iStock

Nola jarraitu birus baten hedapenari pribatutasun-arazorik sortu gabe

COVID-19aren pandemiaren eboluzioa aztertzeko, ezinbestekoa da birusaren hedapenari jarraitzea. Baina jarraitze-metodoek pribatutasun-arazo larriak eragin ditzakete, adibidez, jarraipena irizpide geografikoen arabera egiten bada. Horregatik, matematika erabilita, beste irizpide batzuen araberako metodoak garatu dira. Garrantzitsua da, pandemia batean ere, pertsonen pribatutasunaren alde egiten duten irizpideak ezartzea gaixotasun baten hedapenaren berri izan nahi denean.

Osasunaren Mundu Erakundeak pandemia deklaratu zuen 2020ko martxoan, eta birusak 117.700 milioi kasu baino gehiago eta 2.600 milioi heriotza eragin zituen 2021eko martxorako. Gaixotasunak edozein pertsonari eragin diezaioke; hala ere, badira arrisku handiko pertsona-talde batzuk, batez ere pertsona adinduak eta beste gaixotasun batzuk dituztenak. Testuinguru horretan, mundu osoko gobernuek zenbait erabaki hartu zituzten pandemia gehiago zabaltzea eragozteko: jendea konfinatzea, urruntze soziala eta abar. Neurri haietako bakoitzak ondorio ekonomiko batzuk zituen.

Oro har, pandemiaren eboluzioari irizpide geografikoen arabera jarraitu zaio, gune bateko populazioan bereizketarik egin gabe, eta gaixotutakoen jarraipena egiteko erraminta digitalek pribatutasun-kezka handiak eragin dituzte. Telefono mugikorretako aplikazioen bidez, eta telefono bakoitzaren Bluetooth seinalea baliatuta, pertsonen arteko gertutasuna kontrolatu izan da, infekzio-bideak atzemateko. Nolanahi ere, aplikazio horiek pertsonen informazio osoa jartzen diete eskura agintariei, eta pribatutasuna galdu egiten da.

Talde ez-geografikoak

Arazo horren aurrean, algoritmo berezi bat erabiltzea proposatu dute zenbait adituk. Aspalditik erabiltzen dira datu pertsonalak babesten dituzten metodo informatikoak, eta algoritmo horrek, hain zuzen ere, metodo horietako batzuk baliatzen ditu. Haien bitartez, pertsonak taldekatzen ditu, haien kontaktuak zein diren eta nolako arrisku-maila duten aztertuta. Horrela, datu pertsonalak babesten dituzten taldeak sortzen ditu. Talde bakoitzean adostasun-algoritmo bat erabiltzean, gizabanakoek taldearen egoera epidemiologikoari buruzko informazioa izan dezakete, eta, ondorioz, urruntze sozialeko neurriak egokitu. Ikuspegi espezifiko horrek bermatzen du arrisku handiko taldeetan soilik hartzea neurri zorrotzagoak, eta, hala, murrizketa geografiko zabalekin lotutako eragin ekonomikoa arintzen da.

«Algoritmo honek kontaktuen eta arrisku-mailaren arabera taldekatzen ditu pertsonak, baina pribatutasun pertsonala puskatu gabe»

Harrigarria bada ere, algoritmoak pribatutasun indibidualari eusten dio, eta erakunde zentralik gabe jarduten du; izan ere, pertsona bakoitzak bere taldeko afiliazioaz baino ez du izan behar kontziente, eta ez taldeko kide zehatzez. Algoritmoaren moldagarritasuna ezinbestekoa da, taldeak etengabe doitzen baititu gizarte-harremanen aldaketei eta arrisku-mailei erantzuteko, txertaketaren aurrerapena barne. Konplexutasun konputazionalaren analisiak algoritmoaren eraginkortasuna berresten du, haren baliabide-eskaerak populazioaren tamainarekin hazten baitira.

Horrelako metodoak erabilita, agintariek ez dute datu pertsonalik jaso behar taldeei segimendua egiteko. Beste pandemia bat baletor, horrelako tresnak erabili beharko lituzkete gobernuek, eraginkortasun handiko jarraipena egiteko, pribatutasuna puskatu gabe.

InpAkta 00 (2023) 308-320

https://doi.org/10.1109/ACCESS.2021.3131777

Distributed clustering algorithm for adaptive pandemic control

Xabier Insausti*, Marta Zárraga-Rodríguez, Carolina Nolasco-Ferencikova and Jesús Gutiérrez-Gutiérrez

Tecnun University of Navarra

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.[1]

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

1 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].

* **Corresponding author / Harremanetan jartzeko:** Xabier Insausti. Tecnun, University of Navarra, San Sebastián. - https://orcid.org/0000-0001-9628-0681. **Contributed by** Xabier Insausti Sarasola **editoreak egindako ekarpena**.

How to cite / Nola aipatu: X. Insausti, *et al.* Nolasco-Ferencikova and J. Gutiérrez-Gutiérrez, «Distributed Clustering Algorithm for Adaptive Pandemic Control», in *IEEE Access*, 2021, vol. 9, 160688-160696 (https://doi.org/10.1109/ACCESS.2021.3131777)

Published by UPV/EHU Press under the terms of the Creative Commons Attribution License http://creativecommons.org/licenses/by/4.0/, which permits unrestricted use, provided the original author and source are credited

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 dis- advantaged 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 technologies 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 plat-

form 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 2 states preliminary considerations regarding distributed computation and spectral clustering. Section 3 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 4 and 5, respectively.

2. Preliminaries

2.1. *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 $G = (\mathrm{V}, \mathcal{E})$ with no loops, where $\mathrm{V} = \{1, 2, \ldots, n\}$ is the set of nodes (vertices) and $\mathcal{E}$ is the set of edges. If two nodes $i, j \in \mathrm{V}$ interact between them, then $\{i, j\} \in \mathcal{E}$. We say that these nodes are connected, and can therefore interchange information. Conversely, if $\{\mathrm{i}, \mathrm{j}\} \notin \mathcal{E}$, this means that nodes $i, j \in \mathrm{V}$ are not connected and cannot interchange information.

We assume that each node $i \in \mathrm{V}$ 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 interchanging 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

$$x_i(t+1) = w_{i,i}x_i(t) + \sum_{j\in\mathcal{V}:\{i,\,\mathrm{j}\}\in\mathcal{E}} w_{i,j}x_j(t), \qquad i \in \mathcal{V}, \tag{1}$$

where $w_{i,j} \in \mathbb{R}$ and time $t \in \{0, 1, 2, \ldots\}$ is assumed to be discrete (see [31]). Let $x(t) = (x_1(t), x_2(\mathrm{t}), \ldots, x_n(t))^\top$ 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

$$\boldsymbol{x}(t+1) = W\,\boldsymbol{x}(t) = W^{t+1}\,\boldsymbol{x}(0), \tag{2}$$

where W is the $n \times n$ matrix defined as

$$[W]_{i,j} = \begin{cases} 0 & \text{if } i \neq j \text{ and } \{i,j\} \notin \mathcal{E}, \\ w_{i,j} & \text{otherwise,} \end{cases} \tag{3}$$

for $i, j \in \mathrm{V}$.

2.2. *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 considered here is the so-called *spectral clustering* (see, e.g., [32–34]). Spectral clustering is based on the information provided by the 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 $i \in \mathrm{V}$, and whenever $\{\mathrm{i}, \mathrm{j}\} \in \mathcal{E}$ 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

$$L = \Lambda^{-\frac{1}{2}}(D - \Sigma)\Lambda^{-\frac{1}{2}}, \tag{4}$$

where Λ^{-2} is the $n \times n$ diagonal matrix with $[\Lambda^{-\frac{1}{2}}]_{i,i} = \dfrac{1}{\sqrt{\delta_i}}$,

$$[\Sigma]_{i,j} = \begin{cases} \sigma_{i,j} & \text{if } \{i,j\} \in \mathcal{E}, \\ 0 & \text{if } \{i,j\} \notin \mathcal{E}, \end{cases}$$

and D is the $n \times n$ diagonal matrix with $[D]_{i,i} = \sum_{j=1}^{n} [\Sigma]_{i,j}$.

From [37, Theorem 5.1], L is positive semidefinite. Let $L = U\,\mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda n)U^{-1}$ be an eigenvalue decomposition of L, where the eigenvalues are arranged in non-decreasing order and the eigenvector matrix $U = U = [\boldsymbol{u}_1 | \boldsymbol{u}_2 | \ldots | \boldsymbol{u}_n]$ is real and orthogonal. Assume that G has k components. Then, $\lambda_1 = \ldots = \lambda_k = 0$. In [37, Section 5.1] it is shown that $[\boldsymbol{u}_{k+1}]_i$ indicates which cluster the node i belongs to.

3. Distributed clustering algorithm for adaptive pandemic control

3.1. *Proposed algorithm*

Consider a set of n individuals that interact in a certain geographical region. The entire population and the interactions among them will be modelled with a doubly-weighted undirected graph G with no loops. The node i of the graph represent the i-th individual and the weight of the node i, δ_i, represents the individual's risk level of severe illness from COVID-19. The edge $\{i, j\}$ of the graph represents that there exists an interaction between individuals i and j, and the weight of the edge, $\sigma_{i,j}$, represents the time frame of the social contact between them.

In this section we present an algorithm that adaptively groups individuals according to their social contacts and their risk level of severe illness from COVID-19, that is, we present an algorithm for clustering the doubly-weighted graph G. Since the goal is to keep privacy from authorities, the algorithm presented here is fully distributed. Specifically, it computes the eigenvector $\boldsymbol{u}_{k+1}$ of the Laplacian matrix L of the graph G in a distributed way (see Algorithm 1).

The rest of this section is devoted to proving that $\boldsymbol{u}k{+}1$ can be computed in a distributed way (Theorem 1). Theorem 1 directly provides the steps of Algorithm 1.

Theorem 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) *with $\lambda k{+}1 < \lambda k{+}2$. Then, for almost every column vector $\boldsymbol{u}(0)$,*

$$\lim_{t\to\infty} \big[\boldsymbol{x}(t) - \boldsymbol{x}(t-1)\big]_i \left(\frac{\big[\boldsymbol{x}(t-1) - \boldsymbol{x}(t-2)\big]_i}{\big[\boldsymbol{x}(t) - \boldsymbol{x}(t-1)\big]_i} \right)^{t-1} = C\big[\boldsymbol{u}_{k+1}\big]_i \tag{5}$$

for all $i \in \mathrm{V}$, *where*

$$\boldsymbol{x}(t+1) = \left(I_n - \frac{1}{\lambda_n} L\right) \boldsymbol{x}(t) \qquad \forall t \in \{0,1,2,\ldots\}, \tag{6}$$

C is a non-zero constant, and In denotes the n × n identity matrix.

Proof: See Appendix 6.

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 $i \in \mathrm{V}$ can know the i-th entry of an eigenvector associated to λ_{k+1}. However, in order to compute (6) in a distributed way, each node needs to know λ_n. Lemma 1 shows that λ_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* $\boldsymbol{y}(0)$,

$$\lim_{t\to\infty} \frac{[\boldsymbol{y}(t)]_i}{[\boldsymbol{y}(t-1)]_i} = \lambda_n \qquad \forall i \in \mathcal{V}, \tag{7}$$

where

$$\boldsymbol{y}(t+1) = L\boldsymbol{y}(t) \qquad \forall t \in \{0,1,2,\ldots\}. \tag{8}$$

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 $i \in \mathrm{V}$ can know λ_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 $i \in \mathrm{V}$.

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

$$f(\boldsymbol{x}, t) := W^t \boldsymbol{x}(0),$$

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 λ_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 λ_2 according to
24-26	Assign node i to a cluster depending on the sign of $[u_{k+1}]i$

Algorithm 1
Distributed clustering algorithm for adaptive pandemic control

for all nodes $i \in \mathcal{V}$ **do**
  $s_i \leftarrow 0$
  for all nodes j connected to i **do**
    $w_{i,j} \leftarrow \frac{-\sigma_{i,j}}{\sqrt{\delta_i \delta_j}}$
    $s_i \leftarrow s_i + \sigma_{i,j}$
  end for
  $w_{i,i} \leftarrow \frac{s_i}{\delta_i}$
  $[\boldsymbol{y}(0)]_i \leftarrow \text{rand}()$ > An arbitrary value
end for
for all nodes $i \in \mathcal{V}$ **do**
  $\lambda_n \leftarrow \frac{[f(\boldsymbol{y}, t_0)]_i}{[f(\boldsymbol{y}, t_0 - 1)]_i}$
end for
for all nodes $i \in \mathcal{V}$ **do**
  for all nodes j connected to i **do**
    $w_{i,j} \leftarrow -\frac{w_{i,j}}{\lambda_n}$
  end for
  $w_{i,i} \leftarrow 1 - \frac{w_{i,i}}{\lambda_n}$
  $[\boldsymbol{x}(0)]_i \leftarrow \text{rand}()$ 1 > An arbitrary value
end for
for all nodes $i \in \mathcal{V}$ **do**
  $\beta_i \leftarrow [f(\boldsymbol{x}, t_0)]_i - [f(\boldsymbol{x}, t_0 - 1)]_i$
  $\gamma_i \leftarrow [f(\boldsymbol{x}, t_0 - 1)]_i - [f(\boldsymbol{x}, t_0 - 2)]_i$
  $[C\boldsymbol{u}_{k+1}]_i \leftarrow \beta_i \left(\frac{\gamma_i}{\beta_i}\right)^{t_0 - 1}$
  if $[C\boldsymbol{u}_{k+1}]_i > 0$ then node i belongs to cluster 1
  else node i belongs to cluster 2
  end if
end for

3.2. 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 ℓ, 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*

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

is linear for all $i \in \mathrm{V}$.

Proof: See Appendix 7.

Since the convergence of the sequence considered in Lemma 1 is also linear (see [38, Section 5.8.1]), we conclude that the overall convergence of Algorithm 1 is linear.

3.3. 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 $[f(\boldsymbol{y}, t_0)]_i$ needed in line 11 requires no more than t0 (c_i+1) multiplications (see Equation (1)). Similarly, the computation of $[f(\boldsymbol{x}, t_0)]_i$ needed in lines 21-22 requires no more than t0 (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, t0 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.

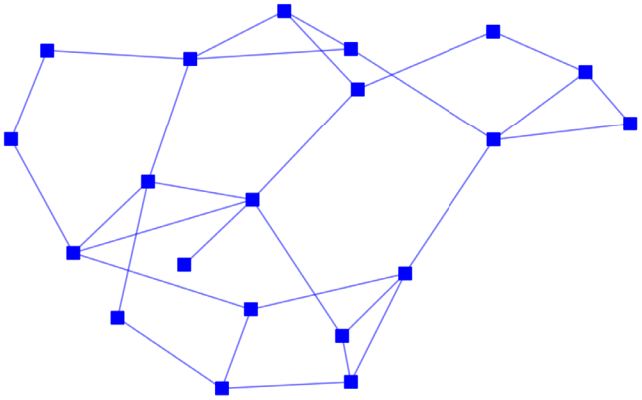
4. Illustrative examples

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

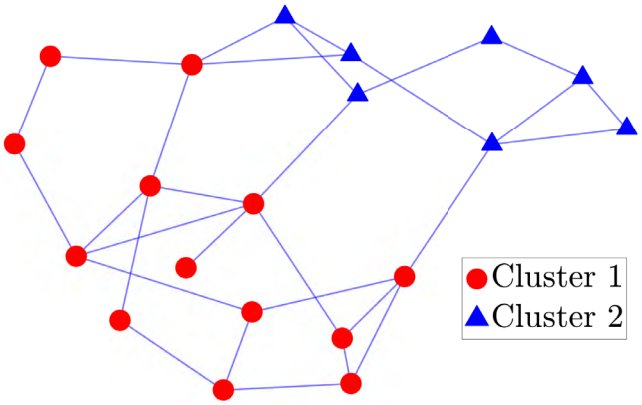
4.1. Example with randomly generated data

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 $i \in \mathrm{V}$. 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 $\{i, j\} \in \mathcal{E}$. 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.



(a) Unweighted graph with n = 20 nodes.

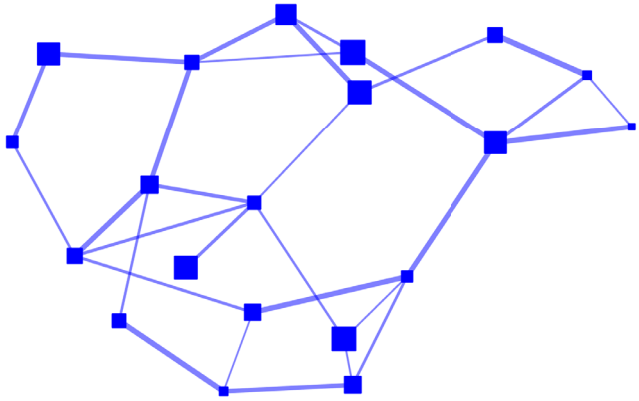


(b) Representation of the 2 clusters created by a single run of Algorithm 1 for the unweighted graph shown in Figure 1a.

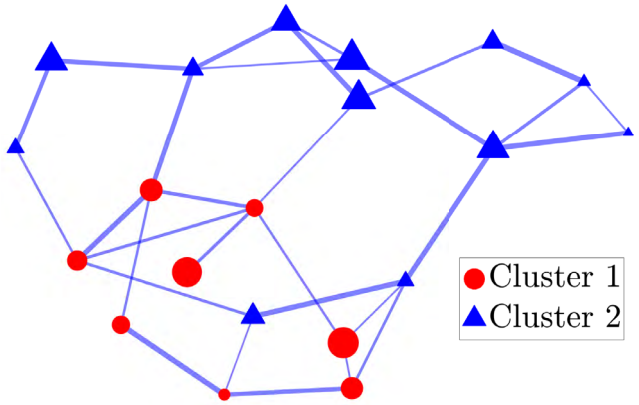
Figure 1. Considered graph and the resulting clustering for Scenario 1

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.



(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

4.2. *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 from Spain were used [43].

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.

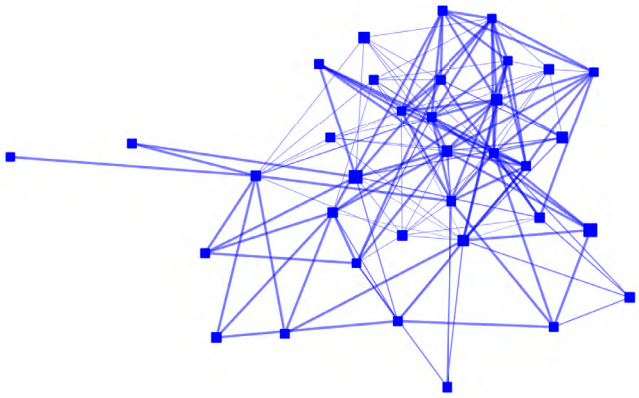
Table 2
Information provided by CoMix about the risk level of severe illness from COVID-19

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

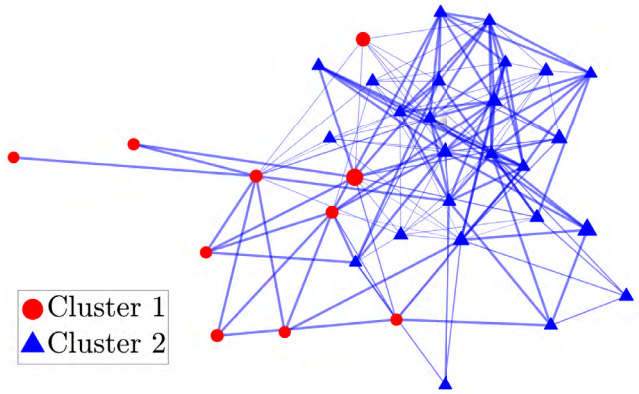
Table 3
Information provided by CoMix about the time frame of the social contacts

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

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



(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

5. 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 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.

6. 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 L. 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 (coronanet v.1.0)," *Nature Human Behaviour*, vol. 4, pp. 756-768, 2020.

[4] N. M. F. *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. Hale, 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," *IESE Business School Working Paper*, no. WP-1240-E, 2020.

[10] S. Maharaj and A. Kleczkowski, "Controlling epidemic spread 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] M. E. Kretzschmar, G. Rozhnova, M. C. J. Bootsma, M. van Boven, J. H. H. M. van de Wijgert, and M. J. M. Bonten, "Impact of delays on effectiveness of contact tracing strategies for covid-19: a modelling study," *The Lancet Public Health*, vol. 5, no. 8, pp. 452-459, 2020.

[14] "Tracetogether," https://www.tracetogether.gov. sg/index. html, accessed: 2021-05-10.

[15] H. Cho, D. Ippolito, and Y. W. Yu, "Contact tracing mobile apps for covid-19: Privacy considerations and related trade-offs," arXiv, 2020.

[16] K. Michael and R. Abbas, "Behind covid-19 contact trace apps: The google-apple partner-ship," *IEEE Consumer Electronics Magazine*, vol. 9, no. 5, pp. 71-76, 2020.

[17] S. Jha, L. Kruger, and P. McDaniel, "Privacy preserving clustering," in *Computer Security – ESORICS 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 397-417.

[18] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk, "Privacy-preserving distributed clustering," *EURASIP Journal on Information Security*, no. 4, 2013.

[19] S. Oliveira and O. Zaiane, "Privacy preserving clustering by data transformation," *Journal of Information and Data Management*, vol. 1, no. 1, 2010.

[20] S. Merugu and J. Ghosh, "Privacy-preserving distributed clustering using generative models," in *Third IEEE International Conference on Data Mining*, 2003, pp. 211-218.

[21] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. New York, NY, USA: Association for Computing Machinery, 2005, pp. 593-599.

[22] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, 1999, pp. 310-315.

[23] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson, "Distributed clustering using collective principal component analysis," *Knowledge and Information Systems*, vol. 3, pp. 422-448, 2001.

[24] P. A. Forero, A. Cano, and G. B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 707-724, 2011.

[25] E. Januzaj, H. Kriegel, and M. Pfeifle, "Dbdc: Density based distributed clustering," in *Advances in Database Technology - EDBT 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg,2004, pp. 88-105.

[26] M. Jia, Y. Wang, C. Shen, and G. Hug, "Privacy-preserving distributed clustering for electricalload profiling," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1429-1444, 2021.

[27] G. Muniraju, S. Zhang, C. T. M. Banavar, A. Spanias, C. Vargas-Rosales, and R. Villalpando-Hernandez, "Location based distributed spectral clustering for wireless sensor networks," in *2017 Sensor Signal Processing for Defence Conference (SSPD)*, 2017.

[28] 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.

[29] X. Shijie, 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.

[30] 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.

[31] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, pp. 65-78, 2004.

[32] 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.

[33] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[34] 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.

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

[36] 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.

[37] 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.

[38] G. Dahlquist and A. Bjorck, *Numerical Methods*. Dover, 2003.

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

[40] 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.

[41] 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.

[42] "The comix study," https://www.uhasselt.be/UH/71795-start/The-CoMix-study, accessed: 2021-09-22.

[43] A. Gimma, K. L. Wong, P. Coletti, and C. I. Jarvis, "Comix social contact data (spain)," Jun. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5040840

[44] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75-174, 2010.