Distributed clustering algorithm for adaptive pandemic control

XABIER INSAUSTI, MARTA ZÁRRAGA-RODRÍGUEZ, CAROLINA NOLASCO-FERENCIKOVA, AND JESÚS GUTIÉRREZ-GUTIÉRREZ (Senior Member, IEEE)

Tecnun, University of Navarra, Manuel Lardizábal 13, 20018 San Sebastián, Spain (e-mail: xinsausti@tecnun.es, mzarraga@tecnun.es, cnolasco@alumni.unav.es, jgutierrez@tecnun.es)

Corresponding author: Xabier Insausti (e-mail: xinsausti@tecnun.es).

This work was supported in part by the Spanish Ministry of Science and Innovation through the ADELE project (PID2019-104958RB-C44).

ABSTRACT The COVID-19 pandemic has had severe consequences on the global economy, mainly due to indiscriminate geographical lockdowns. Moreover, the digital tracking tools developed to survey the spread of the virus have generated serious privacy concerns. In this paper, we present an algorithm that adaptively groups individuals according to their social contacts and their risk level of severe illness from COVID-19, instead of geographical criteria. The algorithm is fully distributed and therefore, individuals do not know any information about the group they belong to. Thus, we present a distributed clustering algorithm for adaptive pandemic control.

INDEX TERMS Adaptive algorithm, Clustering algorithms, COVID-19, Distributed algorithms

I. INTRODUCTION

COVID-19 [1] is a disease caused by the new coronavirus SARS-CoV-2. It was declared a pandemic by the World Health Organization (WHO) in March 2020. First cases were reported in Wuhan, People’s Republic of China, to the WHO on December 31st 2019. Since then, 117.7 billion cases have been reported, with more than 2.6 billion deaths, as of March 10th, 2021 [2]. Those at a higher risk of severe illness from COVID-19 include those aged 60 or over, or with underlying medical problems such as diabetes, cancer, or high-blood pressure. Nevertheless, this highly infectious disease can affect anyone, and can become deadly at any age. Personal health precautions are strongly advised, mainly wearing a mask, physical distancing and handwashing [1].

In response to the pandemic, governments all over the world have implemented non-pharmaceutical measures in order to stop the spread of the virus, or flatten the curve. Social distancing interventions, such as isolation and quarantine of infected patients and their contacts, external and internal border restrictions, workplace distancing, closure of schools, and complete quarantine or lockdown have been the most common [3], [4]. FluTE, a stochastic influenza pandemic simulation model [5], was used to assess the potential effect of different social distancing interventions using Singapore as a study case [6], since it was among the first to report infections. The model predicted quarantine or lockdown to be the most effective measures, particularly combined with school closures and workplace distancing. In fact, Singapore successfully implemented these measures, preventing community spread [6]. It is important to point out that these measures are targeted geographically [7]. This geographical approach affects large population groups, regardless of their economic sector or activity. Therefore, these measures have severe consequences on the regional, national and global economy: they pose a risk of reduced income or even job loss, affecting the most disadvantaged populations [8], and results show an average 2.5-3% global GDP drop per month of complete lockdown [9]. This shows that, despite lockdown and quarantine being the most effective measures, a different non-geographical approach should be taken in order to overcome the aforementioned negative impacts. Furthermore, these measures are most efficient when applied to individuals that belong to groups where transmission is most likely to occur [10]. Hence, individuals should be grouped according to their social contacts, which might not necessarily coincide with geographical areas. However, if the criteria are not geographical, it is more difficult for individuals to know which group they belong to. Furthermore, such groups may change with time and adaptive grouping strategies are needed.

Public health experts across institutions and countries have identified digital tracking measures as useful tools to survey and slow down the spread of the virus. Numerous technolo-
gies have been developed with this purpose, such as digital health certificates, which assign a color-coded COVID status to their users, physical surveillance initiatives [11], symptom checkers, or flow modelling tools, which quantify and track people’s movements in specified geographical regions [12]. These technologies, however, raise severe ethical concerns about putting user’s privacy and security at risk. For instance, out of the 65 digital health certificate applications that are currently in operation globally, 82% are considered to have inadequate privacy policies [11].

One of the most common examples of digital tracking measures are proximity or contact tracing tools, mainly via mobile applications. In particular, studies have predicted them to be beneficial in mitigating the spread of the virus, specifically during the de-escalation of physical distancing [13]. There are over 120 contact tracing applications currently available in over 70 countries [11]. These contact tracing tools gather data from their users, such as their location, their health records or contact information. This has raised ethical concerns surrounding the privacy of users and their data.

For instance, one of the earlier contact tracing tools developed was Singapore’s TraceTogether [14], a mobile application which operates via Bluetooth connection. Nearby phones, with Bluetooth and TraceTogether open in the background, exchange tokens, which are stored encrypted in each phone and in a central server [15]. If a user tests positive for COVID-19, contact tracers can easily use the tokens to identify those at high risk of infection. TraceTogether does not gather more than the necessary information, only the users’ contact/mobile number, identification details and random ID. The tokens sent via Bluetooth are time-varying random strings, and this way, privacy between users is kept. However, when a user is infected, the government can retrieve all mobile numbers of the individuals the infected user has been in contact with [15]. Having this centralized approach leaves no privacy for users from authorities.

For overcoming the privacy concerns of a centralized approach, in an unprecedented joint effort Apple and Google developed a contact tracing platform based on Bluetooth [16]. Specifically, they developed an application programming interface (API) that allows interoperability between Android and iOS. This API requires contact tracing applications to take on a decentralized approach. The contact matching analysis is performed at a local level, which also protects users’ privacy, maintaining their anonymity. Over 37% of contact tracing applications now use Apple and Google’s API [11].

In this paper, we propose a distributed algorithm that adaptively groups individuals (i.e., creates clusters) according to their social contacts and their risk level of severe illness from COVID-19. This will be modelled as a doubly-weighted undirected graph. Moreover, by combining our algorithm with a distributed consensus algorithm, each individual can know the epidemiological situation of the group they belong to and can take the social distancing measures that correspond to the epidemiological situation of their group.

There exist many algorithms to create clusters and, in particular, many works about privacy-preserving clustering have been conducted (see, e.g., [17]–[21]). These works are based on statistical or cryptography techniques to protect data. Our algorithm can use some of the abovementioned techniques for becoming privacy-preserving between nearby users, but since it is fully distributed individuals do not share any information about the cluster they belong to even if no cryptographic methods are used. Therefore, privacy from authorities is kept. That is, only the individuals themselves know which group (cluster) they belong to without having knowledge of the rest of the members of the group.

In the literature, many works deal with distributed clustering of data using a wide variety of techniques and applying the results to different fields (see, e.g., [22]–[29]). In this paper, we focus on spectral clustering techniques because they are easy to implement and have been shown to be more effective in finding clusters than some traditional algorithms such as k-means [30]. Among the previously cited works, [27]–[29] present a similar approach to the one considered in this paper. Specifically, in [27] the authors propose a distributed spectral clustering algorithm but they do not consider weights neither in the nodes nor in the edges. In [28], the authors propose a distributed spectral clustering algorithm but they only consider an edge-weighted graph. Finally, in [29] a spectral clustering for doubly-weighted graphs is proposed but, unlike here, the algorithm is not distributed.

The remainder of this paper is organized as follows. Section II states preliminary considerations regarding distributed computation and spectral clustering. Section III presents the distributed clustering algorithm for adaptive pandemic control, its convergence speed, and its computational complexity. Finally, two illustrative examples and some conclusions are given in Sections IV and V, respectively.

II. PRELIMINARIES

A. DISTRIBUTED COMPUTATION USING A LINEAR ITERATIVE ALGORITHM

Consider a network composed of $n$ nodes, where each node represents the mobile phone (or similar) device of one person. The entire population and the interactions among them can be viewed as an undirected graph $G = (V, E)$ with no loops, where $V = \{1, 2, \ldots, n\}$ is the set of nodes (vertices) and $E$ is the set of edges. If two nodes $i, j \in V$ interact between them, then $\{i, j\} \in E$. We say that these nodes are connected, and can therefore interchange information. Conversely, if $\{i, j\} \notin E$, this means that nodes $i, j \in V$ are not connected and cannot interchange information.

We assume that each node $i \in V$ has an initial value $x_i(0) \in \mathbb{R}$, where $\mathbb{R}$ denotes the set of real numbers. In distributed computation each node computes its target value by exchanging information with its neighbouring nodes.

The approach that will be considered here for distributed
computation is to use a linear iterative algorithm of the form
\[
x_i(t+1) = w_i x_i(t) + \sum_{j \in V: (i,j) \in E} w_{i,j} x_j(t), \quad i \in V,
\]
where \(w_{i,j} \in \mathbb{R}\) and time \(t \in \{0, 1, 2, \ldots\}\) is assumed to be discrete (see [31]). Let \(x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^\top\) be the column vector with the values of the nodes at time instant \(t\), where \(\top\) denotes transpose. The linear iterative algorithm (1) can then be written as
\[
x(t+1) = Wx(t) = W^{t+1}x(0),
\]
where \(W\) is the \(n \times n\) matrix defined as
\[
(W)_{i,j} = \begin{cases} 
  0 & \text{if } i \neq j \text{ and } \{i, j\} \notin E, \\
  w_{i,j} & \text{otherwise,}
\end{cases}
\]
for \(i, j \in V\).

### B. SPECTRAL CLUSTERING

Clustering a graph consists in separating the nodes of the graph into disjoint groups (clusters). There exist many algorithms for graph clustering. The approach that will be presented in this section is based on the notion of eigenvectors of the Laplacian matrix of the graph [35], mainly by an eigenvector corresponding to the smallest nonzero eigenvalue of such matrix, known as Fiedler vector [36].

In this paper \(G\) is assumed to be a doubly-weighted graph, that is, a graph with weights both in the nodes and in the edges. We denote with \(\delta_i > 0\) the weight of node \(i\), for \(i \in V\), and whenever \(\{i, j\} \in E\) we denote with \(\sigma_{i,j} > 0\) the weight of such edge.

In [29, Lemma 1], in the context of doubly-weighted graphs, the notion of weighted Laplacian matrix was presented. The weighted Laplacian matrix of the graph is the \(n \times n\) matrix given by
\[
L = \Lambda^{-\frac{1}{2}}(D - \Sigma) \Lambda^{-\frac{1}{2}},
\]
where \(\Lambda^{-\frac{1}{2}}\) is the \(n \times n\) diagonal matrix with \(\Lambda^{-\frac{1}{2}}\) as in (4). Then, for almost every column vector \(x(0)\),
\[
\lim_{t \to \infty} [x(t) - x(t-1)]_i \left(\begin{array}{c} [x(t-1) - x(t-2)]_i \\ [x(t) - x(t-1)]_i \end{array}\right) = C(u_{k+1})_i
\]
for all \(i \in V\), where
\[
x(t+1) = \left(I_n - \frac{1}{\lambda_n} L\right)x(t) \quad \forall t \in \{0, 1, 2, \ldots\},
\]
\(C\) is a non-zero constant, and \(I_n\) denotes the \(n \times n\) identity matrix.

**Proof:** See Appendix A.

In the rare case in which \(\lambda_{k+1} = \lambda_{k+2}\) the Fiedler vector would not be unique, meaning that it might be any vector in a subspace of dimension larger than one. In this rare case, Algorithm 1 would still work because it would converge to one of such vectors.

Observe that the iterative equation (6) can be computed in a distributed way since it is of the form of (2), and \(\left(I_n - \frac{1}{\lambda_n} L\right)\) satisfies (3). Therefore, from (5) each node \(i \in V\) can know the \(i\)-th entry of an eigenvector associated to \(\lambda_{k+1}\). However, in order to compute (6) in a distributed way, each node needs to know \(\lambda_n\). Lemma 1 shows that \(\lambda_n\) can also be computed in a distributed way.

**Lemma 1:** Consider a doubly-weighted undirected graph \(G\) with no loops, \(n\) nodes, and \(k\) components. Let the Laplacian matrix \(L\) of the graph \(G\) be as in (4). Then, for almost every real \(n\)-dimensional column vector \(y(0)\),
\[
\lim_{t \to \infty} \frac{y(t)}{y(t-1)} = \lambda_n \quad \forall i \in V,
\]
where

\[ y(t + 1) = Ly(t) \quad \forall t \in \{0, 1, 2, \ldots \}. \]  \hspace{1cm} (8)

**Proof:** See [38, Section 5.8.1] or [39, Section 9.3].

Observe that the iterative equation (8) can be computed in a distributed way since it is of the form of (2), and \( L \) satisfies (3). Therefore, from (7) each node \( i \in \mathcal{V} \) can know \( \lambda_n \).

It should be mentioned that the distributed computation of \( u_{k+1} \) can be found in [28], but only for an edge-weighted graph, that is, for the particular case in which \( \delta_i = 1 \) for all \( i \in \mathcal{V} \).

We finish this section by describing Algorithm 1. For ease of notation, we define

\[ f(x, t) := W^t x(0), \]

which is the \( t \)-th iteration of (1) and can clearly be computed in a distributed way. As for Algorithm 1, we fix \( t_0 \) to be the number of iterations of (1) required for a desired precision.

Table 1 describes Algorithm 1 and relates it with the theoretical aspects shown in this section. Observe that Algorithm 1 separates the nodes of the graph into two clusters. However, if the algorithm is used recursively within each cluster, we can separate the nodes of the original graph into as many clusters as desired.

**TABLE 1: Explanation of Algorithm 1**

| Lines | Description |
|-------|-------------|
| 1-7   | In (2), set \( W \) as \( L \) to compute (8) |
| 10-12 | Computation of \( \lambda_n \) according to Lemma 1 |
| 13-17 | In (2), set \( W \) as \( \left(I_n - \frac{1}{\lambda_n} L\right) \) to compute (6) |
| 20-23 | Computation of the \( i \)-th entry of an eigenvector associated to \( \lambda_2 \) according to Theorem 1 |
| 24-26 | Assign node \( i \) to a cluster depending on the sign of \([u_{k+1}]_i\) |

**B. CONVERGENCE SPEED**

In this subsection we study the convergence speed of the proposed algorithm. Specifically, we show that the convergence of the sequences considered in Theorem 1 and Lemma 1 is linear. We recall that the convergence of a sequence \( a_0, a_1, a_2, \ldots \), which converges to \( \ell \), is said to be linear if the limit

\[ \lim_{t \to \infty} \frac{|a_{t+1} - \ell|}{|a_t - \ell|} \]

is a nonzero constant (see [38, p. 224]).

The following theorem deals with the convergence speed of the sequence considered in Theorem 1.

**Theorem 2:** Let \( x(t) \) be as in Theorem 1. Then, the convergence of the sequence

\[ |x(t) - x(t-1)|_i \left( \frac{|x(t-1) - x(t-2)|_i}{|x(t) - x(t-1)|_i} \right)^{t-1} \]

is linear for all \( i \in \mathcal{V} \).

---

**Algorithm 1** Distributed clustering algorithm for adaptive pandemic control

1. for all nodes \( i \in \mathcal{V} \) do
2. \( s_i \leftarrow 0 \)
3. for all nodes \( j \) connected to \( i \) do
4. \( w_{i,j} \leftarrow \frac{-\sigma_{i,j}}{\sqrt{\delta_i \delta_j}} \)
5. \( s_i \leftarrow s_i + \sigma_{i,j} \)
6. end for
7. \( w_{i,i} \leftarrow \frac{2s_i}{\delta_i} \)
8. \( |y(0)|_i \leftarrow \text{rand()} \) \hspace{1cm} ▶ An arbitrary value
9. end for
10. for all nodes \( i \in \mathcal{V} \) do
11. \( \lambda_n \leftarrow \frac{|f(x, t_0)|_i}{|f(y, t_0 - 1)|_i} \)
12. end for
13. for all nodes \( i \in \mathcal{V} \) do
14. for all nodes \( j \) connected to \( i \) do
15. \( w_{i,j} \leftarrow \frac{-w_{i,j}}{\lambda_n} \)
16. end for
17. \( w_{i,i} \leftarrow 1 - \frac{w_{i,i}}{\lambda_n} \)
18. \( |x(0)|_i \leftarrow \text{rand()} \) \hspace{1cm} ▶ An arbitrary value
19. end for
20. for all nodes \( i \in \mathcal{V} \) do
21. \( \beta_i \leftarrow |f(x, t_0)|_i - |f(x, t_0 - 1)|_i \)
22. \( \gamma_i \leftarrow |f(x, t_0 - 1)|_i - |f(x, t_0 - 2)|_i \)
23. \( |Cu_{k+1}|_i \leftarrow \beta_i \left( \frac{\beta_i}{\gamma_i} \right)^{t_0 - 1} \)
24. if \([Cu_{k+1}]_i > 0\) then node \( i \) belongs to cluster 1
25. else node \( i \) belongs to cluster 2
26. end if
27. end for

---
assume that all the social contacts have equal time frames. It is important to remark that Algorithm 1 works.

In this section we present two examples to illustrate how our algorithm depend on the risk of severe illness but also on the social interaction among individuals.

C. COMPUTATIONAL COMPLEXITY

The computational bottleneck in spectral clustering is the computation of the eigenvectors of the Laplacian matrix. To speed up the computation of such eigenvectors, the power iteration method is usually used [40].

In this subsection we study the computational complexity of Algorithm 1 for each node. The computational complexity of Algorithm 1 is essentially determined by the complexity of running twice the power iteration method. In particular, the power iteration method is used to compute the largest eigenvalue of \( L \) (see line 11 of Algorithm 1) and to compute an eigenvector associated to the largest eigenvalue less than one of \( I_n - \frac{1}{\lambda_n} L \) (see lines 21-22 of Algorithm 1). The power iteration method is computationally expensive for large matrices but \( L \) and \( I_n - \frac{1}{\lambda_n} L \) are sparse matrices with only a few non-zero entries. This reduces the computational difficulties, as subsequently explained.

Let \( c_i \) be the number of contacts the \( i \)-th individual has. It is important to remark that \( c_i \) does not depend on \( n \). Consequently, regardless of the value of \( n \), the \( i \)-th row of \( L \) will have at most \( c_i + 1 \) non-zero entries. Therefore, the computation of \( [f(y, t_0)]_i \), needed in line 11 requires no more than \( t_0(c_i + 1) \) multiplications (see Equation (1)). Similarly, the computation of \( [f(x, t_0)]_i \), needed in lines 21-22 requires no more than \( t_0(c_i + 1) \) multiplications.

Observe that \( t_0 \) controls the precision of the power iteration method and is usually not larger than 100 even for a very large \( n \). Moreover, in [41] it is shown that even if \( n \) increases, \( t_0 \) does not need to increase faster than \( O(\log n) \) to keep the same precision. Consequently, in the worst case scenario, the computational complexity of Algorithm 1 is \( O(\log n) \), which makes it suitable for a large \( n \).

Finally, observe that regarding the memory usage of the algorithm, node \( i \) only needs to store \( c_i + 1 \) values (the \( i \)-th row of \( L \)) and therefore the storage requirement of each node does not increase with \( n \).

IV. ILLUSTRATIVE EXAMPLES

In this section we present two examples to illustrate how Algorithm 1 works.

A. EXAMPLE WITH RANDOMLY GENERATED DATA

In this example, we randomly generate a graph \( G \) that models a set of \( n = 20 \) individuals and their interactions. We consider two scenarios. In Scenario 1 (see Figure 1a), we assume that there is no information available about the risk level of severe illness from COVID-19 of each individual, nor about the time frames of their social contacts. Hence, we fix the weight of node \( i, \delta_i = 1 \), for all \( i \in V \). We also assume that all the social contacts have equal time frames and therefore we fix the weight of the edge \( \{i, j\}, \sigma_{i,j} = 1 \), for all \( \{i, j\} \in E \). In Scenario 2 (see Figure 2a), we consider the same graph \( G \), yet we assume that there is information available about the individual’s risk level and time frames of the social contacts. Such information is randomly generated both for the nodes and for the edges. In particular, all the weights are drawn from a uniform distribution between 0 and 1.

Figures 1b and 2b show the 2 clusters created by a single run of Algorithm 1 for Scenario 1 and Scenario 2, respectively.

Observe that the algorithm does not strictly separate the higher and the lower risk individuals. The clusters made by our algorithm depend on the risk of severe illness but also on the social interaction among individuals.

B. EXAMPLE WITH REAL DATA

In this example, we use data from the CoMix study [42] to generate a doubly-weighted graph \( G \) that models a set of \( n = 35 \) individuals. This study follows households all over Europe, collecting information about their behavioural patterns, measures, and proximity contacts, and how these have varied over time during the course of the COVID-19 pandemic. These results are published for an easier assessment of the spread of the virus, and they maintain the anonymity of the participants. For this example, CoMix social contact data
This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Author et al.: Preparation of Papers for IEEE TRANSACTIONS and JOURNALS

(a) Doubly-weighted graph with $n = 20$ nodes. The weights for the nodes and the edges are randomly drawn from a uniform distribution between 0 and 1. In the figure, the sizes of the nodes and the widths of the edges are proportional to their corresponding weights.

(b) Representation of the 2 clusters created by a single run of Algorithm 1 for the doubly-weighted graph shown in Figure 2a. In the figure, the sizes of the nodes and the widths of the edges are proportional to their corresponding weights.

FIGURE 2: Considered graph and the resulting clustering for Scenario 2.

From these data, $n$ random participants are selected. CoMix social contact data provides for each participant their number of contacts and the time frame of such contacts. We have further assumed that all the contacts of the selected individuals are within the considered population. We fix the weights of the nodes and the weights of the edges using the information provided by CoMix social contact data as shown in Tables 2 and 3, respectively.

Figure 3b shows the 2 clusters created by a single run of Algorithm 1 for the considered example.

V. CONCLUSION

In this paper, we have presented a distributed clustering algorithm that groups individuals according to their social contacts and the risk level of severe illness from COVID-19. Once the clusters are made, using a distributed consensus algorithm in each cluster, each individual can know the epidemiological situation of the group they belong to. Such knowledge allows them to take the social distancing

| Age range   | Weight of the node |
|-------------|--------------------|
| 18-29       | 1/6                |
| 30-39       | 1/5                |
| 40-49       | 1/4                |
| 50-59       | 1/3                |
| 60-69       | 1/2                |
| 70-120      | 1                  |

| Time frame         | Weight of the edge |
|--------------------|--------------------|
| less than 5 minutes| 1/16               |
| 5 - 15 minutes     | 1/8                |
| 15 - 60 minutes    | 1/4                |
| 1 - 4 hours        | 1/2                |
| more than 4 hours  | 1                  |

TABLE 2: Information provided by CoMix about the risk level of severe illness from COVID-19

TABLE 3: Information provided by CoMix about the time frame of the social contacts

(a) Doubly-weighted graph with $n = 35$ nodes. The weights for the nodes and the edges are set according to Tables 2 and 3.

(b) Representation of the 2 clusters created by a single run of Algorithm 1 for the doubly-weighted graph shown in Figure 3a.

FIGURE 3: Considered graph and the resulting clustering for the example with real data.
measures that correspond to the epidemiological situation of their group. By using this algorithm, the social distancing measures would only affect groups with high risk of infection instead of entire geographical regions, thus reducing the economic damage. The algorithm is designed so that individuals could know which group they belong to without having knowledge of the rest of the members of the group. Furthermore, there is no central entity with information about the groups because the algorithm only runs at a local level. Groups are created taking into account social contacts and the risk level of severe illness. Since social contacts change continuously and the risk level of severe illness also changes with the vaccination progress, our adaptive algorithm enables the creation of groups according to the information available at the time it is run. Finally, after the computational complexity analysis, we have concluded that our algorithm is sublinear with respect to the population size, which makes it very efficient.

APPENDIX A PROOF OF THEOREM 1

For the reader’s convenience, let $B = I_n - \frac{1}{\lambda_n}L$. Observe that

$$B = I_n - \frac{1}{\lambda_n}U\text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)U^T$$

$$= U\text{diag}(1, 1, \ldots, 1)U^T - U\text{diag}\left(\frac{\lambda_1}{\lambda_n}, \frac{\lambda_2}{\lambda_n}, \ldots, \frac{\lambda_n}{\lambda_n}\right)U^T$$

$$= U\text{diag}\left(1 - \frac{\lambda_1}{\lambda_n}, 1 - \frac{\lambda_2}{\lambda_n}, \ldots, 1 - \frac{\lambda_n}{\lambda_n}\right)U^T$$

is an eigenvalue decomposition of $B$. As $G$ has $k$ components, from [44, Section 4.4], we have $0 = \lambda_1 = \ldots = \lambda_k < \lambda_{k+1}$. Therefore,

$$x(t) = B^t x(0) = B^t \left(\sum_{i=1}^{n} \alpha_i u_i\right) = \sum_{i=1}^{n} \alpha_i B^t u_i$$

$$= \sum_{i=1}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t u_i$$

$$= \sum_{i=1}^{k} \alpha_i u_i + \sum_{i=k+1}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t u_i$$

for all $t \in \{0, 1, 2, \ldots\}$, where $\alpha_i = (x(0))^T u_i$ for all $i \in \{1, 2, \ldots, n\}$. Consequently,

$$x(t) - x(t-1) = \sum_{i=k+1}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t u_i - \sum_{i=k+1}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t^{t-1} u_i = \sum_{i=k+1}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t^{t-1} \left(-\frac{\lambda_i}{\lambda_n}\right) u_i$$

$$= \left(1 - \frac{\lambda_{k+1}}{\lambda_n}\right)^{t-1} \sum_{i=k+1}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t^{t-1} \left(-\frac{\lambda_i}{\lambda_n}\right) u_i$$

$$+ \sum_{i=k+2}^{n} \alpha_i \left(1 - \frac{\lambda_i}{\lambda_n}\right)_t^{t-1} \left(-\frac{\lambda_i}{\lambda_n}\right) u_i.$$
APPENDIX B PROOF OF THEOREM 2

Fix \( i \in \mathcal{V} \). Let

\[
\ell = \alpha_{k+1} \frac{\lambda_{k+1}}{\lambda_n} [u_{k+1}]_i,
\]

\[
\phi_t = \sum_{l=k+2}^{n} \alpha_l \kappa_l^t \frac{\lambda_l}{\lambda_n} [u_l]_i,
\]

and

\[
a_t = \frac{(\ell + \phi_{t-1})^{t-1}}{(\ell + \phi_t)^{t-2}},
\]

for all \( t \in \{2, 3, \ldots\} \). We have

\[
a_{t+1} = \frac{(\ell + \phi_{t-1})^t}{(\ell + \phi_t)^{t-1}} \leq \frac{(1 + \frac{1}{\ell^{t-1}})^t}{(1 + \frac{1}{\ell^{t-2}})^{t-1}} = \frac{1 + \frac{1}{\ell^{t-1}} - 1}{1 + \frac{1}{\ell^{t-2}} - 1},
\]

and

\[
a_t = \frac{(\ell + \phi_{t-2})^{t-1}}{(\ell + \phi_{t-1})^{t-2}} \leq \frac{(1 + \frac{1}{\ell^{t-2}})^{t-1}}{(1 + \frac{1}{\ell^{t-1}})^{t-2}} = \frac{1 + \frac{1}{\ell^{t-2}} - 1}{1 + \frac{1}{\ell^{t-1}} - 1},
\]

for all \( t \in \{2, 3, \ldots\} \). Analogously, we have

\[
a_{t+1} > \frac{\ell e^{\frac{\phi_t}{\ell}} - \frac{1}{\ell^{t-2}}}{e^{\frac{\phi_{t-1}}{\ell}} - \frac{1}{\ell^{t-1}}},
\]

\[
a_t < \frac{\ell e^{\frac{\phi_{t-2}}{\ell}} - \frac{1}{\ell^{t-3}}}{e^{\frac{\phi_{t-1}}{\ell}} - \frac{1}{\ell^{t-2}}},
\]

for all \( t \in \{2, 3, \ldots\} \). Analogously, in the case \( t \to \infty \), the sequence \( \{a_t\} \) converges to \(-\ell\) because

\[
\lim_{t \to \infty} \left| a_t + \ell \right| = \lim_{t \to \infty} \frac{a_{t+1} - \ell}{a_t - \ell} = \lim_{t \to \infty} \frac{a_t + \ell}{a_t - \ell} = \frac{\ell}{\ell} = 1.
\]

We now prove that the convergence of the sequence \( \{a_t\} \) is linear. On the one hand,

\[
\lim_{t \to \infty} \frac{|a_t + \ell|}{|a_t - \ell|} = \lim_{t \to \infty} \frac{|a_{t+1} - \ell|}{|a_t - \ell|} = \lim_{t \to \infty} \frac{\ell}{\ell} = 1
\]

\[
\leq \lim_{t \to \infty} \frac{t \left(1 - \frac{\lambda_k}{\lambda_n} \right) + \frac{\lambda_k}{\lambda_n} - 1}{\left(1 - \frac{\lambda_k}{\lambda_n} \right) + \frac{\lambda_k}{\lambda_n} - 1} = \lim_{t \to \infty} t \left(1 - \frac{\lambda_k}{\lambda_n} \right) + \frac{\lambda_k}{\lambda_n} - 1 = \frac{\lambda_k}{\lambda_n} - 1.
\]

Thus, \( \kappa_{k+2} = \frac{1}{1 + \frac{\lambda_k}{\lambda_n} - 1} = \frac{\lambda_n}{\lambda_n + \lambda_k} \)

\[
\lim_{t \to \infty} \frac{1}{\frac{\lambda_k}{\lambda_n} - 1} = \frac{\lambda_n}{\lambda_n + \lambda_k}.
\]

where

\[
\chi_t = \frac{\phi_t}{\kappa_{k+2}} = \frac{\sum_{l=k+2}^{n} \alpha_l \left( \frac{\kappa_l}{\kappa_{k+2}} \right)^t \frac{\lambda_l}{\lambda_n} [u_l]_i}{\lambda_n}
\]

for all \( t \in \{2, 3, \ldots\} \). Analogously, on the other hand,

\[
\lim_{t \to \infty} \frac{|a_t + \ell|}{|a_t - \ell|} = \frac{\lambda_k}{\lambda_n} - 1.
\]

From (9) and (10) we conclude that

\[
\lim_{t \to \infty} \frac{|a_{t+1} - \ell|}{|a_t - \ell|} = \frac{\lambda_k}{\lambda_n}.
\]
REFERENCES

[1] “Coronavirus disease (covid-19) pandemic,” https://www.who.int/emergencies/diseases/novel-coronavirus-2019, accessed: 2021-05-10.

[2] E. Dong, H. Du, and J. Gardner, “An interactive web-based dashboard to track covid-19 in real time,” The Lancet Infectious Diseases, vol. 20, no. 5, pp. 533–534, 2020.

[3] C. Cheng, J. Barceló, A. Hartnett, R. Kubinec, and L. Messerschmidt, “Covid-19 government response event dataset (coronavet v.1.0),” Nature Human Behaviour, vol. 4, pp. 756–768, 2020.

[4] N. M. Fei, et al., “Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand,” Imperial College, Tech. Rep., 03 2020.

[5] D. L. Chao, M. E. Halloran, V. J. Obenchain, and I. M. Longini, Jr, “FluTE, a publicly available stochastic influenza epidemic simulation model,” PLOS Computational Biology, vol. 6, no. 1, 01 2010.

[6] J. K. et al., “Interventions to mitigate early spread of SARS-CoV-2 in Singapore: a modelling study,” The Lancet Infectious Diseases, vol. 20, no. 6, pp. 678–688, 2020.

[7] T. Hable, N. Angrist, E. Cameron-Blake, L. Hallas, B. Kira, S. Majumdar, A. Petherick, T. Phillips, H. Tatlow, and S. Webster, “Variation in government responses to covid-19 (version 7.0),” University of Oxford, Tech. Rep., 09 2020.

[8] J. A. Lewnard and N. C. Lo, “Scientific and ethical basis for social-distancing interventions against covid-19,” The Lancet Infectious Diseases, vol. 20, no. 6, pp. 631–633, 2020.

[9] N. Fernandes, “Economic effects of coronavirus outbreak (covid-19) on the world economy,” IIEE Business School Working Paper, no. WP-1240-E, 2020.

[10] S. Maharaj and A. Kclezkowski, “Controlling epidemic speed by social distancing: Do it well or not at all,” BMC Public Health, vol. 12, no. 679, 2012.

[11] “Covid-19 digital rights tracker,” https://www.top10vpn.com/research/investigations/covid-19-digital-rights-tracker/, accessed: 2021-05-10.

[12] U. Gasser, M. Ienca, J. Scheibner, J. Sleigh, and E. Vayena, “Digital tools against covid-19: taxonomy, ethical challenges, and navigation aid,” The Lancet Digital Health, vol. 2, no. 8, pp. 425–434, 2020.

[13] S. Basagni, “Distributed clustering for ad hoc networks,” in Advances in Database Technology – EDBT 2004, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 88–105.

[14] M. Jia, Y. Wang, C. Shen, and G. Hig, “Privacy-preserving distributed clustering for electrical load profiling,” IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1429–1444, 2021.

[15] G. Muniraju, S. Zhang, C. T. M. Banavar, A. Spanias, C. Vargas-Rosas, and R. Villalpando-Hernandez, “Location based distributed spectral clustering for wireless sensor networks,” in 2017Sensor Signal Processing for Defence Conference (SSP2017).

[16] A. Bertrand and M. Moonen, “Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks,” Signal Processing, vol. 93, no. 5, pp. 1106–1117, 2013.

[17] X. Shiije, F. Jiayan, and L. X. Li, “Weighted laplacian method and its theoretical applications,” IOP Conference Series: Materials Science and Engineering, vol. 768, no. 072032, mar 2020.

[18] W. Chen, Y. Song, H. Bai, C. Lin, and E. Chang, “Parallel spectral clustering in distributed systems,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 3, pp. 568–586, 2011.

[19] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” Systems & Control Letters, vol. 53, pp. 65–78, 2004.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, ser. NIPS’01. Cambridge, MA, USA: MIT Press, 2001, pp. 849–856.

[21] E. Januzaj, H. Kriegel, and M. Pfeifle, “Dbdc: Density based distributed clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.

[22] D. J. Higham, G. Kalna, and M. Kibble, “Spectral clustering and its use in bioinformatics,” Journal of Computational and Applied Mathematics, vol. 204, no. 1, pp. 25–37, 2007.

[23] F. Chung and F. Graham, Spectral Graph Theory, ser. CBMS Regional Conference Series. Conference Board of the mathematical sciences, 1997.

[24] M. Fiedler, “A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory,” Czechoslovak Mathematical Journal, vol. 25, no. 4, pp. 619–633, 1975.

[25] B. Hendrickson and R. Leland, “An improved spectral graph partitioning algorithm for mapping parallel computations,” SIAM Journal on Scientific Computing, vol. 16, no. 2, pp. 452–469, 1995.

[26] G. Dahlquist and A. Björck, Numerical Methods. Dover, 2003.

[27] J. H. Wilkinson, The algebraic eigenvalue problem. Oxford University Press, 1965.

[28] C. Boutsidis, P. Kambadur, and A. A. Gittens, “Spectral clustering via the power method – provably,” in Proceedings of the 32nd International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 40–48.

[29] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web,” in Proceedings of the 7th International World Wide Web Conference, 1998, pp. 161–172.

[30] A. Girvan and M. Newman, “Community structure in social and biological networks,” Proc. Natl. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, 2002.

[31] A. Girvan and M. Newman, “Community structure in social and biological networks,” Proc. Natl. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, 2002.

[32] S. Fortunato, “Community detection in graphs,” Physics Reports, vol. 486, no. 3, pp. 75–174, 2010.

[33] M. Fiedler, “A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory,” Czechoslovak Mathematical Journal, vol. 25, no. 4, pp. 619–633, 1975.

[34] B. Hendrickson and R. Leland, “An improved spectral graph partitioning algorithm for mapping parallel computations,” SIAM Journal on Scientific Computing, vol. 16, no. 2, pp. 452–469, 1995.

[35] G. Dahlquist and A. Björck, Numerical Methods. Dover, 2003.

[36] J. H. Wilkinson, The algebraic eigenvalue problem. Oxford University Press, 1965.

[37] C. Boutsidis, P. Kambadur, and A. A. Gittens, “Spectral clustering via the power method – provably,” in Proceedings of the 32nd International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 40–48.

[38] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: Bringing order to the web,” in Proceedings of the 7th International World Wide Web Conference, 1998, pp. 161–172.

[39] A. Girvan and M. Newman, “Community structure in social and biological networks,” Proc. Natl. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, 2002.

[40] A. Girvan and M. Newman, “Community structure in social and biological networks,” Proc. Natl. Acad. Sci. USA, vol. 99, no. 12, pp. 7821–7826, 2002.
MARTA ZÁRRAGA-RODRÍGUEZ was born in Oviedo, Spain. She received the industrial engineering degree in 1993 and the Ph.D. degree in industrial engineering in 1999 from the University of Navarra, San Sebastián, Spain. Currently, she is an Assistant Professor at Tecnun, University of Navarra, San Sebastián, Spain. Her research interests include matrix analysis applied to problems in statistical signal processing, distributed computation, and information theory.

CAROLINA NOLASCO-FERENCÍKOVA was born in Madrid, Spain. She is pursuing the fourth year of the bachelor degree in telecommunication systems engineering in Tecnun, University of Navarra, San Sebastián, Spain. Currently, she is a student research assistant in the department of biomedical engineering and sciences at Tecnun.

JESÚS GUTIÉRREZ-GUTIÉRREZ (M’11–SM’11) was born in Granada, Spain. He received the degree in mathematics from the University of Granada, Granada, Spain, in 1999 and the Ph.D. degree in electronics and communications from the University of Navarra, San Sebastián, Spain, in 2004. Currently, he is Full Professor at Tecnun, University of Navarra, San Sebastián, Spain. He is also the head of the department of biomedical engineering and sciences at Tecnun. His research interests include matrix analysis applied to problems in statistical signal processing, distributed computation, and information theory.