66(6):2620–2635, 2020.

Step 2: Data-driven model identification via distributed optimization

The agents can derive a data-driven model of the system by computing the coefficients $\boldsymbol{\theta}^*$ by solving the distributed optimization problem in eq. (6).

Lemma 2

The closed-form updates of the R-ADMM applied to the distributed optimization problem in eq. (6) over a connected graph $\mathcal{G}$ are given by,

$$\boldsymbol{\theta}_i(k+1) = N_i\Big(\boldsymbol{v}_i + \sum_{j \in \mathcal{N}_i} \boldsymbol{y}_{ij}(k)\Big)$$

$$\boldsymbol{y}_{ij}(k+1) = (1-\alpha)\boldsymbol{y}_{ij}(k) + \alpha \boldsymbol{p}_{ji}$$

where $\boldsymbol{p}_{ij} = -\boldsymbol{y}_{ij}(k) + 2\rho\boldsymbol{\theta}_i(k+1)$ are messages sent from agent i to agent j if $(i,j) \in \mathcal{E}$, $\boldsymbol{y}_{ij}$ are auxiliary local variables, $\alpha \in (0,1)$, $\rho > 0$ are design parameters, and

$$N_i = (2[X_i]_m^1{}^\top [X_i]_m^1 + \rho|\mathcal{N}_i| I_m)^{-1},$$

$$\boldsymbol{v}_i = 2[X_i]_m^1{}^\top [\boldsymbol{x}_i]_{2m}^{m+1}.$$

[R0] N. Bastianello, R. Carli, L. Schenato, and M. Todescato. Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence. IEEE Transactions on Automatic Control, 66(6):2620–2635, 2020.

Step 3: Eigenvalues estimate procedure

The agents can now retrieve the eigenvalues of the normalized Laplacian matrix L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$,

$$r^m - \theta_m^* r^{m-1} - \theta_{m-1}^* r^{m-2} - \ldots - \theta_2^* r - \theta_1^*. \tag{7}$$

Lemma 3

A root $r \in \mathbb{C}$ of the monic polynomial in eq. (7) is an eigenvalue of the normalized Laplacian matrix L only if

$$\Re\{r\}^2 + \Im\{r\}^2 = 1 \tag{8}$$

Remarks

- Most likely, the condition in eq. (8) is not only necessary but also sufficient;
- In practice, condition in eq. (8) is verified up to a precision $\varepsilon > 0$.

Step 3: Eigenvalues estimate procedure

The agents can now retrieve the eigenvalues of the normalized Laplacian matrix L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$,

$$r^m - \theta_m^* r^{m-1} - \theta_{m-1}^* r^{m-2} - \ldots - \theta_2^* r - \theta_1^*. \quad (7)$$

Lemma 3

A root $r \in \mathbb{C}$ of the monic polynomial in eq. (7) is an eigenvalue of the normalized Laplacian matrix L only if

$$\Re\{r\}^2 + \Im\{r\}^2 = 1 \quad (8)$$

Remarks

- Most likely, the condition in eq. (8) is not only necessary but also sufficient;
- In practice, condition in eq. (8) is verified up to a precision $\varepsilon > 0$.

Step 3: Eigenvalues estimate procedure

The agents can now retrieve the eigenvalues of the normalized Laplacian matrix L from the roots of the monic polynomial with coefficients $\boldsymbol{\theta}^*$,

$$r^m - \theta_m^* r^{m-1} - \theta_{m-1}^* r^{m-2} - \ldots - \theta_2^* r - \theta_1^*. \tag{7}$$

Lemma 3

A root $r \in \mathbb{C}$ of the monic polynomial in eq. (7) is an eigenvalue of the normalized Laplacian matrix L only if

$$\mathfrak{R}\{r\}^2 + \mathfrak{I}\{r\}^2 = 1 \tag{8}$$

Remarks

- Most likely, the condition in eq. (8) is not only necessary but also sufficient;
- In practice, condition in eq. (8) is verified up to a precision $\varepsilon > 0$.

The proposed protocol: detailed description

Algorithm 1: Distributed Estimation of Distinct Eigenvalues of the Laplacian Matrix

Input: Order $m \in \mathbb{N}_+$, wave speed $c \in (0, \sqrt{2})$,
design parameters $\alpha \in (0,1)$, $\rho > 0$, $\varepsilon > 0$
Init.: $x_i(1) = x_i(0) \in \mathbb{R}$ for all $i \in \mathcal{V}$
$\boldsymbol{y}_{ij}(2m) \in \mathbb{R}^m$ for all $(i,j) \in \mathcal{E}$
Output: Estimated eigenvalues $\hat{\lambda}_j$
for $k = 1, 2, 3, \ldots$ **each node** $i \in \mathcal{V}$ **does**
  if $k \leq 2m - 1$ **then** // Step 1
    gather $x_j(k)$ from each neighbor $j \in \mathcal{N}_i$
    $x_i(k+1) = 2x_i(k) - x_i(k-1) - c^2 \sum_{j \in \mathcal{N}_i} \ell_{ij} x_i(k)$
    send $x_i(k+1)$ to each neighbor $j \in \mathcal{N}_i$
    if $k = 2m - 1$ **then**
      $N_i = (2[X_i]_m^1{}^\top [X_i]_m^1 + \rho|\mathcal{N}_i| I_m)^{-1}$,
      $\boldsymbol{v}_i = 2[X_i]_m^1{}^\top [\boldsymbol{x}_i]_{2m}^{m+1}$
  else // Step 2
    $\boldsymbol{\theta}_i(k+1) = N_i\left(\boldsymbol{v}_i + \sum_{j \in \mathcal{N}_i} \boldsymbol{y}_{ij}(k)\right)$
    $\boldsymbol{p}_{ij} = -\boldsymbol{y}_{ij}(k) + 2\rho\boldsymbol{\theta}_i(k+1)$
    send $\boldsymbol{p}_{ij}$ to each neighbor $j \in \mathcal{N}_i$
    gather $\boldsymbol{p}_{ji}$ from each neighbor $j \in \mathcal{N}_i$
    $\boldsymbol{y}_{ij}(k+1) = (1-\alpha)\boldsymbol{y}_{ij}(k) + \alpha \boldsymbol{p}_{ji}$
    compute the roots of $r^m = \sum_{j=0}^{m-1} \theta_{i,j}(k+1) r^j$
    for each complex root $r_j \in \mathbb{C}$ // Step 3
      if $|\Re\{r_j\}^2 + \Im\{r_j\}^2 - 1| < \varepsilon$ **then**
        output $\hat{\lambda}_j = 2(1 - \Re\{r_j\})/c^2$

Wave Equation → (Step 1)
Distributed Optimization → (Step 2)
Eigenvalue Estimation → (Step 3)

Main results

Theorem

Consider a MAS with n agents interacting according to graph $\mathcal{G}$ and executing Algorithm 1. If:

- the graph $\mathcal{G}$ is undirected and connected;
- the order of the model satisfies $m \geq 2n$;

then each agent asymptotically estimates all distinct eigenvalues of the Laplacian matrix L for almost every initial condition.

Proof sketch:

- The agents generate and store non vanishing/diverging trajectories by executing the discretized wave equation (Proposition 1);
- The agents derive a data-driven model of the system θ^* from the stored data by solving a distributed optimization problem (Lemma 1 and Lemma 2);
- The agents compute the eigenvalues of the normalized Laplacian matrix by selecting the roots of the monic polynomial defined by the coefficients θ^* with unitary magnitude (Proposition 2 and Lemma 3).

Main results

Theorem

Consider a MAS with n agents interacting according to graph $\mathcal{G}$ and executing Algorithm 1. If:

- the graph $\mathcal{G}$ is undirected and connected;
- the order of the model satisfies $m \geq 2n$;

then each agent asymptotically estimates all distinct eigenvalues of the Laplacian matrix L for almost every initial condition.

Proof sketch:

- The agents generate and store non vanishing/diverging trajectories by executing the discretized wave equation (Proposition 1);
- The agents derive a data-driven model of the system $\boldsymbol{\theta}^*$ from the stored data by solving a distributed optimization problem (Lemma 1 and Lemma 2);
- The agents compute the eigenvalues of the normalized Laplacian matrix by selecting the roots of the monic polynomial defined by the coefficients $\boldsymbol{\theta}^*$ with unitary magnitude (Proposition 2 and Lemma 3).

Main results

Theorem

Consider a MAS with n agents interacting according to graph $\mathcal{G}$ and executing Algorithm 1. If:

- the graph $\mathcal{G}$ is undirected and connected;
- the order of the model satisfies $m \geq 2n$;

then each agent asymptotically estimates all distinct eigenvalues of the Laplacian matrix L for almost every initial condition.

Proof sketch:

- The agents generate and store non vanishing/diverging trajectories by executing the discretized wave equation (Proposition 1);
- The agents derive a data-driven model of the system $\boldsymbol{\theta}^*$ from the stored data by solving a distributed optimization problem (Lemma 1 and Lemma 2);
- The agents compute the eigenvalues of the normalized Laplacian matrix by selecting the roots of the monic polynomial defined by the coefficients $\boldsymbol{\theta}^*$ with unitary magnitude (Proposition 2 and Lemma 3).

Main results

Theorem

Consider a MAS with n agents interacting according to graph $\mathcal{G}$ and executing Algorithm 1. If:

- the graph $\mathcal{G}$ is undirected and connected;
- the order of the model satisfies $m \geq 2n$;

then each agent asymptotically estimates all distinct eigenvalues of the Laplacian matrix L for almost every initial condition.

Proof sketch:

- The agents generate and store non vanishing/diverging trajectories by executing the discretized wave equation (Proposition 1);
- The agents derive a data-driven model of the system $\boldsymbol{\theta}^*$ from the stored data by solving a distributed optimization problem (Lemma 1 and Lemma 2);
- The agents compute the eigenvalues of the normalized Laplacian matrix by selecting the roots of the monic polynomial defined by the coefficients $\boldsymbol{\theta}^*$ with unitary magnitude (Proposition 2 and Lemma 3).

Outline

1. Problem statement and motivation
2. Proposed protocol based on the Wave Equation and Distributed Optimization
3. Numerical simulations
4. Conclusions and future perspectives

Example 1: Estimation of unobservable eigenvalues

Consider a network of $n = 7$ agents interacting according to a line graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that the eigenvalues of the normalized Laplacian matrix L are

$$\lambda_1 = 0, \quad \lambda_2 = 0.134, \quad \lambda_3 = 0.5, \quad \lambda_4 = 1, \quad \lambda_5 = 1.5, \quad \lambda_6 = 1.866, \quad \lambda_7 = 2.$$

The system is fully observable from the agents at the periphery of the network (agents 1 and 7) but it is only partially observable from the central agent $i^* = 4$.

By executing the proposed protocol with the following design parameters

$$c = \sqrt{2}, \ \alpha = 0.99, \ \rho = 10, \ m = 2n, \ \varepsilon = 10^{-6}, \tag{9}$$

the agents asymptotically agree upon the vector of coefficients

$$\boldsymbol{\theta}^* = [\ 340, -359, 376, 918, 24.3, 24.6, 24.9, 25.1, 25.1, 25.0, 24.8, 24.5, 296, 276, 255\ ] \cdot 10^{-}$$

whose roots are

$$\begin{aligned}
&r_1 = +1.000 \pm j0.000, && |r_1| \approx 1, && r_2 = +0.867 \pm j0.499, && |r_2| \approx 1,\\
&r_3 = +0.503 \pm j0.865, && |r_3| \approx 1, && r_4 = +0.005 \pm j1.000, && |r_4| \approx 1,\\
&r_5 = -0.492 \pm j0.870, && |r_5| \approx 1, && r_6 = -0.857 \pm j0.516, && |r_6| \approx 1,\\
&r_7 = -0.990 \pm j0.141, && |r_7| \approx 1, && r_8 = +0.338 \pm j0.475, && |r_8| \approx 0.5830.
\end{aligned}$$

The root r_8 can be discarded since its modulus is not 1, estimating correctly all the eigenvalues, up to an error of $\varepsilon = 10^{-6}$.

Example 1: Estimation of unobservable eigenvalues

Consider a network of $n = 7$ agents interacting according to a line graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, such that the eigenvalues of the normalized Laplacian matrix L are

$$\lambda_1 = 0, \quad \lambda_2 = 0.134, \quad \lambda_3 = 0.5, \quad \lambda_4 = 1, \quad \lambda_5 = 1.5, \quad \lambda_6 = 1.866, \quad \lambda_7 = 2.$$

The system is fully observable from the agents at the periphery of the network (agents 1 and 7) but it is only partially observable from the central agent $i^* = 4$.
By executing the proposed protocol with the following design parameters

$$c = \sqrt{2}, \;\; \alpha = 0.99, \;\; \rho = 10, \;\; m = 2n, \;\; \varepsilon = 10^{-6}, \tag{9}$$

the agents asymptotically agree upon the vector of coefficients

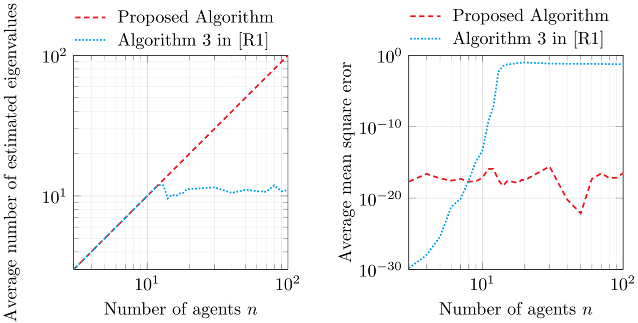
$$\boldsymbol{\theta}^* = [\ 340, -359, 376, 918, 24.3, 24.6, 24.9, 25.1, 25.1, 25.0, 24.8, 24.5, 296, 276, 255\ ] \cdot 10^{-}$$

whose roots are

$$\begin{aligned}
&r_1 = +1.000 \pm j0.000, && |r_1| \approx 1, && r_2 = +0.867 \pm j0.499, && |r_2| \approx 1,\\
&r_3 = +0.503 \pm j0.865, && |r_3| \approx 1, && r_4 = +0.005 \pm j1.000, && |r_4| \approx 1,\\
&r_5 = -0.492 \pm j0.870, && |r_5| \approx 1, && r_6 = -0.857 \pm j0.516, && |r_6| \approx 1,\\
&r_7 = -0.990 \pm j0.141, && |r_7| \approx 1, && r_8 = +0.338 \pm j0.475, && |r_8| \approx 0.5830.
\end{aligned}$$

The root r_8 can be discarded since its modulus is not 1, estimating correctly all the eigenvalues, up to an error of $\varepsilon = 10^{-6}$.

Example 2: Scalability for large-networks



[R1] T. Charalambous, et al. "Distributed finite-time computation of digraph parameters: Left-eigenvector, out-degree and spectrum." in IEEE Transactions on Control of Network Systems 3.2 (2015).

Outline

Conclusions and future directions

Contributions:

- A novel distributed algorithm for Laplacian eigenvalues estimation;
- High accuracy tanks to better numerical stability with respect to the state-of-the-art;
- The memory burden grows linearly with the size of the network.

Work in progress:

- Formal characterization of the condition number of the problem and comparison with the literature;
- Development of an online version of the algorithm to estimate and track time-varying eigenvalues;
- Formulation of the problem in the context of open multi-agent systems.

Conclusions and future directions

Contributions:

- A novel distributed algorithm for Laplacian eigenvalues estimation;
- High accuracy tanks to better numerical stability with respect to the state-of-the-art;
- The memory burden grows linearly with the size of the network.

Work in progress:

- Formal characterization of the condition number of the problem and comparison with the literature;
- Development of an online version of the algorithm to estimate and track time-varying eigenvalues;
- Formulation of the problem in the context of open multi-agent systems.

Conclusions and future directions

Contributions:

- A novel distributed algorithm for Laplacian eigenvalues estimation;
- High accuracy tanks to better numerical stability with respect to the state-of-the-art;
- The memory burden grows linearly with the size of the network.

Work in progress:

- Formal characterization of the condition number of the problem and comparison with the literature;
- Development of an online version of the algorithm to estimate and track time-varying eigenvalues;
- Formulation of the problem in the context of open multi-agent systems.

Conclusions and future directions

Contributions:

- A novel distributed algorithm for Laplacian eigenvalues estimation;
- High accuracy tanks to better numerical stability with respect to the state-of-the-art;
- The memory burden grows linearly with the size of the network.

Work in progress:

- Formal characterization of the condition number of the problem and comparison with the literature;
- Development of an online version of the algorithm to estimate and track time-varying eigenvalues;
- Formulation of the problem in the context of open multi-agent systems.

Conclusions and future directions

Contributions:

- A novel distributed algorithm for Laplacian eigenvalues estimation;
- High accuracy tanks to better numerical stability with respect to the state-of-the-art;
- The memory burden grows linearly with the size of the network.

Work in progress:

- Formal characterization of the condition number of the problem and comparison with the literature;
- Development of an online version of the algorithm to estimate and track time-varying eigenvalues;
- Formulation of the problem in the context of open multi-agent systems.

Conclusions and future directions

Contributions:

- A novel distributed algorithm for Laplacian eigenvalues estimation;
- High accuracy tanks to better numerical stability with respect to the state-of-the-art;
- The memory burden grows linearly with the size of the network.

Work in progress:

- Formal characterization of the condition number of the problem and comparison with the literature;
- Development of an online version of the algorithm to estimate and track time-varying eigenvalues;
- Formulation of the problem in the context of open multi-agent systems.

Distributed Estimation of the Laplacian Spectrum via Wave Equation and Distributed Optimization

Thank you for your attention!

Diego Deplano

Email: diego.deplano@unica.it
Webpage: www.sites.google.com/view/deplanodiego/