Linear Clustering Process on Networks

Jokic, Ivan; Mieghem, P. Van

DOI
10.1109/TNSE.2023.3271360

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Network Science and Engineering

Linear Clustering Process on Networks

Ivan Jokić and Piet Van Mieghem

***Abstract*—We propose a linear clustering process on a network consisting of two opposite forces: attraction and repulsion between adjacent nodes. Each node is mapped to a position on a one-dimensional line. The attraction and repulsion forces move the nodal position on the line, depending on how similar or different the neighbourhoods of two adjacent nodes are. Based on each node position, the number of clusters in a network and each node's cluster membership is estimated. The performance of the proposed linear clustering process is benchmarked on synthetic networks against widely accepted clustering algorithms such as modularity, Leiden method, Louvain method and the non-back tracking matrix. The proposed linear clustering process outperforms the most popular modularity-based methods, such as the Louvain method, on synthetic and real-world networks, while possessing a comparable computational complexity.**

***Index Terms*—Communities, graph clustering, modularity, linear process.**

I. Introduction

NETWORKS [1], [2] abound and increasingly shape our world, ranging from infrastructural networks (transportation, telecommunication, power-grids, water, etc.) over social networks to brain and biological networks. In general, a network consists of a graph or underlying topology and a dynamic process that takes place on the network. Some examples of processes on a network are percolation [3] and epidemic spreading [4], [5], that possess a phase transition [6], [7]. While most real-world processes on networks are non-linear, linearisation allows for hierarchical structuring of processes on the network [8].

The identification of communities and the corresponding hierarchical structure in real-world networks has been an active research topic for decades [9], although a single, precise definition of a community does not seem to exist [10], [11]. In network science, a community is defined as a set of nodes that share links dominantly between themselves, while a minority of links is shared with other nodes in the network. Newman proposed in [12] a spectral clustering algorithm that reveals hierarchical structure of a network, by optimising modularity, a commonly used quality function of a graph partition. Xu et al. proposed an efficient clustering algorithm in [13], capable of detecting clusters while differentiating between hub and outlier nodes. A heuristic, modularity-based two-step clustering algorithm, proposed by Blondel et al. in [14], has proved to be computationally efficient and performed among the best in the comparative study conducted in [15]. Recently, Peixoto proposed in [16] a nested generative model, able to identify nested partitions at different resolutions, which thus overcomes an existing drawback of a majority of clustering algorithms, identifying small, but well-distinguished communities in a large network. Dannon et al. concluded in their comparative study [17] that those clustering algorithms performing the best tend to be less computationally efficient. A class of clustering algorithms exists, that perform clustering based on a dynamic process on the network, such as a random walk [18], consensus process [19] or synchronisation [20]. We refer to [9], [21] for a detailed review on existing clustering algorithms.

Our new idea is the proposal of a linear clustering process (LCP) on a graph, where nodes move in a one-dimensional space and tend to concentrate in groups that lead to network communities and therefore solve the classical[1] community detection problem. Linear means "proportional to the graph", which is needed, because the aim is to cluster the graph and the process should only help and not distract from our main aim of clustering. A non-linear process depends intricately on the underlying graph that we want to cluster and may result in worse clustering! Our LCP leads to a new and non-trivial graph matrix W in (10) in Theorem 1, whose spectral decomposition is at least as good as the best clustering result, based on the non-back tracking matrix [22]. Moreover, the new graph matrix W has a more "natural" relation to clustering than the non-back tracking matrix, that was not designed for clustering initially. Finally, our resulting LCP clustering algorithm seems surprisingly effective and can compete computationally with any other clustering algorithm, while achieving generally a better result!

In Section II, we introduce notations for graph partitioning and briefly review basic theory on clustering such as modularity, normalised mutual information (NMI) measure and different synthetic benchmarks. We introduce the linear clustering process (LCP) on a network in Section III, while the resulting community detection algorithm is described in Section IV and Section V. We compare the performance of our LCP algorithm with that of the non-back tracking matrix, Newman's, Leiden and the Louvain algorithm and provide results in Section VI, after which we conclude.

Manuscript received 28 September 2022; revised 7 March 2023; accepted 24 April 2023. Date of publication 1 May 2023; date of current version 25 October 2023. The work of Piet Van Mieghem was supported by the European Research Council through the European Union's Horizon 2020 Research and Innovation Program under Grant 101019718. Recommended for acceptance by Dr. Xiang Li. *(Corresponding author: Ivan Jokić.)*

The authors are with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: I.Jokic@tudelft.nl; P.F.A.VanMieghem@tudelft.nl).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSE.2023.3271360, provided by the authors.

Digital Object Identifier 10.1109/TNSE.2023.3271360

[1] A solution of the classical (or standard) community problem consists of assigning a cluster membership to each node in a network.

II. Network or Graph Clustering

A graph $G(\mathcal{N}, \mathcal{L})$ consists of a set $\mathcal{N}$ of $N = |\mathcal{N}|$ nodes and a set $\mathcal{L}$ of $L = |\mathcal{L}|$ links and is defined by the $N \times N$ adjacency matrix A, where $a_{ij} = 1$ if node i and node j are connected by a link, otherwise $a_{ij} = 0$. The $N \times 1$ degree vector d obeys $d = A \cdot u$, where the $N \times 1$ all-one vector u is composed of ones. The corresponding $N \times N$ degree diagonal matrix is denoted by $\Delta = \text{diag}(d)$.

The set of neighbours of node i is denoted by $\mathcal{N}_i = \{k \mid a_{ik} = 1, k \in \mathcal{N}\}$ and the degree of node i equals the cardinality of that set, $d_i = |\mathcal{N}_i|$. The set of common neighbours of node i and node j is $\mathcal{N}_i \cap \mathcal{N}_j$, while the set of neighbours of node i that do not belong to node j is $\mathcal{N}_i \setminus \mathcal{N}_j$. The degree of a node i also equals the sum of the number of common and different neighbours between nodes i and j

$$d_i = |\mathcal{N}_i \setminus \mathcal{N}_j| + |\mathcal{N}_i \cap \mathcal{N}_j| \tag{1}$$

The number of common neighbours between nodes i and j equals the ij-th element of the squared adjacency matrix

$$|\mathcal{N}_i \cap \mathcal{N}_j| = \left(A^2\right)_{ij} \tag{2}$$

because $(A^k)_{ij}$ represents the number of walks with k hops between node i and node j (see [23, p. 32]). From (1), (2) and $d_i = (Au)_i = (A^2)_{ii}$, we have

$$|\mathcal{N}_i \setminus \mathcal{N}_j| = \left(A^2\right)_{ii} - \left(A^2\right)_{ij}$$

and

$$|\mathcal{N}_i \setminus \mathcal{N}_j| + |\mathcal{N}_j \setminus \mathcal{N}_i| = \left(A^2\right)_{ii} + \left(A^2\right)_{jj} - 2\left(A^2\right)_{ij}$$

The latter expression is analogous to the effective resistance ω_{ij} between node i and node j,

$$\omega_{ij} = Q^{\dagger}_{ii} + Q^{\dagger}_{jj} - 2Q^{\dagger}_{ij}$$

in terms of the pseudoinverse $Q^{\dagger}_{ii}$ of the Laplacian matrix $Q = \Delta - A$ (see e.g. [24]).

Before introducing our linear clustering process (LCP) in Section III, we briefly present basic graph partitioning concepts, while the overview of the more popular clustering methods is deferred to Appendix A.

A. Network Modularity

Newman and Girvan [25] proposed the modularity as a concept for a network partitioning,

$$m = \frac{1}{2\,L} \cdot \sum_{i=1}^{N} \sum_{j=1}^{N} \left(a_{ij} - \frac{d_i \cdot d_j}{2\,L}\right) \cdot \mathbf{1}_{\{i \text{ and } j \in \text{ same cluster}\}}, \tag{3}$$

where $\mathbf{1}_x$ is the indicator function that equals 1 if statement x is true, otherwise $\mathbf{1}_x = 0$. The modularity m compares the number of links between nodes from the same community with the expected number of intra-community links in a network with randomly connected nodes. When the modularity m close to 0, the estimated partition is as good as a random partition would be. On the contrary, a modularity m close to 1 indicates that the network can be clearly partitioned into clusters. Optimising the modularity is proven to be NP-complete [26] and approximated in [27]. Defining the $N \times N$ modularity matrix C,

$$C_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ belong to the same cluster} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

allows us to rewrite the modularity (3) as a quadratic form,

$$m = \frac{1}{2\,L} \cdot u^T \cdot \left(A \circ C - \frac{1}{2\,L} \cdot \left(d \cdot d^T\right) \circ C\right) \cdot u, \tag{5}$$

where $\circ$ denotes the Hadamard product [28]. The number of clusters in a network is denoted by c, while the $c \times 1$ vector $n = [n_1\ n_2\ \ldots\ n_c]$ defines the size of each cluster, where the number of nodes in cluster i is denoted as n_i.

B. Normalised Mutual Information

Danon et al. [17] proposed the normalised mutual information (NMI) metric, based on a confusion matrix F, whose rows correspond to the original communities, while its columns are related to estimated clusters. Therefore the element F_{ij} of the confusion matrix denotes the number of nodes in the real community i, that also belong to the estimated community j. The normalised mutual information metric between the known P_0 and the estimated partition P_e, denoted as $I_n(P_0, P_e)$, is defined in [17] as follows

$$I_n(P_0, P_e) = \frac{-2\sum_{i=1}^{c_0}\sum_{j=1}^{c_e} F_{ij} \log\left(\frac{F_{ij}N}{F_{i.}F_{.j}}\right)}{\sum_{i=1}^{c_0} F_{i.} \log\left(\frac{F_{i.}}{N}\right) + \sum_{j=1}^{c_e} F_{.j} \log\left(\frac{F_{.j}}{N}\right)}, \tag{6}$$

where the known and the estimated number of clusters are denoted as c_0 and c_e, respectively, the i-th row sum of F is denoted as $F_{i.}$, while its j-th column-sum is denoted as $F_{.j}$. In case two graph partitions are identical, the corresponding NMI measure equals 1, while tending to 0 when two partitions are independent. The NMI measure has been extensively used ever since, while analysing the performance of different clustering algorithms [9].

C. Benchmarks

The performance of the clustering methods in this paper are benchmarked on random graphs, generated by the Stochastic Block Model (SBM), proposed by Holland [29]. The SBM model generates a random graph with community structure, where a link between two nodes exists with different probability, depending on whether the nodes belong to the same cluster or not. We provide additional information on the stochastic block model in Appendix B.1.

Girvan and Newman [30] focused on a special case of the SBM model (GN benchmark), where the graph consists of $N = 128$ nodes, distributed in $c = 4$ communities of equal size, while fixing the average degree $E[D] = 16$. The GN benchmark has been extensively used in literature, despite introducing strong assumptions, such as communities of equal size, each node having the same degree and fixed graph size. Therefore, Lancichinetti et al. [31] proposed the LFR benchmark, where both the node degree vector d and community size vector n follows a power law distribution, a property found in many

real-world networks. Additional details on LFR benchmark are deferred to Appendix B.2.

III. Linear Clustering Process (LCP) on a Graph

A. Concept of the Clustering Process

Each node i in the graph G is assigned a position $x_i[k]$ on a line (i.e. in one-dimensional space) at discrete time k. We define the $N \times 1$ position vector $x[k]$ at discrete time k, where the i-th vector component consists of the position $x_i[k]$ of node i at time k. We initialize the $N \times 1$ position vector $x[0]$ by placing nodes equidistantly on the line and assign integer values from 1 to N to the nodes, thus, $x[0] = [1\ 2\ \ldots\ N]^T$. At last, we restrict the position $x_i[k]$ to positive real values.

We propose a dynamic process that determines the position of nodes over time. The position difference between nodes of the same cluster is relatively small. On the contrary, nodes from different clusters are relatively far away, i.e. their position difference is relatively high. Based on the position vector $x[k]$, we will distinguish clusters, also called communities, in the graph G.

The proposed clustering process consists of two opposite and simultaneous forces that change the position of nodes at discrete time k:

Attraction. Adjacent nodes sharing many neighbours are mutually attracted with a force proportional to the number of common neighbours. In particular, the attractive force between node i and its neighboring node j is proportional to $\alpha \cdot (|\mathcal{N}_j \cap \mathcal{N}_i| + 1)$, where α is the attraction strength and $(|\mathcal{N}_j \cap \mathcal{N}_i| + 1)$ equals the number of common neighbors plus the direct link, i.e. $a_{ij} = 1$.

Repulsion. Adjacent nodes sharing a few neighbours are repulsed with a force proportional to the number of different neighbours. The repulsive force between node i and its neighboring node j is proportional to $\delta \cdot (|\mathcal{N}_j \setminus \mathcal{N}_i| - 1)$, where δ is the repulsive strength and $(|\mathcal{N}_j \setminus \mathcal{N}_i| - 1)$ equals the set of neighbours of node j that do not belong to node i minus the direct link (that is included in $|\mathcal{N}_j \setminus \mathcal{N}_i|$). Since the force should be symmetric and the same if i and j are interchanged, we end up with a resultant repulsive force proportional to $\frac{1}{2} \cdot \delta \cdot (|\mathcal{N}_j \setminus \mathcal{N}_i| + |\mathcal{N}_i \setminus \mathcal{N}_j| - 2)$.

B. LCP in Discrete Time

Since computers operate with integers and truncated real numbers, we concentrate on discrete-time modeling. The continuous-time description is derived in Appendix C. We denote the continuous-time variables by $y(t)$ and the continuous time by t, while the discrete-time counterpart is denoted by $y[k]$, where the integer k denotes the discrete time or k-th timeslot. The transition from the continuous-time derivative to the discrete-time difference is

$$\frac{dx_i(t)}{dt} = \lim_{\Delta t \to 0} \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} \to \left. \frac{x_i(t+\Delta t) - x_i(t)}{\Delta t} \right|_{\Delta t = 1} \stackrel{\text{def}}{=} x_i[k+1] - x_i[k]$$

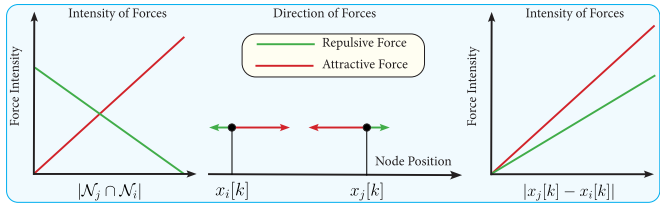


Fig. 1. Dependence of the attractive and repulsive force on the number of common neighbours of adjacent nodes i and j (left-figure). Directions of the attraction and repulsion forces between the adjacent nodes (middle-figure). Dependence of the attractive and repulsive force on the absolute position distance between adjacent nodes i and j (right-figure).

Corresponding to the continuous-time law in Appendix C and choosing the time step $\Delta t = 1$, the governing equation of position $x_i[k]$ of node i at discrete time k is

$$x_i[k+1] = x_i[k] + \sum_{j \in \mathcal{N}_i} \left(\frac{\alpha \cdot (|\mathcal{N}_j \cap \mathcal{N}_i| + 1)}{d_j d_i} - \frac{\frac{1}{2} \cdot \delta \cdot (|\mathcal{N}_j \setminus \mathcal{N}_i| + |\mathcal{N}_i \setminus \mathcal{N}_j| - 2)}{d_j d_i} \right) \cdot (x_j[k] - x_i[k]) \quad (7)$$

where α and δ are, in the discrete-time setting, the strength (in dimensionless units) for attraction and repulsion, respectively. The maximum position difference at the initial state is $x_N[0] - x_1[0] = N - 1$.

Node j attracts an adjacent node i with force proportional to their position difference $(x_j[k] - x_i[k])$. The intensity of the attractive force decreases as nodes i and j are closer on a line. The attraction is also proportional to the number common neighbours $|\mathcal{N}_j \cap \mathcal{N}_i|$ of node i and node j plus the direct link, as nodes tend to share most links with other nodes from the same cluster. On the contrary, node j repulses node i with a rate proportional to their position difference $(x_j[k] - x_i[k])$ and the average of the number of node j neighbours $|\mathcal{N}_j \setminus \mathcal{N}_i|$ that are not connected to the node i and, similarly, the number of node i neighbors, $|\mathcal{N}_j \setminus \mathcal{N}_i|$ that are not connected to the node j. The repulsive and attractive force are, as mentioned above, symmetric in strength, but opposite, if i is interchanged by j.

The directions of both attractive and repulsive forces between two adjacent nodes i and j as well the dependence of both forces on the number of common neighbours $|\mathcal{N}_j \cap \mathcal{N}_i|$ and the absolute position distance $|x_j[k] - x_i[k]|$ are illustrated in Fig. 1.

In the continuous-time setting, as provided in Appendix 39, we eliminate one parameter by scaling the time $t^* = \delta t$. Because the time step $\Delta t = 1$ is fixed and cannot be scaled, the discrete-time model consists of two parameters $\alpha \geq 0$ and $\delta \geq 0$.

So far, we have presented an additive law, derived in the common Newtonian approach. The corresponding multiplicative law in discrete time is

$$x_i[k+1] = x_i[k] \cdot \left(1 + \sum_{j \in \mathcal{N}_i} \left(\frac{\alpha \cdot (|\mathcal{N}_j \cap \mathcal{N}_i| + 1)}{d_i \cdot d_j} - \frac{\frac{1}{2} \cdot \delta \cdot (|\mathcal{N}_j \setminus \mathcal{N}_i| + |\mathcal{N}_i \setminus \mathcal{N}_j| - 2)}{d_i \cdot d_j} \right) \cdot (x_j[k] - x_i[k]) \right) \quad (8)$$

Although the physical intuition is similar, the multiplicative process in (8) behaves different in discrete time than the additive law in (7). Since also the analysis is more complicated, we omit a further study of the multiplicative law.

We present the analogon of (7) in matrix form:

Theorem 1: The discrete time process (7) satisfies the linear matrix difference equation

$$x[k+1] = (I + W - \text{diag}\,(W \cdot u)) \cdot x[k], \tag{9}$$

where the $N \times 1$ vector u is composed of ones, the $N \times N$ identity matrix is denoted by I, while the $N \times N$ topology-based matrix W is defined as

$$W = (\alpha + \delta)\,\Delta^{-1} \cdot \left(A \circ A^2 + A\right) \cdot \Delta^{-1} - \frac{1}{2} \cdot \delta \left(\Delta^{-1} \cdot A + A \cdot \Delta^{-1}\right) \tag{10}$$

where $\circ$ denotes the Hadamard product. In particular,

$$w_{ij} = a_{ij} \frac{\alpha\left(|\mathcal{N}_j \cap \mathcal{N}_i| + 1\right) - \delta\left(\frac{|\mathcal{N}_j \setminus \mathcal{N}_i| + |\mathcal{N}_i \setminus \mathcal{N}_j|}{2} - 1\right)}{d_i d_j} \tag{11}$$

The explicit solution of the difference equation (9) is

$$x[k] = (I + W - \text{diag}\,(W \cdot u))^k\, x[0] \tag{12}$$

where the k-th component of the initial position vector is $(x[0])_k = k$.

Proof: Appendix D.1.

Theorem 1 determines the position of the nodal vector $x[k]$ at time k and shows convergence towards a state, where the sum of attractive and repulsive forces (i.e. the resulting force) acting on a node are in balance. Nodes with similar neighbourhoods are grouped on the line, i.e. in the one-dimensional space, while nodes with a relatively small number of common neighbours are relatively far away. A possible variant of the proposed linear clustering process may map the nodal position into a higher dimensional space, like a circular disk or square in two dimensions, and even with a non-Euclidean distance metric.

C. Time-Dependence of the Linear Clustering Process

The $N \times N$ matrix $I + W - \text{diag}(W \cdot u)$ in the governing (9) has interesting properties. As shown in this section, the related matrix $W - \text{diag}(W \cdot u)$ belongs to the class of M-matrices, whose eigenvalues have a non-negative real part. The (weighted) Laplacian is another element of the M-matrix class.

Property 1: The matrix $I + W - \text{diag}(W \cdot u)$ is a non-negative matrix.

Proof: The governing (9)

$$x[k+1] = (I + W - \text{diag}\,(W \cdot u)) \cdot x[k]$$

holds for any non-negative vector $x[k]$. Let $x[0] = e_m$, the basic vector with components $(e_m)_i = \delta_{mi}$ and δ_{mi} is the Kronecker delta, then we find that the m-th column

$$x[1] = (I + W - \text{diag}\,(W \cdot u))_{col(m)}$$

must be a non-negative vector. Since we can choose m arbitrary, we have established that $I + W - \text{diag}(W \cdot u)$ is a non-negative matrix. □

Property 2: The principal eigenvector of the matrix $I + W - \text{diag}(W \cdot u)$ is the all-one vector u belonging to eigenvalue 1. All other eigenvalues of matrix $I + W - \text{diag}(W \cdot u)$ are real and, in absolute value, smaller than 1.

Proof: Appendix D.2.

The linear discrete-time system in (9) converges to a steady-state, provided that $\lim_{k\to\infty} ||x[k+1]|| = \lim_{k\to\infty} ||x[k]|| = ||x_s||$, which is only possible if the matrix $(I + W - \text{diag}(W \cdot u))$ has all eigenvalues in absolute value smaller than 1 and the largest eigenvalue is precisely equal to 1. Property 2 confirms convergence and indicates that the steady-state vector $x_s = u$ in which the position of each node is the same. However, the steady state solution $x_s = u$ is a trivial solution, as observed from the governing equation in (7), because the sum vanishes and the definition of the steady state tells that $x[k+1] = x[k]$, which is obeyed by any discrete-time independent vector. In other words, the matrix (9) can be written as

$$x[k+1] - x[k] = (W - \text{diag}\,(W \cdot u)) \cdot (x[k] - u)$$

which illustrates that, if $x[k]$ obeys the solution, then $r[k] = x[k] + s \cdot u$ for any complex number s is a solution, implying that a shift in the coordinate system of the positions does not alter the physics.

Let us denote the eigenvector y_k belonging to the k-th eigenvalue β_k of the matrix $W - \text{diag}(W \cdot u)$, where $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_N$, then the eigenvalue decomposition of the real, symmetric matrix is

$$W - \text{diag}\,(W \cdot u) = Y \text{diag}(\beta) Y^T$$

where the eigenvalue vector $\beta = (\beta_1, \beta_2, \ldots, \beta_N)$ and Y is the $N \times N$ orthogonal matrix with the eigenvectors $y_1, y_2, \ldots, y_N$ in the columns obeying $Y^T Y = Y Y^T = I$. Since $\beta_1 = 0$ and $y_1 = \frac{u}{\sqrt{N}}$, it holds for $k > 1$ that $u^T y_k = 0$, which implies that the sum of the components of eigenvectory_k for $k > 1$ is zero (just as for any weighted Laplacian [24]). The position vector in (12) is rewritten as

$$x[k] = Y \text{diag}(1+\beta)^k Y^T x[0] = \sum_{j=1}^{N} (1+\beta_j)^k y_j \left(y_j^T x[0]\right)$$

Hence, we arrive at

$$x[k] - \frac{u^T x[0]}{\sqrt{N}} u = \sum_{j=2}^{N} (1+\beta_j)^k (y_j^T x[0])\, y_j \tag{13}$$

As explained above, the left-hand side is a translated position vector and physically not decisive for the clustering process. Since $-1 < \beta_j < 0$ for $j > 1$, relation (13) indicates that, for $k \to \infty$, the right-hand side tends to zero and the steady-state solution is clearly uninteresting for the clustering process. We

rewrite (13) as

$$x[k] - \frac{u^T x[0]}{\sqrt{N}} u = (1+\beta_2)^k \left((y_2^T x[0])\, y_2 + \sum_{j=3}^{N} \left(\frac{1+\beta_j}{1+\beta_2} \right)^k (y_j^T x[0])\, y_j \right).$$

Since $|1+\beta_2| > |1+\beta_3|$, we observe that

$$\frac{x[k] - \frac{u^T x[0]}{\sqrt{N}} u}{(1+\beta_2)^k\, (y_2^T x[0])} = y_2 + O\left(\frac{1+\beta_3}{1+\beta_2} \right)^k, \qquad (14)$$

which tells us that the left-hand side, which is a normalized or scaled, shifted position vector, tends to the second eigenvector y_2 with an error that exponentially decreases in k. Hence, for large enough k, but not too large k, the scaled shifted position vector provides us the information on which we will cluster the graph.

The steady state in Property 2 can be regarded as a reference position of the nodes and does not affect the LCP process nor the $N \times 1$ eigenvector y_2, belonging to the second largest eigenvalue $(1+\beta_2)$ of the $N \times N$ "operator" matrix $I + W - \text{diag}(W \cdot u)$, which is analogous to Fiedler clustering based on the $N \times N$ Laplacian Q. While the Laplacian matrix Q essentially describes diffusion and not clustering, our operator $I + W - \text{diag}(W \cdot u)$ changes the nodal positions, based on attraction and repulsion, from which clustering naturally arises.

Property 3: The two parameters in the matrix W in (10) satisfy the bounds

$$0 \le \alpha \le \frac{d_{\max} - 1}{d_{\max} - \frac{1}{2}\left(1 + \frac{d_{\min}}{d_{\max}}\right)} \le 1 \qquad (15)$$

$$0 \le \delta \le \frac{1}{d_{\max} - \frac{1}{2}\left(1 + \frac{d_{\min}}{d_{\max}}\right)} \qquad (16)$$

Proof: Appendix D.3.

The influence of the attraction strength α and the repulsion strength δ on the eigenvalues β_k and the $N \times 1$ eigenvector y_2 of the $N \times N$ matrix W is analysed in Appendix E.

IV. From the Eigenvector y_2 to Clusters in the Network

The interplay of the attractive and repulsive force between nodes drives the nodal position in discrete time k eventually towards a steady state $\lim_{k\to\infty} x[k] = u$. However, the scaled and shifted position vector $x[k]$ in (14) converges in time towards the second eigenvector y_2 with an exponentially decreasing error. In this section, we estimate the clusters in network, based on the eigenvector y_2.

By sorting the eigenvector y_2 to $\hat{y_2}$, the components of y_2 are reordered and the corresponding relabeling of the nodes of the network reveals a block diagonal structure of the adjacency matrix A. We define the $N \times N$ permutation matrix R in a way the following equalities hold:

$$\hat{y_2} = R \cdot y_2,$$
$$(\hat{y_2})_i = (y_2)_{r_i} \le (\hat{y_2})_j = (y_2)_{r_j}, \qquad i < j, \qquad (17)$$

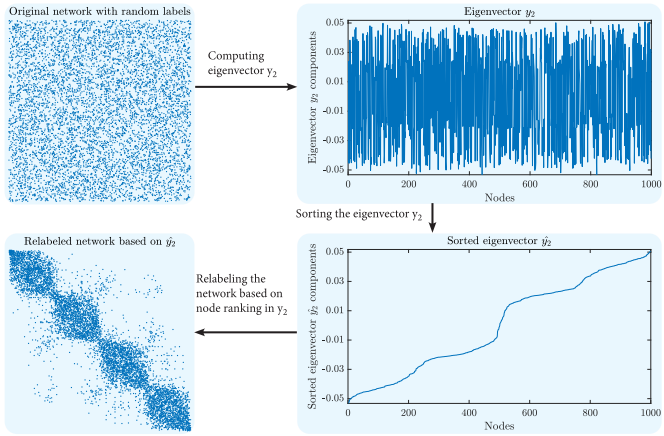


Fig. 2. Adjacency matrix A of an SSBM network of $N = 1000$ nodes, $c = 4$ clusters and parameters $b_{in} = 26$, $b_{out} = 0.67$ (top-left). Eigenvector y_2 components (top-right). Sorted eigenvector $\hat{y_2}$ components (bottom-right). Relabeled adjacency matrix $\hat{A}$ based on the sorted eigenvector $\hat{y_2}$ (bottom-left).

where the $N \times 1$ ranking vector $r = R \cdot w$ and $w = [1, 2, \ldots, N]$, with r_i denoting the node i ranking in the eigenvector y_2. The permutation matrix R allow us to define the $N \times N$ relabeled adjacency matrix $\hat{A}$, the $N \times 1$ relabeled degree vector $\hat{d}$ of G, and the $N \times 1$ sorted eigenvector $\hat{y_2}$ as follows:

$$\begin{cases} \hat{A} &= R^T \cdot A \cdot R \\ \hat{d} &= R \cdot d \\ \hat{y_2} &= R \cdot y_2. \end{cases} \qquad (18)$$

Groups of nodes that have relatively small difference in the eigenvector y_2 components, while relatively large difference compared to other nodes in the network, compose a cluster. Therefore, the community detection problem transforms into recognizing intervals of similar values in the sorted eigenvector $\hat{y_2}$.

Fig. 2 exemplifies the idea, where the adjacency matrix A of a randomly labeled SSBM network of $N = 1000$ nodes and $c = 4$ clusters is presented in the upper-left part, as a heat map. The eigenvector y_2 is drawn in the upper-right part, while the sorted eigenvector $\hat{y_2}$ is drawn on the bottom-right side. Finally, the relabeled adjacency matrix $\hat{A}$, based on nodal ranking of y_2 is depicted on the lower-left side. The sorted eigenvector $\hat{y_2}$ reveals a stair with four segments, equivalent to four block matrices on the main diagonal in relabeled adjacency matrix $\hat{A}$.

The eigenvector y_2 represents a continuous measure of how similar neighbours of two nodes are. There are two different approaches to identify network communities for a given eigenvector y_2:

- Cluster identification based on the sorted eigenvector $\hat{y_2}$. This approach is explained in Section IV-A.
- Cluster identification based on the ranking vector r. This approach does not rely on the eigenvector y_2 components, but solely on nodal ranking, as explained in Section IV-B.

A. Community Detection Based on Nodal Components of the Eigenvector y_2

To identify clusters, we observe the difference in eigenvector y_2 components between nodes with adjacent ranking. If $(\hat{y_2})_{i+1} - (\hat{y_2})_i < \theta$, where θ denotes a predefined threshold, then the nodes r_i and r_{i+1} belong to the same cluster, else the nodes r_i and r_{i+1} are boundaries of two adjacent clusters. The resulting cluster membership function is

$$C_{r_{i+1},r_i} = \begin{cases} 1 & (\hat{y_2})_{i+1} - (\hat{y_2})_i < \theta \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

where the threshold value θ is determined heuristically. The cluster estimation in (19) can be improved by using other more advanced approaches, such as the K-means algorithm.

B. Modularity-Based Community Detection

By implementing (4) and (18) into (3) we obtain:

$$m = \frac{1}{2\,L} \cdot u^T \cdot \left(\hat{A} \circ \hat{C} - \frac{1}{2\,L} \cdot \left(\hat{d} \cdot \hat{d}^T \right) \circ \hat{C} \right) \cdot u, \tag{20}$$

where $\hat{C} = R^T \cdot C \cdot R$. As shown in Fig. 2, the network relabeling based on the ranking vector r reveals block diagonal structure in $\hat{A}$. Thus, the relabeled modularity matrix $\hat{C}$ has the following block diagonal structure:

$$\hat{C} = \begin{bmatrix} J_{n_1 \times n_1} & O_{n_1 \times n_2} & \dots & O_{n_1 \times n_c} \\ O_{n_2 \times n_1} & J_{n_2 \times n_2} & \dots & O_{n_1 \times n_c} \\ \vdots & \vdots & \dots & \vdots \\ O_{n_1 \times n_1} & O_{n_c \times n_2} & \dots & J_{n_c \times n_c} \end{bmatrix}, \tag{21}$$

where c denotes number of clusters in network, where the i-th cluster is composed of n_i nodes. We highlight that relation (21) holds only in the case of a classical community problem, i.e. when each node belongs to exactly one community. We define the $N \times 1$ vectors $\hat{e_i}$ for $i = \{1, 2, \dots, c\}$ as

$$\hat{e_i} = \begin{bmatrix} O_{(1 \times \sum_{j=1}^{i-1} n_j)} & u_{(1 \times n_i)} & O_{(1 \times \sum_{j=i+1}^{N} n_j)} \end{bmatrix}^T, \tag{22}$$

that allows us to redefine $\hat{C} = \sum_{i=1}^{c} \hat{e_i} \cdot \hat{e_i}^T$ and further simplify (20):

$$m = \frac{1}{2\,L} \cdot \sum_{i=1}^{c} \hat{e_i}^T \cdot \left(\hat{A} - \frac{1}{2\,L} \cdot \left(\hat{d} \cdot \hat{d}^T \right) \right) \cdot \hat{e_i}. \tag{23}$$

Since the vector $\hat{e_i}$ consists of zeros and ones, the (23) represents the sum of elements of the matrix $(\hat{A} - \frac{1}{2L} \cdot (\hat{d} \cdot \hat{d}^T))$ corresponding to each individual cluster.

We estimate clusters for a given ranking vector r by optimising the modularity m recursively. In the first iteration, we examine all possible partitions of the network in two clusters and compute their modularity. The partition that generates the highest modularity is chosen. In the second iteration, we repeat for each subgraph the same procedure and find the best partitions into two clusters. Once we determine the best partitions for both subgraphs, we adopt them if the obtained modularity of the generated partition exceeds the modularity of a parent cluster from the previous iteration. The recursive procedure stops when the modularity m cannot be further improved, as described by pseudocode (2), provided in Appendix F. This version of the proposed process is denoted as LCP in Section VI.

C. Modularity-Based Community Detection for a Known Number of Communities

The Algorithm 2 also applies for graph partition with a known number of communities c. In that case, instead of stopping the recursive procedure described in Algorithm 2 when the modularity m cannot be further improved, we stop at iteration $(\log_2 c + 1)$. In each iteration, the partition with the maximum modularity is accepted, even if negative.

As a result, we obtain $2c$ estimated clusters with the $2c \times 2c$ aggregated modularity matrix M_c:

$$(M_c)_{gh} = \sum_{i \in g, j \in h} \left(\hat{A} - \frac{1}{2\,L} \cdot \hat{d} \cdot \hat{d}^T \right)_{ij}, \tag{24}$$

where $g, h \in \{1, 2, \dots, 2c\}$ denote estimated communities. The aggregated modularity matrix M_c allows us to merge adjacent clusters, until we reach c communities in an iterative way. We observe the $(2c - 1 \times 1)$ vector μ, where $\mu_g = (M_c)_{g,g+1}$. The maximum element of μ indicates which two adjacent clusters can be merged, so that modularity index m is negatively affected the least. By repeating this procedure c times, we end up with the graph partition in c clusters. This version of the proposed process is denoted as LCP_c in Section VI.

D. Non-Back Tracking Method Versus LCP

Angel et al. [32, p. 12] noted that the $2\,N$ non-trivial eigenvalues of the $2L \times 2\,L$ non-back tracking matrix B from (36) are contained in eigenvalues of the $2N \times 2\,N$ matrix B^*:

$$B^* = \begin{bmatrix} A & I - \Delta \\ I & O \end{bmatrix}, \tag{25}$$

where the $N \times N$ matrix with all zeros is denoted as O. The $2N \times 2\,N$ matrix B^*, written as

$$B^* = \begin{bmatrix} I + (A - \Delta) + (\Delta - I) & -(\Delta - I) \\ I & O \end{bmatrix}$$

can be considered as a state-space matrix of a process on a network, similar to our LCP process in (7), with the last N states storing delayed values of the first N states. The $2N \times 2\,N$ matrix B^* defines the set of N second-order difference equations, where the governing equation for the node i position is

$$x_i[k+1] = x_i[k] + \sum_{j \in \mathcal{N}_i} (x_j[k] - x_i[k]) + (d_i - 1) \cdot (x_i[k] - x_i[k-1]) \tag{26}$$

We recognize the second term in (26) as an attraction force between neighbouring nodes with uniform intensity, while in our LCP (7) the attraction force intensity is proportional to the number of neighbours two adjacent nodes share. Further, while we propose a repulsive force between adjacent nodes in (7), node

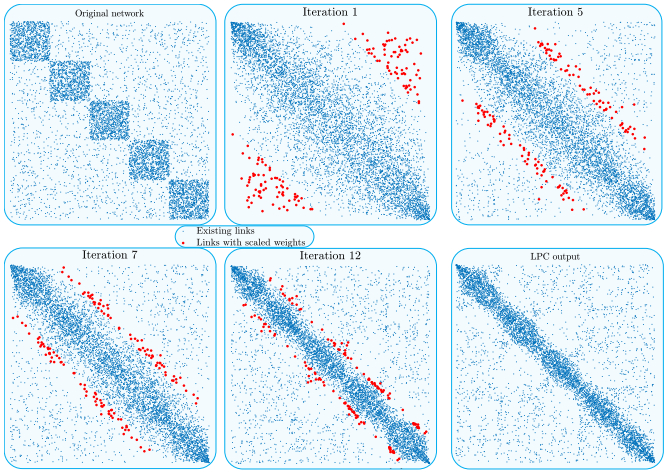


Fig. 3. Adjacency matrix A of an SSBM network of $N = 1000$ nodes, $c = 5$ clusters of equal size, with parameters $b_{in} = 26$ and $b_{out} = 2.25$ (top-left). Following 4 subfigures present the relabeled adjacency matrix based on the ranking vector r in iterations 1,5,7 and 12, respectively. In each iteration, the weights of 2% links are scaled (red colour). The weight of each link is allowed to be scaled once. The relabelled adjacency matrix $\hat{A}$ after 15 iterations of scaling weights of links between clusters (bottom-right).

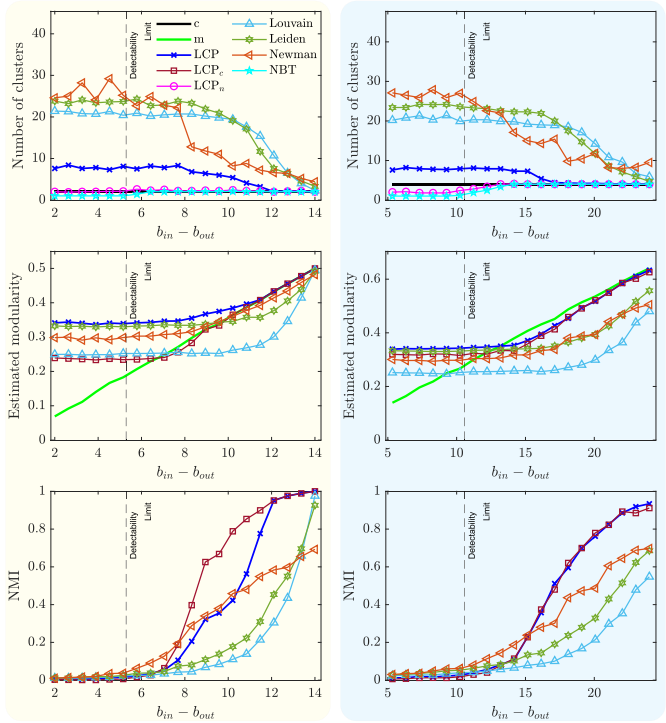


Fig. 4. The estimated number of clusters (upper figures) in SSBM graphs with $N = 1000$ nodes, average degree $d_{av} = 7$, $c = 2$ (left-hand side) and $c = 4$ (right-hand side) clusters, respectively, for different values of parameters b_{in} and b_{out}. The modularity of the estimated partitions is presented in the central figures, while the NMI measure per each clustering algorithm is provided at the bottom figures. The vertical dashed line indicates the clustering detectability threshold.

i in (26) is repulsed from its previous position $x_i[k]$ in direction of the last position change $(x_i[k] - x_i[k-1])$.

We implement the weighted intensity of the attractive force as in (7), ignoring the repulsive force by letting $\delta = 0$, and define the $2N \times 2N$ matrix W^*, corresponding to B^*, (27) shown at the bottom of this page. We estimate the number of clusters c in a network from W^* similarly as in the non-back tracking method in Section A.4 by counting the number of eigenvalues in W^* with real component larger than $\sqrt{\lambda_1(W^*)}$. This approach is denoted as LCP$_n$ in Section VI.

V. Reducing Intensity of Forces Between Clusters

The idea behind a group of methods in community detection, called divisive algorithms, consists of determining the links between nodes from different clusters. Once these links have been identified, they are removed and thus only the intra-community links remain [30]. We invoke a similar idea to our linear clustering process.

An outstanding property of our approach is that the LCP defines the nodal position as a metric, allowing us to perform clustering in multiple ways. The position distance between any two, not necessarily adjacent nodes indicates how likely the two nodes belong to the same cluster. Then, the position metric also allows us to classify links as either intra- or inter-community. Thus, we iterate the linear clustering process (7) and, in each iteration, we identify and scale the weights of the inter-community links.

The attraction and repulsive forces are defined as linear functions of the position difference between two neighbouring nodes, as presented in Fig. 1. While linear functions greatly simplify the complexity and enable a rigorous analysis, the linearity of forces introduces some difficulties in the process. Firstly, as two adjacent nodes are further away, both the attractive and the repulsive force between them increase in intensity. Similarly, as the neighbouring nodes are closer on a line, both forces decrease in intensity and converge to zero as the nodes converge to the same position. Secondly, the attractive force between any two neighbouring nodes is always of higher intensity than the repulsive force, causing the process to converge towards the trivial steady-state.

Non-linearity in the forces can be introduced in the proposed linear clustering process iteratively by scaling the weights of inter-community links between iterations, that artificially decreases the strength of forces between the two nodes from different clusters. In other words, we reduce the importance of links between nodes from different clusters, based on the partition from previous iteration.

A. Scaling the Weights of Inter-Community Links

The difference $|(y_2)_i - (y_2)_j|$ in the eigenvector y_2 components of nodes i and j indicates how similar neighbourhoods

$$W^* = \begin{bmatrix} I + \alpha \cdot \left(A \circ A^2 + A - \text{diag}\left(\left(A \circ A^2 + A\right) \cdot u\right)\right) + (\Delta - I) & -(\Delta - I) \\ I & O \end{bmatrix}. \tag{27}$$

of these nodes are. A normalized measure for the difference in neighbouring nodes i and j is the difference $(|r_i - r_j|)$ of their rankings in the sorted eigenvector $\hat{y_2}$. Thus, links that connect nodes with the highest ranking difference are most likely inter-community links. We define the $N \times N$ scaling matrix S as follows:

$$s_{ij} = \begin{cases} 1, & \text{if } |r_j - r_i| < \theta_r \\ \upsilon, & \text{otherwise}, \end{cases} \tag{28}$$

where the ij-th element equals 1 if the absolute value of the ranking difference between nodes i and j is below a threshold θ_r, otherwise some positive value $0 \leq \upsilon \leq 1$. Based on the $N \times N$ scaling matrix S in (28), we update the governing equation as follows:

$$x[k+1] = \left(I + \tilde{W} - \text{diag}\left(\tilde{W} \cdot u\right)\right) \cdot x[k],$$

where $\tilde{W} = S \circ W$. Scaling the link weights in (28) only impacts the clustering process in (9), as defined in the equation above. However, modularity-based community detection, explained in Section IV-B, operates on the $N \times N$ adjacency matrix A in each iteration. Therefore, our implementation of scaling the weights of inter-community connections in network helps the process to better distinguish between clusters (i.e. eventually provides better relabeling in (18)), without modifying the $N \times N$ adjacency matrix A and, hence, without negatively affecting the modularity m optimisation in Algorithm 2. An example of removing links (i.e. $\upsilon = 0$) is depicted on Fig. 3, where in each iteration weights of $\frac{15}{4}\%$ identified inter-cluster links are scaled. Scaling the weights of links between clusters significantly improves the quality of the identified graph partition.

VI. Benchmarking LCP With Other Clustering Methods

Computational complexity of the entire proposed clustering process equals $O(N \cdot L)$, as derived in Appendix G. In this section, we benchmark the linear clustering process (7) against popular clustering algorithms (introduced in Appendix A), both on synthetic and real-world networks. The non-back tracking algorithm (Appendix A.4) and our LCP$_n$ (Section IV-D) estimate only number of clusters, Newman's method (Appendix A.3), the Leiden method (Appendix A.2) the Louvain method (Appendix A.1) and our LCP (Section IV-B) estimate both number of clusters and the cluster membership of each node, while LCP$_c$ (Section IV-C) requires the number of communities c to perform graph partitioning. The attractive strength $\alpha = 0.95$ and the repulsive strength $\delta = 10^{-3}$ are used in all simulations. Weights of 60% links in total are scaled using (28), evenly over 30 iterations, where in i-th iteration scaled weight is $\frac{0.05 \cdot i}{30}$.

A. Clustering Performances on Stochastic Block Generated Graphs

We compare the clustering performance of our LCP with that of clustering methods introduced in Appendix A, on a same graph generated by the symmetric stochastic block model (SSBM) with clusters of equal size. All graphs have $N = 1000$ nodes. We vary the parameters b_{in} and b_{out} using (37) in a way to keep the average degree $d_{av} = 7$ fixed. For each SSBM network, we execute the clustering methods 10^2 times and present the mean number of estimated clusters and mean modularity of produced partitions in Figs. 4, 5.

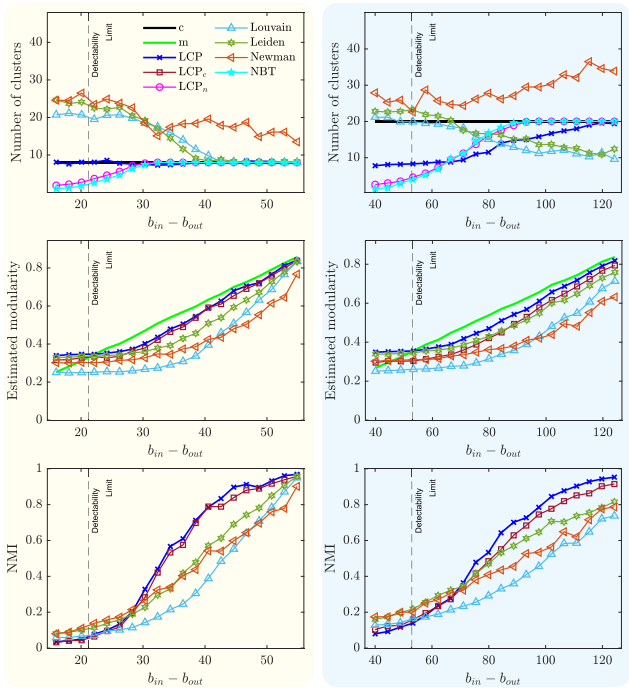


Fig. 5. The estimated number of clusters (upper figures) in SSBM graphs with $N = 1000$ nodes, average degree $d_{av} = 7$, $c = 8$ (left-hand side) and $c = 20$ (right-hand side) clusters, respectively, for different values of parameters b_{in} and b_{out}. The modularity of the estimated partitions is presented in the central figures, while the NMI measure per each clustering algorithm is provided at the bottom figures. The vertical dashed line indicates the clustering detectability threshold.

The clustering performance on SSBM graphs with $c = 2$ clusters ($c = 4$ clusters) is presented on the left-hand side (right-hand side) of Fig. 4, respectively. The non-back tracking algorithm and our LCP$_n$ achieve the best performance in estimating the number of communities c, as shown in the upper part of Fig. 4. Further, our LCP outperforms each considered modularity-based method in identifying the number of communities c and in modularity m. Furthermore, when clusters are visible (i.e. above the detectability threshold), the NMI value (presented in the bottom figures) of our LCP and our LCP$_c$ significantly outperforms other clustering algorithms. Fig. 4 illustrates a significant difference in performance between our LCP and the non-back tracking matrix (NBT) method. Our LCP (in blue) and the other three modularity-based methods perform poorly in recognising the number c of clusters for a wide range of $b_{in} - b_{out}$ (around and below the detectability threshold). Poor performance occurs because modularity-based methods generate partitions of higher modularity than the original network (in black) but with different communities! Consequently, the NMI measure deteriorates in these regimes. Our LCP$_n$ (in red), for a given number of communities c, identifies partitions with

TABLE I
CLUSTERING PERFORMANCE OF OUR LCP AND CONSIDERED EXISTING CLUSTERING ALGORITHMS ON REAL-WORLD NETWORKS

Real-world networks			LCP		Louvain		Leiden		Newman		NBT	LCP_c
Network name	N	L	c	m	c	m	c	m	c	m	c	c
Karate Club	34	78	3	**0.3922**	4	0.3565	4	0.3729	5	0.3776	2	1
Dolphins	62	159	4	0.5057	4	0.4536	5	**0.5105**	6	0.4894	2	2
Polbooks	105	441	3	**0.5160**	4	0.4897	4	0.5026	8	0.4160	3	2
Football	115	613	7	**0.5894**	7	0.5442	7	0.5635	11	0.4623	10	5
Facebook	347	2519	8	**0.4089**	16	0.3726	18	0.3792	23	0.3770	8	4
Polblogs	1490	19090	19	**0.4224**	7	0.3385	11	0.3117	4	0.3459	8	1
Co-autorship	1589	2742	40	0.9296	272	**0.9423**	270	0.9410	28	0.7393	23	16

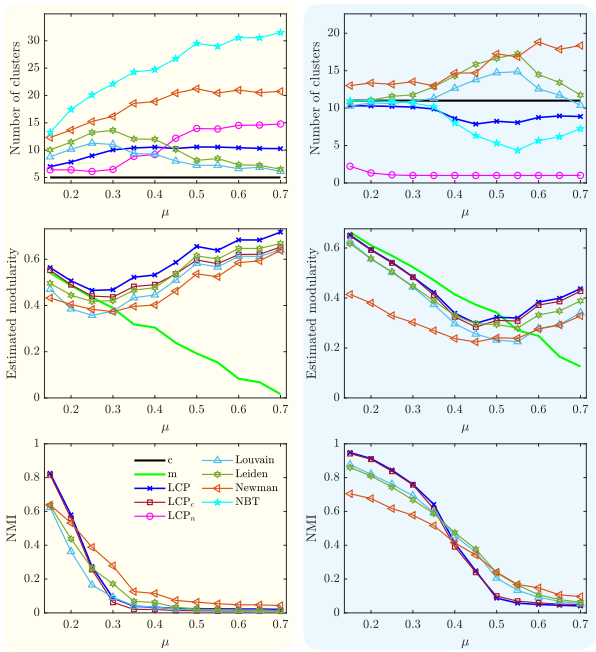


Fig. 6. The estimated number of clusters (upper figures) in LFR benchmark graphs of $N = 500$ nodes with the average degree $d_{av} = 12$, consisting of $c = 5$ (left-hand side, with $\gamma = 1$ and $\beta = 2$) and $c = 11$ (right-hand side with $\gamma = 2$ and $\beta = 3$) clusters, respectively, for different values of parameter μ. The modularity of the estimated partitions is presented in the central figures, while the NMI measure per each clustering algorithm is provided at the bottom figures.

higher modularity m than of the original network, even within the theoretically detectable regime.

Fig. 5 illustrates results for SSBM graphs of $N = 1000$ nodes, with $c = 8$ (left-hand side) and $c = 20$ (right-hand side) clusters. Our LCP consistently outperforms the other three methods in estimated modularity m over the entire range of $b_{in} - b_{out}$ values. Except for $b_{in} - b_{out}$ values around and below the detectability treshold, the NMI measure of our LCP is superior to other three methods (bottom figures).

B. Clustering Performances on LFR Benchmark Graphs

Fig. 6 illustrates clustering results on LFR benchmark graphs of $N = 500$ nodes with $c = 5$ (left-hand part) and $c = 11$ (right-hand part) communities. Compared to Newman, Louvain and Leiden algorithm, our LCP is among the best in estimating the number of clusters c (upper figures) while outperforming each considered method in estimated modularity m (middle figures). In addition, our LCP provides the highest NMI measure when the clusters are visible (i.e. for low μ value). For relatively large values of μ, our LCP identifies partitions different from the original one but with considerably higher modularity. Therefore, the NMI measure deteriorates in this regime (lower figures). When a graph is generated by the LFR benchmark, the non-backtracking method (NBT) and our LCP_n fail to estimate the number of clusters c.

C. Clustering Performances on Real-World Networks

Table I summarises the clustering performance of our LCP and those considered existing algorithms on seven real-world networks of different sizes, number of links and community structure. In five out of seven cases, our LCP provides partition with the highest modularity m, compared to other algorithms. LCP's superiority in achieved modularity m aligns with the results obtained on synthetic benchmarks. While the estimated number of clusters c of each method cannot be judged as the ground truth is unknown, LCP's estimated number of communities c is, on average, the closest to that of the non-back tracking matrix, known as one of the best predictors in the literature.

VII. CONCLUSION

In this paper, we propose a linear clustering process (LCP) on a network consisting of an attraction and repulsion process between neighbouring nodes, proportional to how similar or different their neighbours are. Based on nodal positions, we are able to estimate both the number c and the nodal membership of communities. Our LCP outperforms modularity-based clustering algorithms, such as Newman's, Leiden and the Louvain method on both synthetic and real-world networks, while being of the same computational complexity. The proposed LCP allows estimating the number c of clusters as accurately as the non-back tracking matrix, in case of SSBM graphs. A potential improvement of the proposed linear clustering process lies in a more effective way of scaling inter-community link weights between successive iterations.

The linear clustering process LCP is described by a matrix $I + W - \text{diag}(W \cdot u)$, which can be regarded as an operator acting on the position of nodes, comparable to quantum mechanics (QM). In QM, an operator describes a dynamical action on a set of particles. Since quantum mechanical operators are linear, the dynamics are exactly computed via spectral decomposition. In a same vein, our operator $I + W - \text{diag}(W \cdot u)$ is linear and describes via attraction and repulsion a most likely ordering of the position of nodes that naturally leads to clusters, via

spectral decomposition, in particular, via the eigenvector y_2 in Section III-C.

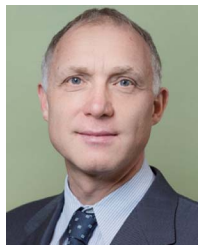
ACKNOWLEDGMENT

The authors are grateful to S. Fortunato for useful comments. This research is part of NExTWORKx, a collaboration between TU Delft and KPN on future communication networks.

REFERENCES

[1] A.-L. Barabási, "Network science," *Philos. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 371, no. 1987, 2013, Art. no. 20120375.
[2] M. Newman, *Networks*. Oxford, U.K.: Oxford Univ. Press, 2018.
[3] B. Karrer, M. E. Newman, and L. Zdeborová, "Percolation on sparse networks," *Phys. Rev. Lett.*, vol. 113, no. 20, 2014, Art. no. 208702.
[4] B. Prasse and P. V. Mieghem, "The viral state dynamics of the discrete-time NIMFA epidemic model," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1667–1674, Jul.–Sep. 2020.
[5] R. Pastor-Satorras, C. Castellano, P. V. Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Rev. Modern Phys.*, vol. 87, no. 3, pp. 925–979, 2015.
[6] A. D. Sánchez, J. M. López, and M. A. Rodriguez, "Nonequilibrium phase transitions in directed small-world networks," *Phys. Rev. Lett.*, vol. 88, no. 4, 2002, Art. no. 048701.
[7] H. E. Stanley, *Phase Transitions and Critical Phenomena*, vol. 7. Oxford, U.K.: Clarendon Press, 1971.
[8] I. Jokić and P. V. Mieghem, "Linear processes on complex networks," *J. Complex Netw.*, vol. 8, no. 4, 2020, Art. no. cnaa030.
[9] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3-5, pp. 75–174, 2010.
[10] G. Budel and P. V. Mieghem, "Detecting the number of clusters in a network," *J. Complex Netw.*, vol. 8, no. 6, Dec. 1, 2020, Art. no. cnaa047.
[11] M. E. Newman, "Communities, modules and large-scale structure in networks," *Nature Phys.*, vol. 8, no. 1, pp. 25–31, 2012.
[12] M. E. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006.
[13] X. Xu, N. Yuruk, Z. Feng, and T. A. Schweiger, "Scan: A structural clustering algorithm for networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2007, pp. 824–833.
[14] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mechanics: Theory Experiment*, vol. 2008, no. 10, 2008, Art. no. P10008.
[15] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E*, vol. 80, no. 5, 2009, Art. no. 056117.
[16] T. P. Peixoto, "Hierarchical block structures and high-resolution model selection in large networks," *Phys. Rev. X*, vol. 4, no. 1, 2014, Art. no. 011047.
[17] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *J. Stat. Mechanics: Theory Experiment*, vol. 2005, no. 09, 2005, Art. no. P09008.
[18] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Proc. Comput. Inf. Sci.20th Int. Symp.*, Istanbul, Turkey, 2005, vol. 20, pp. 284–293.
[19] R. Lambiotte, R. Sinatra, J.-C. Delvenne, T. S. Evans, M. Barahona, and V. Latora, "Flow graphs: Interweaving dynamics and structure," *Phys. Rev. E*, vol. 84, no. 1, 2011, Art. no. 017102.
[20] A. Arenas, A. Diaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," *Phys. Rev. Lett.*, vol. 96, no. 11, 2006, Art. no. 114102.
[21] D. Jin et al., "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1149–1170, Feb. 2023.
[22] F. Krzakala et al., "Spectral redemption in clustering sparse networks," *Proc. Nat. Acad. Sci.*, vol. 110, no. 52, pp. 20935–20940, 2013.
[23] P. Van Mieghem, *Graph Spectra for Complex Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
[24] P. Van Mieghem, K. Devriendt, and H. Cetinay, "Pseudoinverse of the Laplacian and best spreader node in a network," *Phys. Rev. E*, vol. 96, no. 3, 2017, Art. no. 032311.
[25] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004, Art. no. 026113.
[26] U. Brandes et al., "On modularity clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 2, pp. 172–188, Feb. 2008.
[27] P. Van Mieghem, X. Ge, P. Schumm, S. Trajanovski, and H. Wang, "Spectral graph analysis of modularity and assortativity," *Phys. Rev. E*, vol. 82, no. 5, 2010, Art. no. 056113.
[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge Univ. Press, 2012.
[29] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, 1983.
[30] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci.*, vol. 99, no. 12, pp. 7821–7826, 2002.
[31] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, no. 4, 2008, Art. no. 046110.
[32] O. Angel, J. Friedman, and S. Hoory, "The non-backtracking spectrum of the universal cover of a graph," *Trans. Amer. Math. Soc.*, vol. 367, no. 6, pp. 4287–4318, 2015.

Ivan Jokić received the B.Sc. degree in energetics and control theory and the M.Sc. degree in control theory from the University of Montenegro, Podgorica, Montenegro, in 2015 and 2018, respectively. He is currently working toward the Ph.D. degree since February 2019 from the Delft University of Technology, The Netherlands. His main research interests include graph theory, network dynamics, systems theory, and networked systems identification.



Piet Van Mieghem received the Master degree (*magna cum laude*) and the Ph.D. degree (*summa cum laude* with Congratulations) in electrical engineering from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1987 and 1991, respectively. Before joining Delft, he was with the Interuniversity Micro Electronic Center (IMEC) from 1987 to 1991. He is currently a Professor with the Delft University of Technology, Delft, The Netherlands, with a Chair in telecommunication networks and Chairman of the section Network Architectures and Services (NAS) since 1998. . He is a board Member of the Netherlands Platform of Complex Systems, the Steering Committee Member of the Dutch Network Science Society, an External Faculty Member with the Institute for Advanced Study (IAS), University of Amsterdam, Amsterdam, The Netherlands. During 1993–1998, he was a Member of the Alcatel Corporate Research Center in Antwerp, where he was engaged in performance analysis of ATM systems and in network architectural concepts of both ATM networks (PNNI) and the Internet. During 1992–1993, he was a Visiting Scientist with the Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, and a Visiting Professor with the Department of Electrical Engineering, University of California, Los Angeles, Los Angeles, CA, USA, in 2005, Center of Applied Mathematics, Cornell University, Ithaca, NY, USA, in 2009, Department of Electrical Engineering, Stanford University, Stanford, CA, in 2015, and with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, in 2022. He is the author of four books: *Performance Analysis of Communications Networks and Systems*, *Data Communications Networking*, *Graph Spectra for Complex Networks* and *Performance Analysis of Complex Networks and Systems.* The focus of his chair is broadened from telecommunication networks to network science. His main research interests include the modelling and analysis of complex networks, such as infrastructural, biological, brain, social networks and in new Internet-like architectures and algorithms for future communications networks. He is currently on the Editorial Board of the *OUP Journal of Complex Networks*. He was a Member of the Editorial Board of Computer Networks during 2005–2006, the IEEE/ACM TRANSACTIONS ON NETWORKING during 2008–2012, the *Journal of Discrete Mathematics* during 2012–2014, and *Computer Communications* during 2012–2015. He was awarded an Advanced ERC grant 2020 for ViSiON, Virus Spread in Networks.