Non-backtracking cycles: length spectrum theory and graph mining applications

Leo Torres[1]* ![ORCID], Pablo Suárez-Serrato[2,3] and Tina Eliassi-Rad[1,4]

*Correspondence: leo@leotrs.com
[1]Network Science Institute, Northeastern University, 360 Huntington Ave, 02115-5000, Boston, MA, USA
Full list of author information is available at the end of the article

Abstract

Graph distance and graph embedding are two fundamental tasks in graph mining. For graph distance, determining the structural dissimilarity between networks is an ill-defined problem, as there is no canonical way to compare two networks. Indeed, many of the existing approaches for network comparison differ in their heuristics, efficiency, interpretability, and theoretical soundness. Thus, having a notion of distance that is built on theoretically robust first principles and that is interpretable with respect to features ubiquitous in complex networks would allow for a meaningful comparison between different networks. For graph embedding, many of the popular methods are stochastic and depend on black-box models such as deep networks. Regardless of their high performance, this makes their results difficult to analyze which hinders their usefulness in the development of a coherent theory of complex networks. Here we rely on the theory of the *length spectrum* function from algebraic topology, and its relationship to the non-backtracking cycles of a graph, in order to introduce two new techniques: Non-Backtracking Spectral Distance (NBD) for measuring the distance between undirected, unweighted graphs, and Non-Backtracking Embedding Dimensions (NBED) for finding a graph embedding in low-dimensional space. Both techniques are interpretable in terms of features of complex networks such as presence of hubs, triangles, and communities. We showcase the ability of NBD to discriminate between networks in both real and synthetic data sets, as well as the potential of NBED to perform anomaly detection. By taking a topological interpretation of non-backtracking cycles, this work presents a novel application of topological data analysis to the study of complex networks.

Keywords: Graph distance, Graph embedding, Algebraic topology, Length spectrum
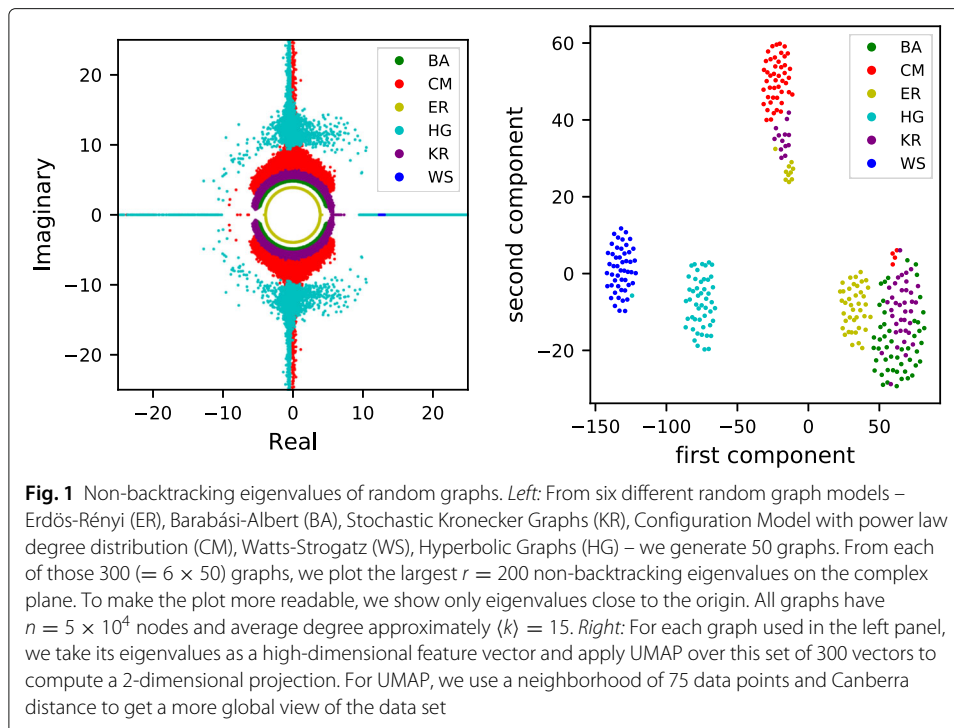
Introduction

As the network science literature continues to expand and scientists compile more examples of real life networked data sets coming from an ever growing range of domains (Clauset et al.; Kunegis 2013), there is a need to develop methods to compare complex networks, both within and across domains. Many such graph distance measures have been proposed (Soundarajan et al. 2014; Koutra et al. 2016; Bagrow and Bollt 2018; Bento and Ioannidis 2018; Onnela et al. 2012; Schieber et al. 2017; Chowdhury and Mémoli 2017; 2018; Berlingerio et al. 2013; Yaveroğlu et al. 2014), though they vary in the features they use for comparison, their interpretability in terms of structural features of complex

networks, computational costs, as well as in the discriminatory power of the resulting distance. This reflects the fact that complex networks represent a wide variety of systems whose structure and dynamics are difficult to encapsulate in a single distance score. For the purpose of providing a principled, interpretable, efficient and effective notion of distance, we turn to the *length spectrum* function. The length spectrum function can be defined on a broad class of metric spaces that includes Riemannian manifolds and graphs. The discriminatory power of the length spectrum is well known in other contexts: it can distinguish certain one-dimensional metric spaces up to isometry (Constantine and Lafont 2018), and it determines the Laplacian spectrum in the case of closed hyperbolic surfaces (Leininger et al. 2007). However, it is not clear if this discriminatory power is also present in the case of complex networks. Accordingly, we present a study on the following question: is the length spectrum function useful for the comparison of complex networks?

We answer this question in the positive by introducing the Non-Backtracking Spectral Distance (NBD): a principled, interpretable, efficient, and effective measure that quantifies the distance between two undirected, unweighted networks. NBD has several desirable properties. First, NBD is based on the theory of the length spectrum and the set of non-backtracking cycles of a graph (a non-backtracking cycle is a closed walk that does not retrace any edges immediately after traversing them); these provide the theoretical background of our method. Second, NBD is interpretable in terms of features of complex networks such as existence of hubs and triangles. This helps in the interpretation and visualization of distance scores yielded by NBD. Third, NBD is more computationally efficient than other comparable methods. Indeed, NBD depends only on the computation of a few of the largest eigenvalues of the non-backtracking matrix of a graph, which requires only slightly more computational time than the spectral decomposition of the adjacency matrix. Fourth, we have extensive empirical evidence that demonstrates the effectiveness of NBD at distinguishing real and synthetic networks.

NBD is intimately related to the eigenvalues of the non-backtracking matrix, which serve as a proxy for the length spectrum of a graph (see "Relaxed Length Spectrum" section). Motivated by the usefulness of the eigenvalues of the non-backtracking matrix, we then turn our focus to its eigenvectors. We discuss the potential of using the eigenvectors as a graph embedding technique, which we refer to as Non-Backtracking Embedding Dimensions (or NBED for short). This technique computes a low dimensional embedding of each directed edge in a graph. We show that the peculiar visualizations yielded by NBED are particularly apt at providing a rich visualization of the embedded network, and use the patterns found therein to perform anomaly detection on the Enron email corpus (Klimt and Yang 2004).

The contributions of this paper are as follows. By highlighting the topological interpretation of the non-backtracking cycles of a graph, we propose Non-Backtracking Spectral Distance (NBD) as a novel graph distance method, and discuss the potential of Non-Backtracking Embedding Dimensions (NBED) as a promising embedding technique. Along the way, we present an efficient algorithm to compute the non-backtracking matrix, and discuss the data visualization capabilities of its complex eigenvalues (see Fig. 1) and eigenvectors (Fig. 12). With this work we also release publicly available source code that implements NBD, NBED and the related matrix computations (Torres 2018).

Fig. 1 Non-backtracking eigenvalues of random graphs. *Left:* From six different random graph models – Erdös-Rényi (ER), Barabási-Albert (BA), Stochastic Kronecker Graphs (KR), Configuration Model with power law degree distribution (CM), Watts-Strogatz (WS), Hyperbolic Graphs (HG) – we generate 50 graphs. From each of those 300 ($= 6 \times 50$) graphs, we plot the largest $r = 200$ non-backtracking eigenvalues on the complex plane. To make the plot more readable, we show only eigenvalues close to the origin. All graphs have $n = 5 \times 10^4$ nodes and average degree approximately $\langle k \rangle = 15$. *Right:* For each graph used in the left panel, we take its eigenvalues as a high-dimensional feature vector and apply UMAP over this set of 300 vectors to compute a 2-dimensional projection. For UMAP, we use a neighborhood of 75 data points and Canberra distance to get a more global view of the data set

The rest of this paper is structured as follows. "Background" section provides necessary background information on the length spectrum, non-backtracking cycles, and the non-backtracking matrix. In "Related work" section we discuss previous investigations related to ours. "Operationalizing the length spectrum" section explains the connection between these objects, as well as a discussion of the properties of the non-backtracking matrix that make it relevant for the study of complex networks. It also presents our efficient algorithm to compute it. "NBD: Non-backtracking distance" section presents our distance method NBD and provides experimental evidence of its performance by comparing it to other distance techniques. In "NBED: Non-backtracking embedding dimensions" section we discuss our embedding method NBED based on the eigenvectors of the non-backtracking matrix and provide extensive visual analysis of the resulting edge embeddings, as well as mention its shortcomings and necessary future lines of study. We conclude in "Discussion and conclusions" section with a discussion of limitations and future work.

Background

Here we introduce two theoretical constructions: the length spectrum of a metric space and the set of non-backtracking cycles of a graph. Our analysis pivots on the fact that the latter is intimately related to a particular subset of the (domain of the) former. For a list of all symbols and abbreviations used in this work, see Table 1.

Length Spectrum

Consider a metric space X and a point $p \in X$. A closed curve that goes through p is called a *loop*, and p is called its *basepoint.* Two loops are *homotopic* to one another relative to a given basepoint when there exists a continuous transformation from one to the other that fixes the basepoint. The *fundamental group* of X with basepoint p is denoted by

Table 1 Notation used in this work

Symbol	Definition
$\pi_1(X, p)$	The fundamental group of X with basepoint p
$[c]$	Homotopy class of loop c
$\mathcal{L}, \mathcal{L}'$	Length spectrum function, relaxed length spectrum
$Conv(X)$	If X is a graph, $Conv(X)$ is its 2-core
$G = (V, E)$	An undirected graph with node set V and edge set E
n, m	Number of nodes and number of edges of a graph G
e, e^{-1}	A directed edge $e = u \rightarrow v$ and its inverse $e^{-1} = v \rightarrow u$
NBC	Non-backtracking cycle, in which no edge is followed by its inverse
B	$2m \times 2m$ non-backtracking matrix of a graph
B'	$2n \times 2n$ matrix whose eigenvalues are the same as those of B, save for ± 1
$\lambda_k = a_k + ib_k$	k-th largest eigenvalue, in magnitude, of B
$Re(\lambda), Im(\lambda)$	Real and imaginary parts of the complex number λ
P, Q	$n \times 2m$ directed incidence matrices of a graph
p_k	The fraction of a graphs' nodes with degree k
γ	The degree exponent in the case when $p_k \sim k^{-\gamma}$
$\langle k^i \rangle$	i-th moment of the degree distribution p_k
$\deg(u)$	Degree of node u
$nnz(A)$	Number of non-zero elements of binary matrix A
$NBD(G, H)$	Non-backtracking distance between graphs G and H
$d(\{\lambda_k\}, \{\mu_k\})$	Distance between eigenvalues λ_k and eigenvalues μ_k
r	Number of eigenvalues computed from one graph
$\rho = \sqrt{\lambda_1}$	Magnitude threshold for eigenvalue computation
r_0	Number of eigenvalues whose magnitude is greater than ρ
σ	Spread parameter of the RBF kernel

$\pi_1(X, p)$ and is defined as the first homotopy group of X, i.e., the set of all loops in X that go through p, under the equivalence relation of based homotopy. An element of the fundamental group is called a *homotopy class*. If a homotopy class contains the loop c, we denote it as $[c]$. The group operation is the product of two homotopy classes, defined as the concatenation of two of their representative loops and written multiplicatively as $[c_1][c_2]$ whenever $[c_1]$ and $[c_2]$ are elements of $\pi_1(X, p)$. Since c_1 starts at the same node at which c_2 ends, they can be traversed one after the other, which uniquely defines another based loop. A *conjugacy class* of the fundamental group is a set C of homotopy classes such that for any two classes $[c_1], [c_2] \in C$ there exists a third class $[g]$ that satisfies $[c_1] = [g^{-1}][c_2][g]$. Here, the inverse of the loop g is defined as the same loop traversed in reverse. Closed curves without a distinguished basepoint are called *free loops*, and they correspond to conjugacy classes of $\pi_1(X, p)$. For an example of conjugacy classes of the fundamental group of a graph G, see Fig. 2. A well-known fact of homotopy theory is that if X is path-connected (i.e., if any two points can be connected through a continuous curve in X) then $\pi_1(X, p)$ is unique up to isomorphism, regardless of basepoint p. In the present work we only consider connected graphs, hence, we simply write $\pi_1(X)$ when there is no ambiguity. For more on homotopy, we refer the interested reader to (Munkres 2000; Hatcher 2017).

The length spectrum $\mathcal{L}$ is a function on $\pi_1(X)$ which assigns to each homotopy class of loops the infimum length among all of the representatives in its conjugacy class.[1] Note, importantly, that the definition of length of a homotopy class considers the length of those loops not only in the homotopy class itself, but in all other conjugate classes. In the case

of compact geodesic spaces, such as finite metric graphs, this infimum is always achieved. For a finite graph where each edge has length one, the value of $\mathcal{L}$ on a homotopy class then equals the number of edges contained in the length-minimizing loop. That is, for a graph $G = (V, E)$, $v \in V$, if $[c] \in \pi_1(G, v)$ and c achieves the minimum length k in all classes conjugate to $[c]$, we define $\mathcal{L}([c]) = k$.

Our interest in the length spectrum is supported by the two following facts. First, graphs are *aspherical*. More precisely, the underlying topological space of any graph is aspherical, i.e., all of its homotopy groups of dimension greater than one are trivial.[2] Therefore, it is logical to study the only non-trivial homotopy group, the fundamental group $\pi_1(G)$. Second, Constantine and Lafont (Constantine and Lafont 2018) showed that the length spectrum of a graph determines a certain subset of it up to isomorphism. Thus, we aim to determine when two graphs are close to each other by comparing their length spectra relying on the main theorem of (Constantine and Lafont 2018). For completeness, we briefly mention it here; it is known as *marked length spectrum rigidity*.

Theorem 1 (Constantine and Lafont 2018) *For a metric space X define Conv(X) as the minimal set to which X retracts by deformation. Let X_1, X_2 be a pair of compact, non-contractible, geodesic spaces of topological dimension one. If the marked length spectra of X_1 and X_2 are the same, then Conv(X_1) is isometric to Conv(X_2).*

When G_1, G_2 are graphs, $Conv(G_i), i = 1, 2$, corresponds to the subgraph resulting from iteratively removing nodes of degree 1 from G_i, which is precisely the 2-core of G_i (Batagelj and Zaversnik 2011). Equivalently, the 2-core of G is the maximal subgraph in which all vertices have degree at least 2. Thus, Theorem 1 states that *when two graphs have the same length spectrum, their 2-cores are isomorphic*.

Given these results, it is natural to use the length spectrum as the basis of a measure of graph distance. Concretely, given two graphs, we aim to efficiently quantify how far their 2-cores are from being isomorphic by measuring the distance between their length spectra. In "Relaxed Length Spectrum" section, we explain our approach at implementing a computationally feasible solution for this problem.

Non-Backtracking Cycles

Consider an undirected, unweighted graph $G = (V, E)$ and suppose $|E| = m$. For an undirected edge $e = \{u, v\} \in E$, we write $u \to v$ and $v \to u$ for its two possible orientations, and define $\vec{E}$ as the set of all $2m$ directed edges. Given a directed edge $e = u \to v \in \vec{E}$, define e^{-1} as the same edge traversed in the inverse order, $e^{-1} = v \to u$. A *path* in G is a sequence of directed edges $e_1 e_2 ... e_k$ such that if $e_i = u_i \to v_i$ then $v_i = u_{i+1}$ for $i = 1, ..., k - 1$. Here, k is called *length* of the path. A path is *closed* if $v_k = u_1$. A path is called *non-backtracking* when an edge is never followed by its inverse, $e_{i+1} \neq e_i^{-1}$ for all i. A closed path is also called a *cycle*. A cycle is a *non-backtracking cycle* (or NBC for short) when it is a closed non-backtracking path and, in addition, $e_k \neq e_1^{-1}$.[3]

The *non-backtracking matrix B* is a $2m \times 2m$ matrix whose rows and columns are indexed by directed edges. For two edges $u \to v$ and $k \to l$, B is given by
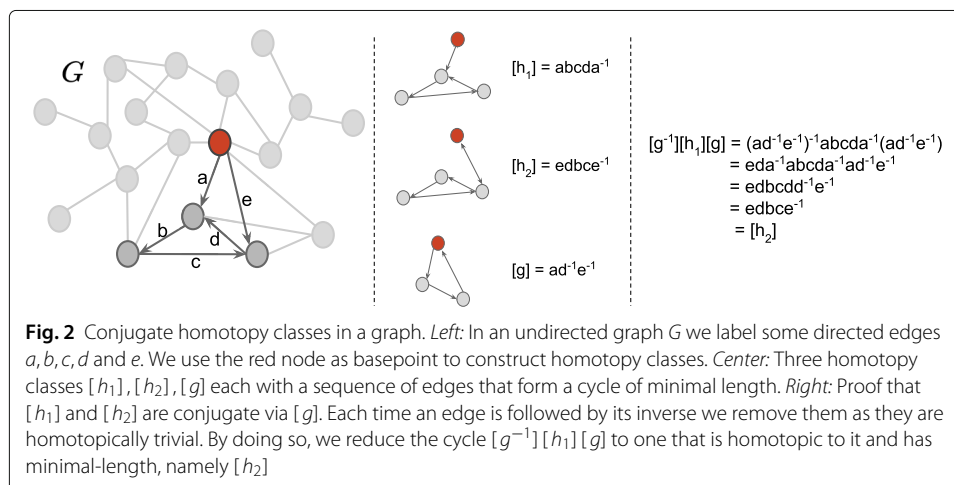
$$B_{k \to l, i \to j} = \delta_{jk}(1 - \delta_{il}), \tag{1}$$

where δ_{uv} is the Kronecker delta. Thus, there is a 1 in the entry indexed by row $k \to l$ and column $u \to v$ when $u \neq l$ and $v = k$, and a 0 otherwise. Intuitively, one can interpret the B matrix as the (unnormalized) transition matrix of a random walker that does not perform backtracks: the entry at row $k \to l$ and column $u \to v$ is positive if and only if a walker can move from node u to node v (which equals node k) and then to l, without going back to u.

Importantly, NBCs are topologically relevant because backtracking edges are homotopically trivial, that is, the length-2 cycle $u \to v, v \to u$ can always be contracted to a point (Terras 2011). Indeed, in Fig. 2 we removed edges are precisely those that form backtracks.

Observe that the matrix B tracks each pair of incident edges that do not comprise a backtrack. As a consequence we have the following result.

Lemma 1 $B^p_{k \to l, i \to j}$ *equals the number of non-backtracking paths of length* $p + 1$ *that start with edge* $i \to j$ *and end with edge* $k \to l$. *Moreover,* $tr(B^p)$ *is proportional to the number of closed non-backtracking paths (i.e., NBCs) of length* p *in G.*

Proof 1 *The first fact is reminiscent of the well-known fact that* A^p_{uv} *gives the number of paths that start at node u and end at node v, for any* $u, v \in G$, *where A is the adjacency matrix of G. The proof is the same as the proof of this latter fact, so we omit the details for brevity. The only difference is that* B^p *gives the number of non-backtracking paths of length* $p + 1$, *while* A^p *gives the number of paths of length p. To see why, notice that if the entry* $B_{k \to l, i \to j}$ *is positive, then the length-2 path* $i \to j \to l$ *must exist and it cannot contain any backtracks. Accordingly, if* $B^2_{k \to l, i \to j}$ *is positive then the length-3 path* $i \to j \to k \to l$ *must be non-backtracking. It remains to show that* $tr(B^p)$ *is proportional to the number of NBCs of length k, and not the number of NBCs of length* $p + 1$ *as one may think from the previous paragraph. Suppose* $a \to b \to c \to a$ *is a valid closed non-backtracking path, that is, the nodes a, b, c form a triangle. In this case,* $B^2_{c \to a, a \to b}$ *must be positive, as must be* $B^3_{a \to b, a \to b}$. *The latter represents the existence of some node c such that the path* $a \to b \to c \to a \to b$ *exists and is non-backtracking, or, equivalently, that* $a \to b \to c \to a$ *is a non-backtracking cycle. Note that* $B^3_{b \to c, b \to c}$ *and* $B^3_{c \to a, c \to a}$ *are also positive and all signal the existence of the same triangle a, b, c. Thus the sum of all diagonal elements* $tr(B^3)$

Fig. 2 Conjugate homotopy classes in a graph. *Left:* In an undirected graph *G* we label some directed edges a, b, c, d and e. We use the red node as basepoint to construct homotopy classes. *Center:* Three homotopy classes $[h_1], [h_2], [g]$ each with a sequence of edges that form a cycle of minimal length. *Right:* Proof that $[h_1]$ and $[h_2]$ are conjugate via $[g]$. Each time an edge is followed by its inverse we remove them as they are homotopically trivial. By doing so, we reduce the cycle $[g^{-1}][h_1][g]$ to one that is homotopic to it and has minimal-length, namely $[h_2]$

counts every triangle exactly six times, since there are three edges each with two possible orientations. A proof by induction shows that the above argument holds for any p.

The second claim of Lemma 1 will be fundamental in our later exposition. We prefer to count NBCs of length p using $tr(B^p)$, as opposed to the entries of B^{p-1}, because of the well-known fact that $tr(B^p) = \sum_i \lambda_i^p$, for any square matrix B, and where each λ_i is an eigenvalue of B (Lang 2004). Also of importance is the fact that B is not symmetric, and hence its eigenvalues are, in general, complex numbers. Since B is a real matrix, all of its eigenvalues come in conjugated pairs: if $a + ib$ is an eigenvalue then so is $a - ib$, where i is the imaginary unit. This implies that we can write the above equation as $tr(B^p) = \sum_i \lambda_i^p = \sum_i \text{Re}\left(\lambda_i^p\right)$, where $\text{Re}(z)$ is the real part of a complex number, since the imaginary part of conjugated eigenvalues cancel out. In the rest of this work we refer to B's eigenvalues as the *non-backtracking eigenvalues*.

If one is interested in B's eigenvalues rather than in the matrix itself, one may use the so-called Ihara determinant formula (Hashimoto 1989; Bass 1992), which states that the eigenvalues of B different than ± 1 are also the eigenvalues of the $2n \times 2n$ block matrix

$$B' = \begin{pmatrix} A & I - D \\ I & 0 \end{pmatrix} \tag{2}$$

where A is the adjacency matrix, D is the diagonal matrix with the node degrees, and I is the $n \times n$ identity matrix.

Related work

Hashimoto (1989) discussed the non-backtracking cycles of a graph (and the associated non-backtracking matrix) in relation to the theory of Zeta functions in graphs. Terras (2011) explained the relationship between non-backtracking cycles and the free homotopy classes of a graph. More recently, the non-backtracking matrix has been applied to diverse applications such as node centrality (Martin et al. 2014; Grindrod et al. 2018) and community detection (Krzakala et al. 2013; Bordenave et al. 2015; Kawamoto 2016), and to the data mining tasks of clustering (Ren et al. 2011) and embedding (Jiang et al. 2018). In particular, the application to community detection is of special interest because it was shown that the non-backtracking matrix performs better at spectral clustering than the Laplacian matrix in some cases (Krzakala et al. 2013). Hence, there is recent interest in describing the eigenvalue distribution of the non-backtracking matrix in models such as the Erdős-Rényi random graph and the stochastic block model (Gulikers et al. 2017). Our work differs from other applied treatments of the non-backtracking matrix in that we arrive at its eigenvalues from first principles, as a relaxed version of the length spectrum. Concretely, we use the eigenvalues to compare graphs because the spectral moments of the non-backtracking matrix describe certain aspects of the length spectrum (see "Operationalizing the length spectrum" section). The spectral moments of other matrices (e.g., adjacency and Laplacian matrices) also describe structural features of networks (Estrada 1996; Preciado et al. 2013).
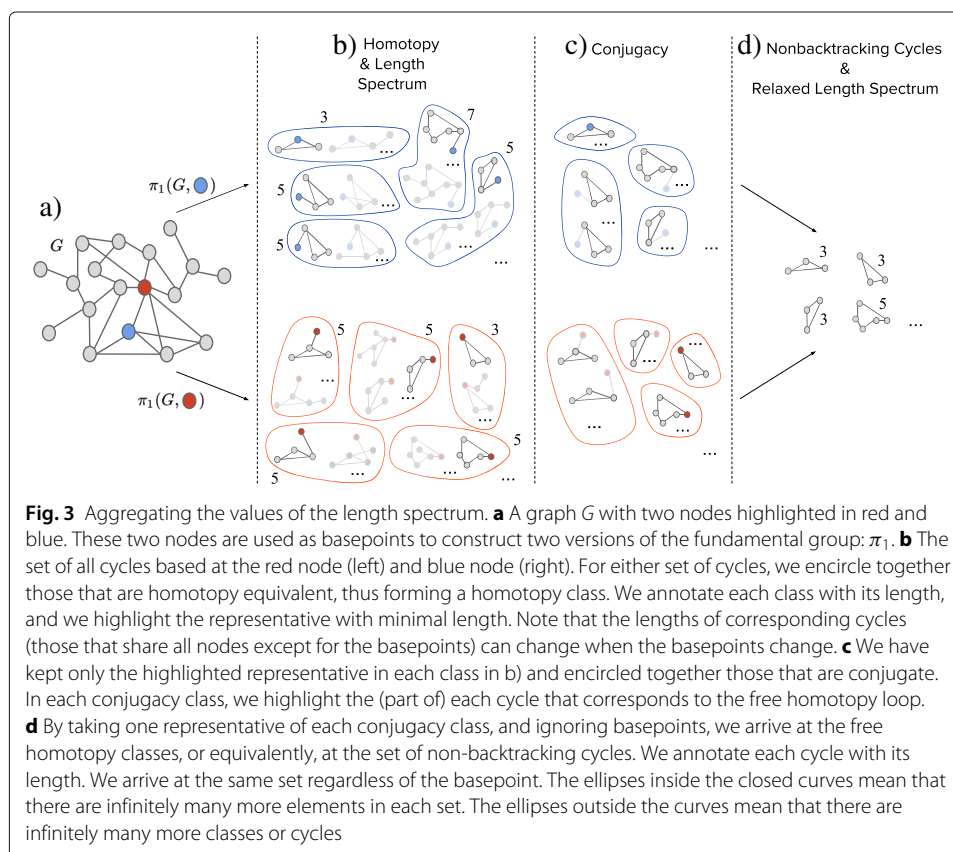
Many distance methods have been proposed recently (Soundarajan et al. 2014; Koutra et al. 2016; Bagrow and Bollt 2018; Bento and Ioannidis 2018; Onnela et al. 2012; Schieber et al. 2017; Chowdhury and Mémoli 2017; 2018; Berlingerio et al. 2013; Yaveroğlu et al. 2014). This proliferation is due to the fact that there is no definitive way of comparing

two graphs, especially complex networks since they present a vast diversity of structural features. The methods that are most related to ours fall in two categories: combinatorial enumeration or estimation of different subgraphs (such as motifs or shortest paths), or those that depend on spectral properties of some matrix representation of the network, such as the method Lap from "Data sets and base lines" section.

The topic of embedding has also seen a sharp increase of activity in recent years, motivated by the ubiquity of machine learning and data mining applications to structured data sets; see (Goyal and Ferrara 2018; Hamilton et al. 2017) for recent reviews. Our method NBED differs from most others in that it yields an edge embedding that differentiates the two possible orientations of an edge. Our focus is in providing an interpretable visual analysis of the resulting embedding, and on its application to anomaly detection.

Operationalizing the length spectrum

We want to quantify the dissimilarity between two graphs by measuring the dissimilarity between their length spectra. To do this, there are two main obstacles to overcome: (i) computing the exact length spectrum of a given graph is not computationally feasible as it depends on enumerating the length of every NBC, and (ii) it is not immediately clear how to compare two length spectra functions that come from two distinct graphs, because they are defined on disjoint domains (the fundamental groups of the graphs)[4]. In order to overcome these obstacles, we propose the use of what we call *the relaxed length spectrum*, denoted by $\mathcal{L}'$, and whose construction comes in the form of a two-step aggregation of the values of $\mathcal{L}$; see Fig. 3 for an overview of this procedure.

Fig. 3 Aggregating the values of the length spectrum. **a** A graph *G* with two nodes highlighted in red and blue. These two nodes are used as basepoints to construct two versions of the fundamental group: π_1. **b** The set of all cycles based at the red node (left) and blue node (right). For either set of cycles, we encircle together those that are homotopy equivalent, thus forming a homotopy class. We annotate each class with its length, and we highlight the representative with minimal length. Note that the lengths of corresponding cycles (those that share all nodes except for the basepoints) can change when the basepoints change. **c** We have kept only the highlighted representative in each class in b) and encircled together those that are conjugate. In each conjugacy class, we highlight the (part of) each cycle that corresponds to the free homotopy loop. **d** By taking one representative of each conjugacy class, and ignoring basepoints, we arrive at the free homotopy classes, or equivalently, at the set of non-backtracking cycles. We annotate each cycle with its length. We arrive at the same set regardless of the basepoint. The ellipses inside the closed curves mean that there are infinitely many more elements in each set. The ellipses outside the curves mean that there are infinitely many more classes or cycles

Relaxed Length Spectrum

The first step of this procedure is to focus on the image of the length spectrum rather than the domain (i.e., focus on the collection of lengths of cycles). The second step is to aggregate these values by considering the size of the level sets of either length spectrum.

Concretely, when comparing two graphs G and H, instead of comparing $\mathcal{L}_G$ and $\mathcal{L}_H$ directly, we compare the number of cycles in G of length 3 with the number of cycles in H of the same length, as well as the number of cycles of length 4, of length 5, etc, thereby essentially comparing the histogram of values that each $\mathcal{L}$ takes. Theoretically, focusing on this histogram provides a common ground to compare the two functions. In practice, this aggregation allows us to reduce the amount of memory needed to store the length spectra because we no longer keep track of the exact composition of each of the infinitely many (free) homotopy classes. Instead, we only keep track of the frequency of their lengths. According to this aggregation, we define the *relaxed* version of the length spectrum as the set of points $\mathcal{L}' = \{(k, n(k)) : k = 1, 2, ..\}$, where $n(k)$ is the number of conjugacy classes of π_1 (i.e., free homotopy classes) of length k.

The major downside of removing focus from the underlying group structure and shifting it towards (the histogram of values in) the image is that we lose information about the combinatorial composition of each cycle. Indeed, $\pi_1(G)$ holds information about the number of cycles of a certain length k in G; this information is also stored in $\mathcal{L}'$. However, the group structure of $\pi_1(G)$ also allows us to know how many of those cycles of length k are formed by the concatenation of two (three, four, etc.) cycles of different lengths. This information is lost when considering only the sizes of level sets of the image, i.e., when considering $\mathcal{L}'$.

The next step makes use of the non-backtracking cycles (NBCs). We rely on NBCs because the set of conjugacy classes of $\pi_1(G)$ is in bijection with the set of NBCs of G see e.g., Terras (2011), Hashimoto (1989). In other words, to compute the set $\mathcal{L}'$ we need only account for the lengths of all NBCs. Indeed, consider the non-backtracking matrix B of G and recall that $tr\left(B^k\right)$ equals the number of NBCs of length k in the graph. This gives us precisely the set $\mathcal{L}' = \left\{\left(k, tr\left(B^k\right)\right)\right\}_{k=1}^{\infty}$. Recall that $tr\left(B^k\right) = \sum_i \lambda_i^k$, where each λ_i is a non-backtracking eigenvalue. Therefore, the eigenvalues of B contain all the information necessary to compute and compare $\mathcal{L}'$. In this way, we can study the (eigenvalue) spectrum of B, as a proxy for the (length) spectrum of π_1.

We have deviated from the original definition of the length spectrum in important ways. Fortunately, our experiments indicate that $\mathcal{L}'$ contains enough discriminatory information to distinguish between real and synthetic graphs effectively; see "Clustering networks" and "Rewiring edges" sections. We discuss this limitation further in our concluding remarks in "Discussion and conclusions" section. Beyond experimental results, one may ask if there are theoretical guarantees that the relaxed version of the length spectrum will keep some of the discriminatory power of the original. Indeed, even though our inspiration for this work is partially the main rigidity result of (Constantine and Lafont 2018), we can still trust the eigenvalue spectrum of B to be useful when comparing graphs. On the one hand, the spectrum of B has been found to yield fewer isospectral graph pairs (i.e., non-isomorphic graphs with the same eigenvalues) than the adjacency and Laplacian matrices in the case of small graphs (Durfee and Martin 2015). On the other hand, B is tightly related to the theory of graph zeta functions (Hashimoto 1989), in particular the Ihara Zeta function, which is known to determine several graph

properties such as girth, number of spanning trees, whether the graph is bipartite, regular, or a forest, etc (Cooper 2009). Thus, both as a relaxed version of the original length spectrum, but also as an object of interest in itself, we find the eigenvalue spectrum of the non-backtracking matrix B to be an effective means of determining the dissimilarity between two graphs. For the rest of this work we focus on the eigenvalues of B and on using them to compare complex networks.

Computing B

As mentioned previously, the eigenvalues of B different than ± 1 are also the eigenvalues of the $2n \times 2n$ block matrix B' defined in Eq.2. Computing its eigenvalues can then be done with standard computational techniques, and it will not incur a cost greater than computing the eigenvalues of the adjacency or Laplacian matrices, as B' is precisely four times the size of A and it has only $2n$ more non-zero entries (see Eq. 2). This is what we will do in "NBD: Non-backtracking distance" section in order to compare two relaxed length spectra. However, in "NBED: Non-backtracking embedding dimensions" section we will need the eigenvectors of B, which cannot be computed from B'. For this purpose, we now present an efficient algorithm for computing B. Computing its eigenvectors can then be done with standard techniques.

Given a graph with n nodes and m undirected edges in edge list format, define the $n \times 2m$ incidence matrices $P_{x,u \to v} = \delta_{xu}$ and $Q_{x,u \to v} = \delta_{xv}$, and write $C = P^T Q$. Observe that $C_{k \to l, u \to v} = \delta_{vk}$ and therefore,

$$B_{k \to l, u \to v} = C_{k \to l, u \to v}(1 - C_{u \to v, k \to l}) \tag{3}$$

Note that an entry of B may be positive only when the corresponding entry of C is positive. Therefore, we can compute B in a single iteration over the nonzero entries of C.

Now, C has a positive entry for each pair of incident edges in the graph, from which we have the following result.

Lemma 2 *Set $nnz(C)$ to be the number of non-zero entries in C, and $\langle k^2 \rangle$ the second moment of the degree distribution. Then, $nnz(C) = n\langle k^2 \rangle$.*

Proof 2 *This follows by direct computation: $\sum_{k \to l, i \to j} \delta_{kj} = \sum_k \left(\sum_l a_{lk} \right) \left(\sum_i a_{ik} \right) = n\langle k^2 \rangle$*

Computing P and Q takes $O(m)$ time. Since computing the product of sparse matrices takes time proportional to the number of positive entries in the result, computing C takes $O(n\langle k^2 \rangle)$. Thus we can compute B in time $O(m + n\langle k^2 \rangle)$. In the case of a power-law degree distribution with exponent $2 \leq \gamma \leq 3$, the runtime of our algorithm falls between $O(m + n)$ and $O(m + n^2)$. Note that if a graph is given in adjacency list format, one can build B directly from the adjacency list in time $\Theta(n\langle k^2 \rangle - n\langle k \rangle)$ by generating a sparse matrix with the appropriate entries set to 1 in a single iteration over the adjacency list.

Spectral properties of B

In our experiments we have found that the eigenvalues of B behave nicely with respect to certain fundamental properties of complex networks such as degree distribution and triangles. If the theory of the length spectrum justifies the use of the non-backtracking

eigenvalues to compare graphs in general, the properties we present next justify their use for complex networks in particular.

Lemma 3 *We have $nnz(B) = n\left(\langle k^2 \rangle - \langle k \rangle\right)$.*

Proof 3 *We directly compute $\sum_{k \to l, i \to j} \delta_{kj}(1 - \delta_{il}) = \sum_k (\deg(k) - 1) \deg(k) = n\left(\langle k^2 \rangle - \langle k \rangle\right)$.*

Contrast this to $nnz(A) = n\langle k \rangle$ for the adjacency matrix A. As a consequence of this, we have found experimentally that the larger $\langle k^2 \rangle$, the larger the range of the non-backtracking eigenvalues along the imaginary axis; see Fig. 4.
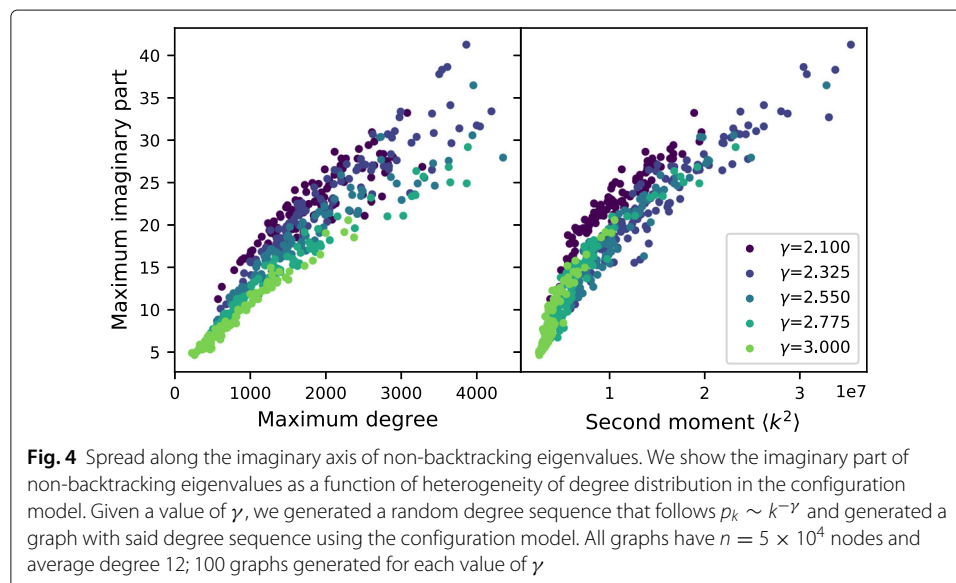
Next, we turn to B's eigenvalues and their relation to the number of triangles. Write $\lambda_k = a_k + ib_k \in \mathbb{C}$ for the non-backtracking eigenvalues, where i is the imaginary unit and $k = 1, .., 2m$. The number of triangles in a network is proportional to $tr(B^3) = \sum_k Re\left(\lambda_k^3\right)$, which, by a direct application of the binomial theorem, equals

$$tr\left(B^3\right) = \sum_{k=1}^{2m} a_k \left(a_k^2 - 3b_k^2\right). \tag{4}$$

In general, the eigenvalues of B with small absolute value tend to fall on a circle in the complex plane (Krzakala et al. 2013; Angel et al. 2015; Wood and Wang 2017). However, if $\sum_k a_k^2$ is large and $\sum_k b_k^2$ is small (implying a large number of triangles), the eigenvalues cannot all fall too close to a circle, since they will need to have different absolute values. Hence, the more triangles in the graph, the less marked the circular shape of the eigenvalues. Another way of saying the same thing is that the more triangles exist in the graph, the eigenvalues have larger and positive real parts.

Eigenvalues: visualization

We close this section with a visualization of the non-backtracking eigenvalues. We choose six different random graph models: Erdös-Rényi (ER) (Erdös and Rényi 1960),

Fig. 4 Spread along the imaginary axis of non-backtracking eigenvalues. We show the imaginary part of non-backtracking eigenvalues as a function of heterogeneity of degree distribution in the configuration model. Given a value of γ, we generated a random degree sequence that follows $p_k \sim k^{-\gamma}$ and generated a graph with said degree sequence using the configuration model. All graphs have $n = 5 \times 10^4$ nodes and average degree 12; 100 graphs generated for each value of γ

(Bollobás 2001), Barabási-Albert (BA) (Barabási and Albert 1999), Stochastic Kronecker Graphs (KR) (Leskovec et al. 2010; Seshadri et al. 2013), Configuration Model with power law degree distribution (CM; $p_k \sim k^{-\gamma}$ with $\gamma = 2.3$) (Newman 2003), Watts-Strogatz (WS) (Watts and Strogatz 1998), Hyperbolic Graphs (HG; $\gamma = 2.3$) (Krioukov et al. 2010; Aldecoa et al. 2015). From each model, we generate 50 graphs with $n = 10,000$ nodes and approximately $\langle k \rangle = 15$; see Table 2 for more information about these graphs. For each graph, we plot the largest $r = 200$ eigenvalues on the complex plane in Fig. 1.

Each model generates eigenvalue distributions presenting different geometric patterns. As expected from the previous section, HG has a less marked circular shape around the origin since it is the model that generates graphs with the most triangles. Furthermore, HG also has the largest maximum degree, and therefore its eigenvalues have a greater spread along the imaginary axis. We also show in Fig. 1 a low-dimensional projection of the non-backtracking eigenvalues using UMAP (McInnes et al. 2018). Notice how, even in two dimensions, the non-backtracking eigenvalues are clustered according to the random graph model they come from. This provides experimental evidence that the non-backtracking eigenvalues shall be useful when comparing complex networks. For a more in-depth explanation of the parameters used for UMAP, see Appendix 1.

NBD: Non-backtracking distance

Based on the previous discussion, we propose a method to compute the distance between two complex networks based on the non-backtracking eigenvalues. In this section, we use d to refer to an arbitrary metric defined on subsets of $\mathbb{R}^2$. That is, the distance between two graphs G, H is given by $d(\{\lambda_k\}, \{\mu_k\})$, where λ_k, μ_k are the eigenvalues of G, H, respectively, which we identify with points in $\mathbb{R}^2$ by using their real and imaginary parts as coordinates. With these notations we are finally ready to propose the Non-Backtracking Spectral Distance, or NBD for short.

Definition 1 *Consider two graphs G, H, and let $\{\lambda_k\}_{k=1}^{k=r}, \{\mu_k\}_{k=1}^{k=r}$, be the r non-backtracking eigenvalues of largest magnitude of G and H, respectively. We define the NBD between G and H as $NBD(G, H) = d(\{\lambda_k\}, \{\mu_k\})$.*

Note that in Definition 1 we leave open two important degrees of freedom: the choice of d and the choice of r. We will discuss the optimal choices for these parameters in the next section. Regardless of these choices, however, we have the following results.

Proposition 1 *If d is a metric over subsets of $\mathbb{R}^2$, then NBD is a pseudo-metric over the set of graphs.*

Proof 4 *NBD inherits several desirable properties from d: non-negativity, symmetry, and, importantly, the triangle inequality. However, the distance between two distinct graphs may be zero when they share all of their r largest eigenvalues. Thus, NBD is not a metric over the space of graphs but a pseudo-metric.*

Computing NBD

Algorithm 1 presents our method to compute the NBD between two graphs. It makes use of the following fact, which simplifies the computation. It is known (see e.g. Durfee and

Table 2 Data sets used in this work

data set	network	n	$\langle k \rangle$	$\bar{c}$
Online social (social)	Facebook	4k	43.69	0.61
	sdot8	77k	14.13	0.06
	sdot9	82k	14.18	0.06
	Epinions	76k	10.69	0.14
	Twitter	81k	33.02	0.57
	Wiki	7k	28.32	0.14
Coauthorship (CA)	AstroPh	19k	21.11	0.63
	CondMat	23k	8.08	0.63
	GrQc	5k	5.53	0.53
	HepPh	12k	19.74	0.61
	HepTh	10k	5.26	0.47
Peer-to-peer file-sharing (P2P)	P2P04	11k	7.35	0.01
	P2P05	9k	7.20	0.01
	P2P06	9k	7.23	0.01
	P2P08	6k	6.59	0.01
	P2P09	8k	6.41	0.01
	P2P24	27k	4.93	0.01
	P2P25	23k	4.82	0.01
	P2P30	36k	4.82	0.01
	P2P31	63k	4.73	0.01
Autonomous systems (AS)	AS331	11k	4.12	0.30
	AS407	11k	4.10	0.29
	AS414	11k	4.16	0.30
	AS421	11k	4.19	0.30
	AS428	11k	4.13	0.29
	AS505	11k	4.13	0.29
	AS512	11k	4.12	0.29
	AS519	11k	4.11	0.29
	AS526	11k	4.19	0.30
`BA-1k`		1k	13.90	0.05
`CM-1k`		1k	10.20	0.17
`ER-1k`		1k	14.99	0.15
`HG-1k`		997 ± 2	14.72	0.82
`KR-1k`		1021 ± 1	14.54	0.03
`WS-1k`		1k	14.00	0.67
`BA-5k`		5k	13.98	0.02
`CM-5k`		5k	11.40	0.13
`ER-5k`		5k	15.00	0.00
`HG-5k`		4982 ± 8	14.67	0.82
`KR-5k`		4089 ± 3	18.27	0.01
`WS-5k`		5k	14.00	0.67
`BA-10k`		10k	13.99	0.01
`CM-10k`		10k	11.97	0.12
`ER-10k`		10k	14.99	0.00
`HG-10k`		9958 ± 15	14.96	0.83
`KR-10k`		8170 ± 11	18.32	0.01
`WS-10k`		10k	14.00	0.67

n: number of nodes in the largest connected component; $\langle k \rangle$: average degree; $\bar{c}$: average clustering coefficient. *Bottom:* random graph data sets. We generate 50 graphs of each data set, for a total of 900 random graphs. *Top:* real networks, grouped in four data sets

Martin (2015)) that the multiplicity of 0 as an eigenvalue of B equals the number of edges outside of the 2-core of the graph. For example, a tree, whose 2-core is empty, has all its eigenvalues equal to 0. On the one hand, we could use this valuable information as part of our method to compare two graphs. On the other hand, the existence of zero eigenvalues does not change the value of $tr\left(B^k\right), k \geq 0$, and thus leaves the relaxed length spectrum $\mathcal{L}' = \left\{\left(k, tr\left(B^k\right)\right)\right\}_k$ intact. Moreover, iteratively removing the nodes of degree one (a procedure called "shaving") reduces the size of B (or the sparsity of B'; see Eq. 2), which makes the computation of non-zero eigenvalues faster.

Algorithm 1 Non-Backtracking Spectral Distance

Input: Two graphs G, H**, distance d, number of eigenvalues r**
Output: real number d, distance between G, H

  1: $\tilde{G}, \tilde{H} \leftarrow \text{shave}(G), \text{shave}(H)$
  2: $\{\lambda_k\}_k, \{\mu_k\}_k \leftarrow$ the r largest eigenvalues of B' corresponding to $\tilde{G}, \tilde{H}$
  3: **return** $d(\{\lambda_i\}, \{\mu_i\})$

Given two graphs G, H, we first compute their 2-cores by shaving them, that is, iteratively removing the nodes of degree 1 until there are none; call the new graphs $\tilde{G}$ and $\tilde{H}$. Then for each graph, we compute the B' matrix using Eq. 2 to then compute the largest r eigenvalues. Lastly, we compute the distance d between the resulting sets of eigenvalues. We proceed to analyze the runtime complexity of each line of Algorithm 1. Line 1: the shaving step has a worst-case scenario of $O(n^2)$, where n is the number of nodes. Indeed, the shaving algorithm must iterate two steps until completion: first, identify all nodes of degree 1, and second, remove those nodes from the graph. For the first step, querying the degree of all nodes takes $O(n)$ time, while the deletion step takes an amount of time that is linear in the amount of nodes that are removed at that step. In the worst-case scenario, consider a path graph. At each iteration, the degrees of all nodes must be queried, and exactly two nodes are removed. The algorithm terminates after $n/2$ steps, at each of which it must query the degree of all remaining nodes, thus it takes $O(n^2)$. In several classes of complex networks, however, the number of nodes removed by the algorithm (i.e. those outside the 2-core) will be a small fraction of n. Line 2: the eigenvalue computation can be done in $O(n^3)$ using, for example, power iteration methods, though its actual runtime will greatly depend on the number of eigenvalues and the structure of the matrix. Line 3: finally, computing the distance between the computed eigenvalues depends on which distance is being used; see next section.

Choice of distance and dependence on r

Previous works provide evidence for the fact that the majority of the non-backtracking eigenvalues have magnitude less than $\rho = \sqrt{\lambda_1}$, where λ_1 is the largest eigenvalue (which is guaranteed to be real due to the Perron-Frobenius theorem since all of B's entries are real and non-negative). Indeed, in (Saade et al. 2014) the authors show that, in the limit of large network size, non-backtracking eigenvalues with magnitude larger than ρ occur with probability 0. Therefore, to compare two graphs, we propose to compute all the eigenvalues that have magnitude larger than ρ. Informally, this means we are comparing

the outliers of the eigenvalue distribution, since they occur with less probability the larger the network becomes. This provides a heuristic to choose the value of r.

To test this heuristic, as well as choose the distance d, we perform the following experiment. Given two graphs G_1 and G_2, we compute the largest 1000 non-backtracking eigenvalues of both graphs. Then, we compute three distance measures between $\{\lambda_i\}_1^r$ and $\{\mu_i\}_1^r$ for increasing values of r. The distances are defined as follows:

- The `Euclidean` metric computes the Euclidean distance between the vectors whose elements are the eigenvalues sorted in order of magnitude:
  Euclidean $(\{\lambda_i\}, \{\mu_i\}) = \sqrt{\sum_i |\lambda_i - \mu_i|^2}$, with $|\lambda_i| \geq |\lambda_{i+1}|$ and $|\mu_i| \geq |\mu_{i+1}|$ for all i.
  If two eigenvalues have the same magnitude, we use their real parts to break ties. Computing this distance takes $O(r)$ computations, where r is the number of eigenvalues.
- The `EMD` metric computes the Earth Mover Distance (a.k.a. Wasserstein distance) (Rubner et al. 1998) which is best understood as follows. Write $\lambda_k = a_k + ib_k$ and $\mu_k = c_k + id_k$ for all k. Consider the sets $\{(a_k, b_k)\}_{k=1}^r$ and $\{(c_k, d_k)\}_{k=1}^r$ as subsets of $\mathbb{R}^2$, and imagine a point mass with weight $1/r$ at each of these points. Intuitively, `EMD` measures the minimum amount of work needed to move all point masses at the points (a_k, b_k) to the points (c_k, d_k). This distance requires the computation of the (Euclidean) distance between every pair of eigenvalues and thus it takes $O(r^2)$ runtime.
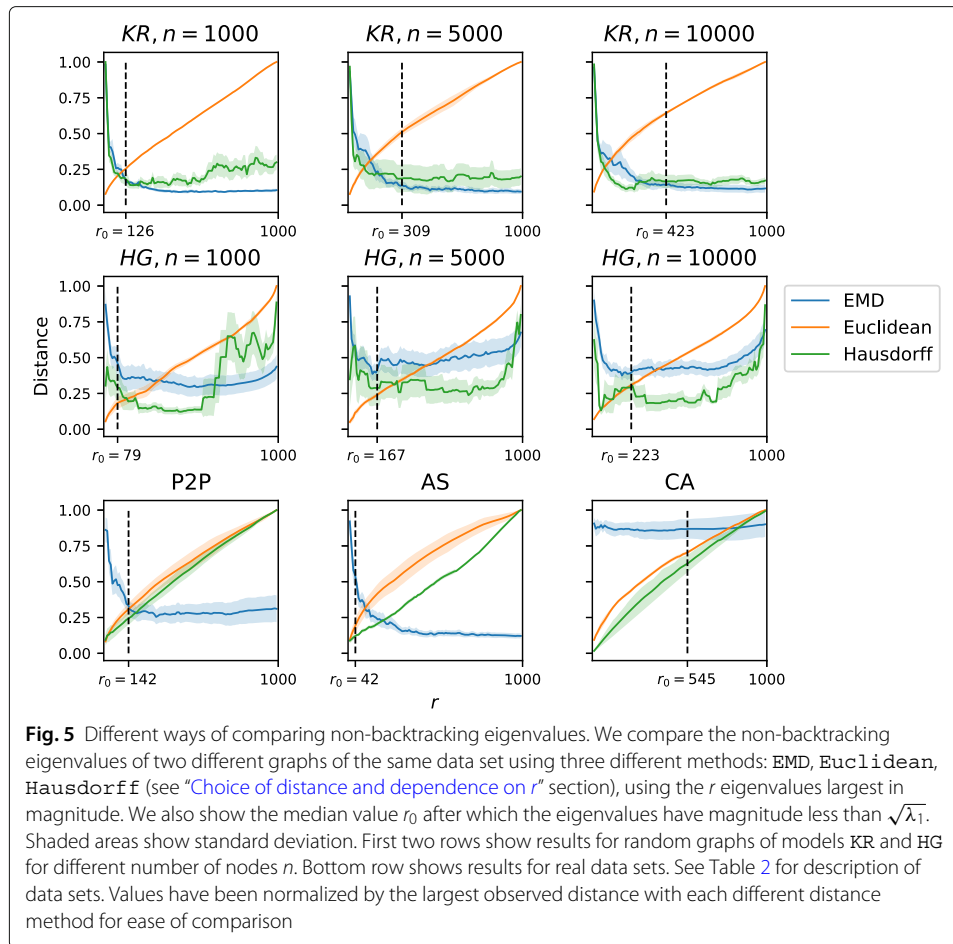- `Hausdorff` metric is a standard distance between compact sets in a metric space (Munkres 2000); it is defined as

$$\text{Hausdorff}(\{\lambda_i\}, \{\mu_i\}) = \max\left(\max_i \min_j |\lambda_i - \mu_j|, \max_j \min_i |\lambda_i - \mu_j|\right). \qquad (5)$$

Similarly as above, this computation takes $O(r^2)$ runtime.

From each of the following data sets in turn we take two graphs and run this experiment: HG, KR, P2P, AS, CA (see Table 2 and "Data sets and base lines" section for full description of data sets). We report the results in Fig. 5, where we plot the average of the three distances for each data set across several repetitions of the experiment (10 repetitions for random graphs and CA, 36 for P2P and AS), as well as the median value of r after which the eigenvalues have magnitude less than ρ; we call this value r_0.

In Fig. 5 we see that `Hausdorff` has high variance and a somewhat erratic behavior on random graphs. `Euclidean` has the least variance and most predictable behavior, continuing to increase as the value of r grows. In the KR data sets, `Hausdorff` presents an "elbow" near r_0, after which it levels off without much change for increasing values of r. Interestingly, the plots suggest that, in the case of `EMD`, the steepest decrease usually occurs before r_0. Also of note is the fact that on HG, all distances have a slight increase for large values of r.

Since `Hausdorff` does not appear to have a predictable behavior, and `Euclidean` continues to increase with no discernible inflection point even when using the largest $r = 1000$ eigenvalues, we favor `EMD` when comparing non-backtracking eigenvalues of two graphs. Furthermore, `EMD` has a predictable "elbow" behavior that happens near r_0. Thus we conclude that this is an appropriate choice for the number of eigenvalues to

Fig. 5 Different ways of comparing non-backtracking eigenvalues. We compare the non-backtracking eigenvalues of two different graphs of the same data set using three different methods: `EMD`, `Euclidean`, `Hausdorff` (see "Choice of distance and dependence on *r*" section), using the *r* eigenvalues largest in magnitude. We also show the median value r_0 after which the eigenvalues have magnitude less than $\sqrt{\lambda_1}$. Shaded areas show standard deviation. First two rows show results for random graphs of models `KR` and `HG` for different number of nodes *n*. Bottom row shows results for real data sets. See Table 2 for description of data sets. Values have been normalized by the largest observed distance with each different distance method for ease of comparison

use. In the following we use `EMD` as the distance *d* and r_0 as the value of *r* for all our experiments.[5]

NBD: Experiments
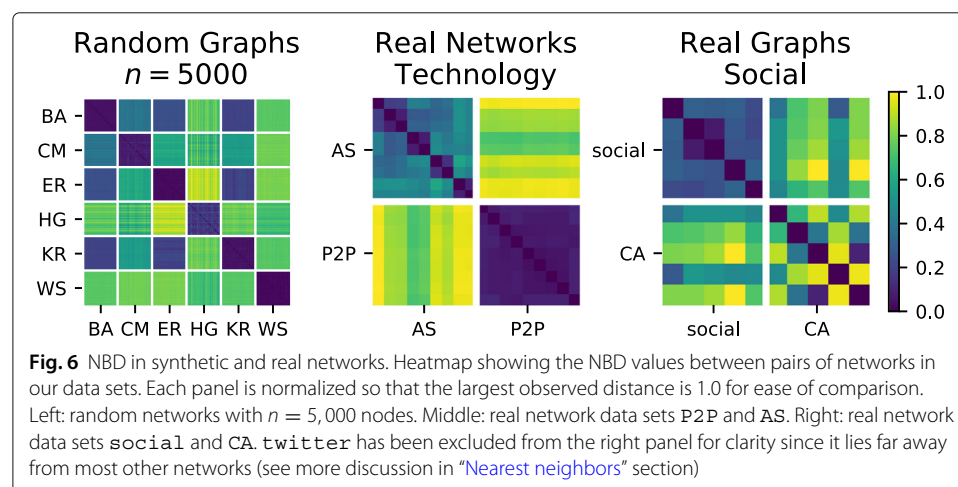
Data sets and base lines

A description of the data sets we use throughout this article can be found in Table 2. We use random graphs obtained from six different models: Erdös-Rényi (`ER`) (Erdös and Rényi 1960; Bollobás 2001), Barabási-Albert (`BA`) (Barabási and Albert 1999), Stochastic Kronecker Graphs (`KR`) (Leskovec et al. 2010; Seshadhri et al. 2013), Configuration Model with power law degree distribution (`CM`) (Newman 2003), Watts-Strogatz (`WS`) (Watts and Strogatz 1998), and Hyperbolic Graphs (`HG`) (Krioukov et al. 2010; Aldecoa et al. 2015). We generated `CM` and `HG` with degree distribution $p_k \sim k^{-\gamma}$ with exponent $\gamma = 2.3$, and `WS` with rewiring probability $p = 0.01$. All graphs have approximate average degree $\langle k \rangle = 15$. From each model, we generate three batches, each with 50 graphs: the first with $n = 1,000$ nodes, the second with $n = 5,000$, and the third with $n = 10,000$. This yields 6×3 data sets that we refer to by model and graph size (see Table 2). We also use real networks, divided in four different data sets: `social`, `CA`, `AS`, and `P2P`. `social` contains six networks obtained from human online social networks (`facebook`, `twitter`, `sdot8`, `sdot9`, `epinions`, and `wiki`) (McAuley and Leskovec 2012; Leskovec et al.

[2010](#); Leskovec et al. [2009](#); Richardson et al. [2003](#)), CA contains five co-authorship networks (`AstroPh`, `CondMat`, `GrQc`, `HepPh`, `HepTh`) obtained from the arXiv pre-print server (Leskovec et al. [2007](#)), `P2P` contains nine snapshots of the peer-to-peer connections of a file-sharing network (Leskovec et al. [2007](#)), and `AS` contains nine snapshots of the Internet autonomous systems infrastructure (Leskovec et al. [2005](#)). Note that `P2P` and `AS` are temporal snapshots of the same network and thus we may assume that they have been generated using the same growth process. The same is not true for `social` and `CA`: even though they contain similar kinds of networks, their generating mechanisms may be different. The real networks were obtained through SNAP (Leskovec and Krevl [2014](#)) and ICON (Clauset et al.). In all, we have 929 different networks (900 random, 29 real), comprising 22 different data sets (6×3 random, 4 real).

Figure [6](#) shows the result of computing NBD on different synthetic and real data sets. Observe that NBD is capable of distinguishing between random graphs generated from different models, as well as between the technological data sets `P2P` and `AS`. NBD detects less of a difference between data sets `social` and `CA`, which points toward the homogeneity of connectivity patterns in human-generated networks.

Among all distance methods available, we choose the following three to compare against NBD. The first method is `ESCAPE` (Pinar et al. [2017](#)) which given a graph G computes a 98-dimensional vector v of motif counts, i.e., v_i is the number of times that the ith motif appears in G for $i = 1, ..., 98$. To compare two graphs we then compute the cosine distance between their `ESCAPE` feature vectors. The second method is Graphlet Correlation Distance (Yaveroğlu et al. [2014](#)), or `GCD` for short, which characterizes a graph with a so-called *graphlet correlation matrix*, a 11×11 matrix whose entries are the correlation between the number of per-node occurrences of 11 small motifs. The distance between two graphs is then the Frobenius norm of the difference of their graphlet correlation matrices. The third method is taken from Schieber et al (Schieber et al. [2017](#)), and will be referred to here as `S`. Distance measure `S` depends on the distribution of shortest path length distances between each pair of nodes in a graph. In this way, we are comparing NBD, which depends on the number and length of non-backtracking cycles, to other distance methods that also depend on the occurrence of a particular type of subgraphs: `ESCAPE` and `GCD` depend on motif counts, while distance S depends on lengths of

Fig. 6 NBD in synthetic and real networks. Heatmap showing the NBD values between pairs of networks in our data sets. Each panel is normalized so that the largest observed distance is 1.0 for ease of comparison. Left: random networks with $n = 5,000$ nodes. Middle: real network data sets `P2P` and `AS`. Right: real network data sets `social` and `CA`. `twitter` has been excluded from the right panel for clarity since it lies far away from most other networks (see more discussion in "Nearest neighbors" section)
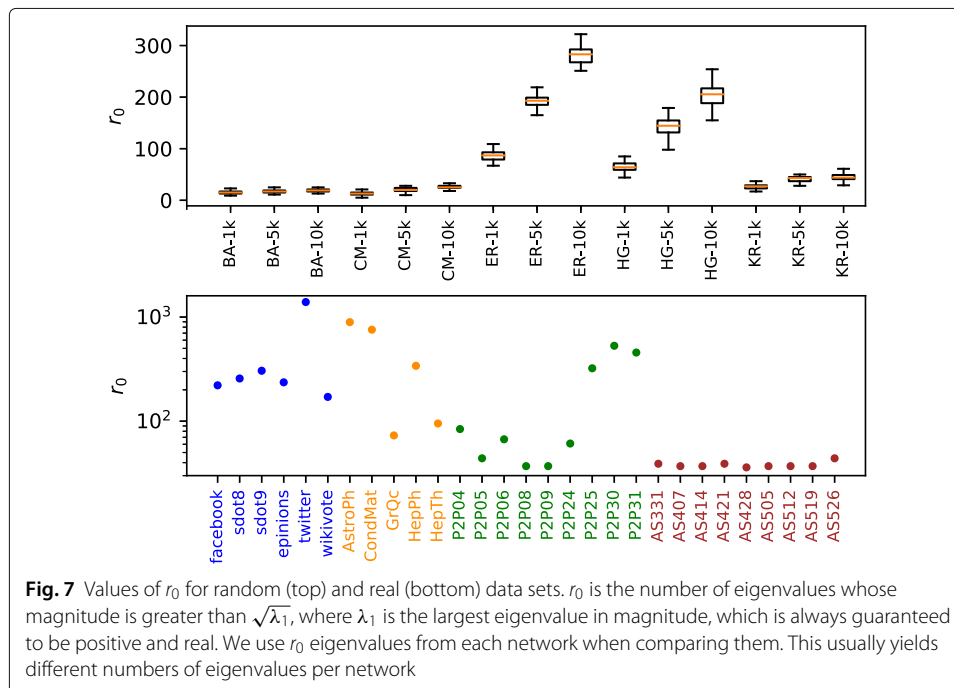
shortest paths. As a baseline, we use the 1-dimensional EMD between the $r = 300$ largest Laplacian eigenvalues of each graph; we call this baseline `Lap`. We report the performance of two versions of NBD: NBD-300 uses the largest $r = 300$ non-backtracking eigenvalues of each graph, while NBD-ρ uses $r = \min\{300, r_0\}$ where r_0 is the number of eigenvalues whose magnitude is greater than $\rho = \sqrt{\lambda_1}$, as explained before. We use a maximum of 300 because most of our networks have $r_0 < 300$; see Fig. 7 for values of r_0 for all our data sets. This means that NBD-ρ sometimes uses vastly different numbers of eigenvalues to compare two networks. For example, when comparing a `BA-10k` graph to `twitter`, we are comparing approximately 15 eigenvalues versus $\min\{300, r_0 = 1393\} = 300$. Note that `EMD` handles this difference naturally.

Using each distance method in turn, we obtain a 929×929 distance matrix containing the distance between each pair of networks in our data sets. Using these distance matrices we perform the following experiments.

Clustering networks

Our main experiment is to evaluate the distance methods in a clustering context (Yaveroğlu et al. 2015). We use each distance matrix as the input to a spectral clustering algorithm. To form a proximity graph from the distance matrix, we use the RBF kernel. That is, if the distance between two graphs under the i-th distance method is d, their proximity is given by $\exp\left(-d^2/2\sigma_i^2\right)$. We choose σ_i as the mean within-data set average distance, that is, we take the average distance among all pairs of `ER` graphs, the average distance among all pairs of `BA` graphs, and so on for each data set of random graphs, and average all of them together (von Luxburg 2007). Note this yields a different σ for each distance method, and that we do not use graphs from real data sets to tune this parameter. We program the algorithm to find the exact number of ground-truth clusters. We report

Fig. 7 Values of r_0 for random (top) and real (bottom) data sets. r_0 is the number of eigenvalues whose magnitude is greater than $\sqrt{\lambda_1}$, where λ_1 is the largest eigenvalue in magnitude, which is always guaranteed to be positive and real. We use r_0 eigenvalues from each network when comparing them. This usually yields different numbers of eigenvalues per network
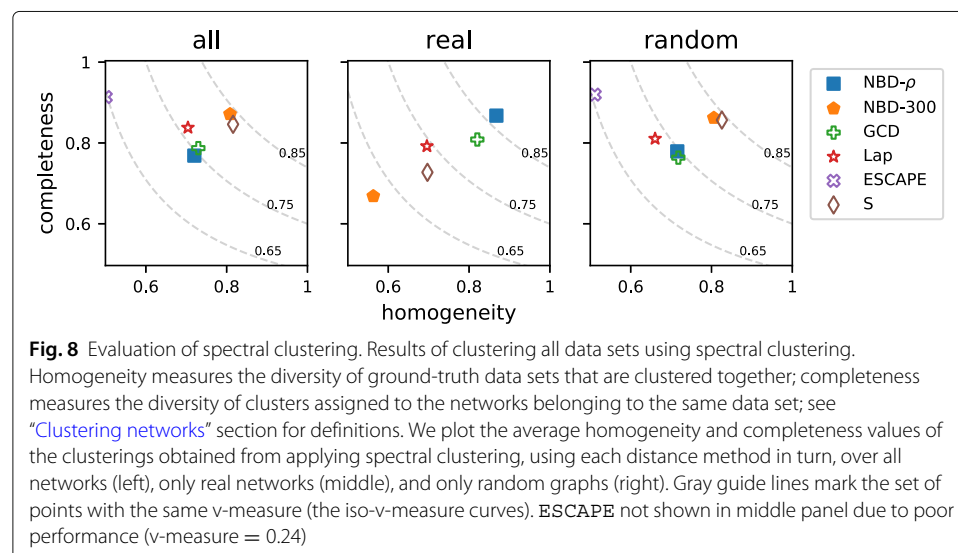
the results of this experiment using all graphs (22 clusters), only real graphs (4 clusters), and only random graphs (18 clusters).

Once we have obtained the clusters, we use as performance metrics *homogeneity*, *completeness*, and *v-measure*. Homogeneity measures the diversity of data sets within a cluster, and is maximized when all data sets in each cluster come from the same data set (higher is better). Completeness is the converse; it measures the diversity of clusters assigned to the graphs in one data set (higher is better). The v-measure is the harmonic mean of homogeneity and completeness (Rosenberg and Hirschberg 2007). Concretely, let C be the ground-truth class of a network chosen uniformly at random, and let K be the cluster assigned to a network chosen uniformly at random. Homogeneity is defined as $1 - H(C|K)/H(C)$ where $H(C)$ is the entropy of the class distribution, and $H(C|K)$ is the conditional entropy of the class distribution given the cluster assignment. If the members of each cluster have the same class distribution as the total class distribution, then $H(C|K) = H(C)$, and therefore homogeneity is 0. If each cluster contains elements of the same class, i.e., if the conditional distribution of classes given the clusters is constant, then $H(C|K) = 0$ and homogeneity is 1. If $H(C) = 0$, then homogeneity is defined to be 1 by convention. Similarly, completeness is defined as $1 - H(K|C)/H(K)$, and conventionally equal to 1 when $H(K) = 0$. Note that homogeneity is analogous to *precision*, while completeness is analogous to *recall*, which are used in binary classification. Correspondingly, v-measure is analogous to the F1-score.

In Fig. 8 we present the results of this experiment. When clustering all graphs, NBD-300 performs best (v-measure 0.84), followed by S (v-measure 0.83). Distances S and GCD perform well across data sets, however, numerical methods for computing eigenvalues are orders of magnitude faster than the state-of-the-art methods for computing all shortest path distances (needed to compute S) or per-node motif counts (needed for GCD); see "Runtime comparison" section. ESCAPE performs well in completeness but not in homogeneity in random graphs, and extremely poorly in real graphs (v-measure 0.27).

Interestingly, when clustering only the real networks, NBD-300 performs more poorly than when using random graphs, while NBD-ρ obtains the highest v-measure score across

Fig. 8 Evaluation of spectral clustering. Results of clustering all data sets using spectral clustering. Homogeneity measures the diversity of ground-truth data sets that are clustered together; completeness measures the diversity of clusters assigned to the networks belonging to the same data set; see "Clustering networks" section for definitions. We plot the average homogeneity and completeness values of the clusterings obtained from applying spectral clustering, using each distance method in turn, over all networks (left), only real networks (middle), and only random graphs (right). Gray guide lines mark the set of points with the same v-measure (the iso-v-measure curves). ESCAPE not shown in middle panel due to poor performance (v-measure = 0.24)
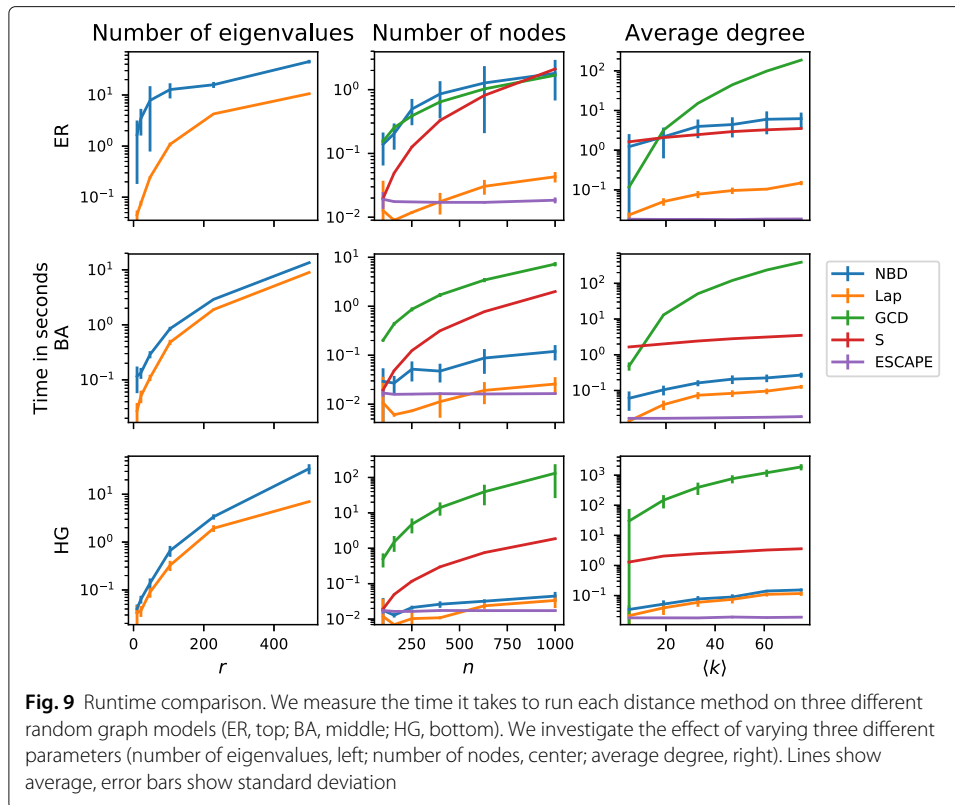
all our experiments (0.87). This indicates that the choice of $r = r_0$ as the number of eigenvalues larger than $\sqrt{\lambda_1}$ can be both more efficient and more informative than computing a set constant value of, say, $r = 300$.

We conclude from this experiment that (i) counting motifs locally (number of per-node occurrences) is better than counting them globally (number of occurrences in the whole graph), because GCD performs better than ESCAPE across data sets, (ii) that comparing length distributions of paths or cycles seems to yield better performance than motif counting, because NBD and S perform better than others in general, and (iii) that non-backtracking eigenvalues provide slightly better performance at distinguishing between real and synthetic graphs than other combinatorial methods, as well as other spectral methods such as Lap. We highlight that NBD is in general also more efficient than the combinatorial methods, while being only slightly more time-consuming than the spectral method Lap; see next section.

Runtime comparison

As part of "Computing NBD" section we included a complexity analysis of the runtime of NBD (Algorithm 1). Here, we include a direct comparison of runtime of NBD and all other baseline algorithms. Observe that all five graph distance algorithms used have something in common: they all run in two steps. The first step is to compute a certain kind of summary statistics of each of the two graphs to be compared, and the second step is to compare those precomputed statistics. For NBD and Lap, the first step computes eigenvalues of the corresponding matrices, while GCD and ESCAPE compute motif counts, and S computes the shortest path distance between each pair of nodes in the graph. The bulk of the computational time of all methods is spent in the computation of these statistics, rather than in the comparison. Therefore, in this section we compare the runtime of this first step only. Another reason to focus on the runtime of this first step only is that it depends on one single graph (in all cases, the computation of the corresponding statistics is done on each graph in parallel), and thus we can focus on investigating how the structure of a single graph affects computation time. All experiments were conducted on a computer with 16 cores and 100GB of RAM with Intel Skylake chip architecture, with no other jobs running in the background. Note that all distance algorithms are implemented in Python, except for ESCAPE which is implemented in C++. Thus we focus on comparing the four algorithms that are implemented in Python. For big-O complexity analyses we refer the interested reader to each of the corresponding references for each algorithm.

Figure 9 shows results of this experiment. In all panels, the vertical axis shows the running time of the algorithm in seconds, and the horizontal axis presents the value of a structural parameter. Each row corresponds to one of the three following random graph models: ER (top row), BA (middle row), HG (bottom row). Each column corresponds to the variation of a single structural parameter, while keeping the other two fixed. Concretely, the left column shows how the number of eigenvalues r used for NBD and Lap affects their runtime on graphs with $n = 1000$ and $\langle k \rangle = 15$. The center column shows how varying the number of nodes n affects the runtime of all five distances while holding $r = 10$ and $\langle k \rangle = 15$ fixed. The right column shows how varying $\langle k \rangle$ affects the runtime while keeping $r = 10$ and $n = 1000$ fixed. Note that all panels are presented on a log-linear scale.
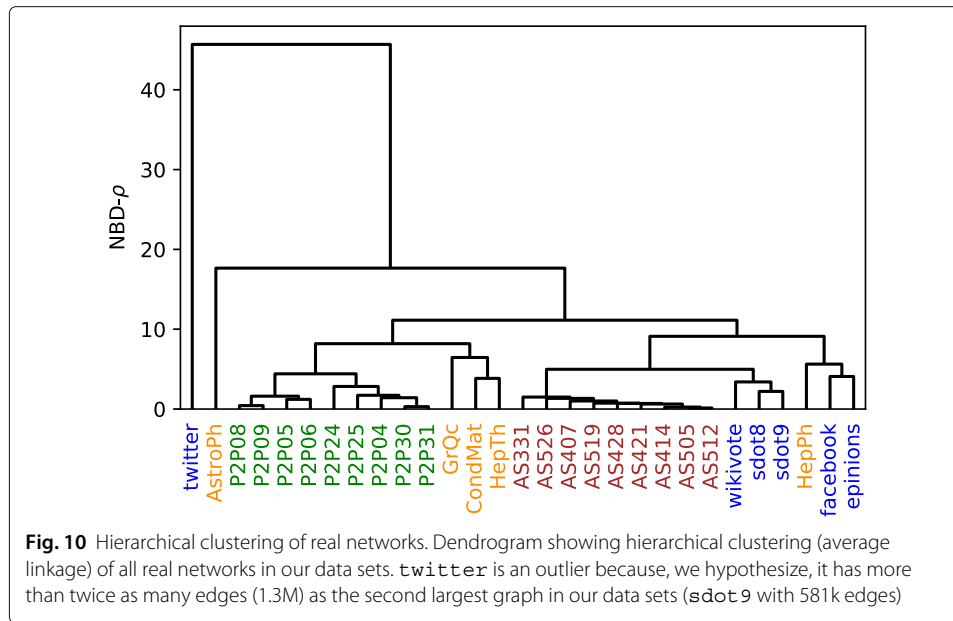
Fig. 9 Runtime comparison. We measure the time it takes to run each distance method on three different random graph models (ER, top; BA, middle; HG, bottom). We investigate the effect of varying three different parameters (number of eigenvalues, left; number of nodes, center; average degree, right). Lines show average, error bars show standard deviation

The left column of Fig. 9 shows evidence for the fact that the runtime scaling of NBD is very close to that of Lap, as claimed in "Computing B" section because the former uses the non-backtracking matrix, which is always four times the size of the Laplacian matrix. However, their runtimes do not differ by only a constant factor in the cases of ER and HG. We hypothesize that this is because the non-backtracking matrix has a more complicated structure than the Laplacian depending on the structure of the underlying graph (see "Spectral properties of B" section and especially Lemma 3). In all other panels we see that NBD and Lap present similar scaling behavior, but NBD is usually slower. The middle column presents the effect of varying the number of nodes in the graph on all five distance methods. NBD is consistently the third fastest method for HG and BA graphs, after ESCAPE and Lap, while GCD is consistently the slowest by two or three orders of magnitude in some cases. Note that the scaling behavior of all methods is fairly similar between BA and HG graphs, while for ER it is markedly different. When varying the number of nodes of ER graphs, for example, NBD and GCD seem to scale at the same rate, and take the about the same runtime, while S scales more poorly than all others. In all other panels, GCD seems to scale more poorly than other methods.

From the results presented in Fig. 9 we conclude that NBD is a viable alternative in the case of complex networks with heterogeneous degree distributions (modeled by BA and HG) when the application at hand requires comparison of counts of motifs or subgraphs such as non-backtracking paths of arbitrary length.

Nearest neighbors

In Fig. 10 we present a dendrogram plot computed on all real networks using a hierarchical clustering (average linkage) of NBD-ρ. We can see that all networks of data sets

Fig. 10 Hierarchical clustering of real networks. Dendrogram showing hierarchical clustering (average linkage) of all real networks in our data sets. `twitter` is an outlier because, we hypothesize, it has more than twice as many edges (1.3M) as the second largest graph in our data sets (`sdot9` with 581k edges)

`P2P` and `AS` are perfectly clustered with each other in one cluster, while five networks of the `social` data set (all except `twitter`) are clustered with each other, although they are broken up into two clusters. Data set `AS` is also broken up into three different clusters: one with `GrQc`, `CondMat`, and `HepTh`, while `HepPh` and `AstroPh` lie in their own cluster. To test the robustness of these clusters, we ran the same experiment (i) leaving one data set out each time, and (ii) randomly choosing the ordering of the input data sets. These variations are made to address the shortcomings of hierarchical clustering as some implementations may break ties in arbitrary ways. Neither of these variations affected the clusters reported in Fig. 10.

To better understand these clusters, we present in Table 3 the $k = 10$ networks that are closest to each of these data sets: `facebook`, `AstroPh`, and `twitter`. We show the nearest networks among all networks, including random graphs, as well as the nearest real networks only. In the case of `facebook`, we see that, even when including random

Table 3 For networks `facebook`, `AstroPh`, `twitter` we show the 10 networks closest to each, as determined by NBD-ρ, and report the distance score in parentheses

rank	facebook	AstroPh	twitter
1	epinions (4.08)	CondMat (11.33)	AstroPh (31.38)
2	HepPh (5.26)	epinions (13.72)	CondMat (40.11)
3	sdot9 (6.05)	facebook (14.18)	epinions (41.49)
4	sdot8 (6.24)	GrQc (14.71)	facebook (42.45)
5	wiki (6.56)	HepTh (14.84)	HepTh (42.60)
6	AS414 (8.98)	sdot9 (16.96)	HepPh (44.30)
7	AS505 (9.03)	HepPh (16.97)	GrQc (44.54)
8	AS512 (9.05)	sdot8 (17.37)	sdot9 (44.66)
9	AS421 (9.06)	AS407 (17.62)	wiki (45.64)
10	AS407 (9.16)	AS414 (17.68)	sdot8 (45.88)

Note that all social networks except for `twitter` are within the top 5 nearest networks to `facebook`, in spite of their imperfect clustering in Fig. 10. Likewise, all coauthorship networks are within the top 7 networks closest to `AstroPh`. Interestingly, `twitter` is also closest to other social networks, as well as some coauthorship networks, even though it is an outlier and the distances are far greater than others
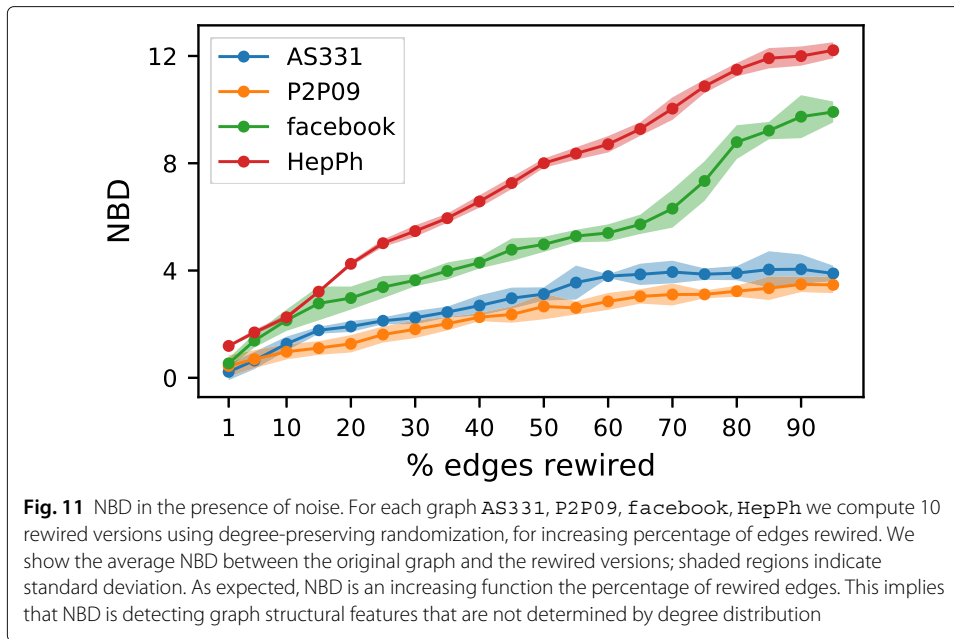
networks, `epinions` is the network closest to it. This is perhaps surprising since these two networks have substantially different values for the number of nodes, average degree, and average clustering coefficient (see Table 2). The networks closest to `facebook`, after `epinions`, all come from the `HG` random graph model, which is expected because this model has been shown to produce heavy-tailed degree distributions with high clustering, which both `facebook` and `epinions` present as well. When considering only real networks, all but one of the `social` networks are within the 10 nearest networks to `facebook`. In the case of `AstroPh`, the closest network to it is another coauthorship network, `CondMat`. Like in the previous case, it is close to many random graphs of model `HG`. Among the real networks, all other coauthorship graphs are in the nearest 10. Finally, `twitter` is the data set that lies farthest from all others. This may be due to the fact that it has substantially more edges than all others: it has 1.3M edges, while the second largest, `epinions`, has only 748k. We see that the closest random networks to `twitter` all come from the random model `WS`. This may be due to the fact that both twitter and `WS` have many large eigenvalues with zero imaginary part, so the earth mover distance between them will be small. We also observe that, even though twitter is far from all the networks in the dendrogram, all of the `social` graphs are within the 10 nearest networks to it.

From these observations we conclude that NBD is able to identify networks that are generated by similar mechanisms: all technological graphs – i.e., autonomous systems of the Internet (`AS`) and automatically generated peer-to-peer file sharing networks (`P2P`) – are perfectly clustered together, and all online social graphs are ranked near each other, even when some of these have vastly different elementary statistics such as number of nodes, number of edges, and average clustering coefficient.

Rewiring edges

We are interested in the behavior of NBD in the presence of noise. For this purpose, given a graph G, we compute G' by rewiring some of the edges of G in such a way that the degree distribution is kept intact (degree preserving randomization). We then compute the NBD between G and G'. For this experiment we use four networks, one from each of our real data sets: `AS331`, `P2P09`, `facebook`, and `HepPh`. We show our results in Fig. 11.

As expected, the NBD between a graph and its rewired versions grows steadily as the percentage of rewired edges increases. Note that `AS331` and `P2P09` seem to plateau when the percentage of rewired edges is 65% or larger, while `facebook` and `HepPh` have a slight jump close to 70%. Observe too that the variance of the NBD is kept fairly constant, which indicates that the distribution of non-backtracking cycle lengths in all rewired versions of each graph is very similar to each other. This may be due to the fact that edge rewiring destroys short cycles more quickly than long cycles. Furthermore, observe that for high percentages of rewired edges, the rewired graph is essentially a configuration model graph with the same degree distribution as the original. Therefore, by comparing the NBD values from Figs. 10 and 11, we can conclude that the NBD between these four networks and a random graph with the same degree distribution is larger than the average distance to other networks in the same data set (cf. Fig. 10 and discussion in previous section). We conclude that NBD is detecting graph structural properties – stored in the distribution of non-backtracking cycle lengths and encoded in the non-

Fig. 11 NBD in the presence of noise. For each graph `AS331`, `P2P09`, `facebook`, `HepPh` we compute 10 rewired versions using degree-preserving randomization, for increasing percentage of edges rewired. We show the average NBD between the original graph and the rewired versions; shaded regions indicate standard deviation. As expected, NBD is an increasing function the percentage of rewired edges. This implies that NBD is detecting graph structural features that are not determined by degree distribution

backtracking eigenvalues –, that are not solely determined by the degree-distribution of the graph.

NBED: Non-backtracking embedding dimensions

In this Section we discuss the potential of a second application of the eigendecomposition of the non-backtracking matrix in the form of a new embedding technique. In this case, instead of the eigenvalues, we use the eigenvectors. We propose the Non-Backtracking Embedding Dimensions (or NBED for short) as an edge embedding technique that assigns to each undirected edge of the graph two points in $\mathbb{R}^d$ – one for each orientation –, where d is the embedding dimension. We are able to find many interesting patterns that encode the structure of the network, which we then put to use for anomaly detection. While the visualizations found with NBED are certainly peculiar and rich in information, future work will prove essential in determining the full extent of possible applications of NBED.

Concretely, given a graph G with edge set E and directed edge set $\vec{E}$, consider a unit eigenvector v corresponding to the non-backtracking eigenvalue λ. Whenever λ is real, the entries of v are guaranteed to be real. Given the d eigenvectors $\{v^i\}_{i=1}^{i=d}$ corresponding to the d eigenvalues with largest magnitude, the embedding of an edge $(k, l) \in E$ is a point in $\mathbb{R}^{2d}$ whose entries are given by

$$\left(f\left(v_{k \to l}^1\right), f\left(v_{k \to l}^2\right), ..., f\left(v_{k \to l}^d\right), f\left(v_{l \to k}^1\right), f\left(v_{l \to k}^2\right), ..., f\left(v_{l \to k}^d\right) \right), \tag{6}$$

where f is defined as $f\left(v_{k \to l}^i\right) = \text{Re}\left(\lambda_i\right) \text{Re}\left(v_{k \to l}^i\right) - \text{Im}\left(\lambda_i\right) \text{Im}\left(v_{k \to l}^i\right)$ and Re, Im are the real and imaginary pats of a complex number, respectively. As mentioned previously, λ_1, the largest eigenvalue, is always guaranteed to be real and positive, and the entries of v^1 are also real and positive. Thus, $f\left(v_{k \to l}^1\right) = v_{k \to l}^1$ for every $k \to l$. Whenever any $\lambda_i, i = 2, 3, ...$, has non-zero imaginary part, the entries of v^i may also be complex numbers in general, and $f\left(v_{k \to l}^i\right)$ is simply a linear combination of the real and imaginary parts.
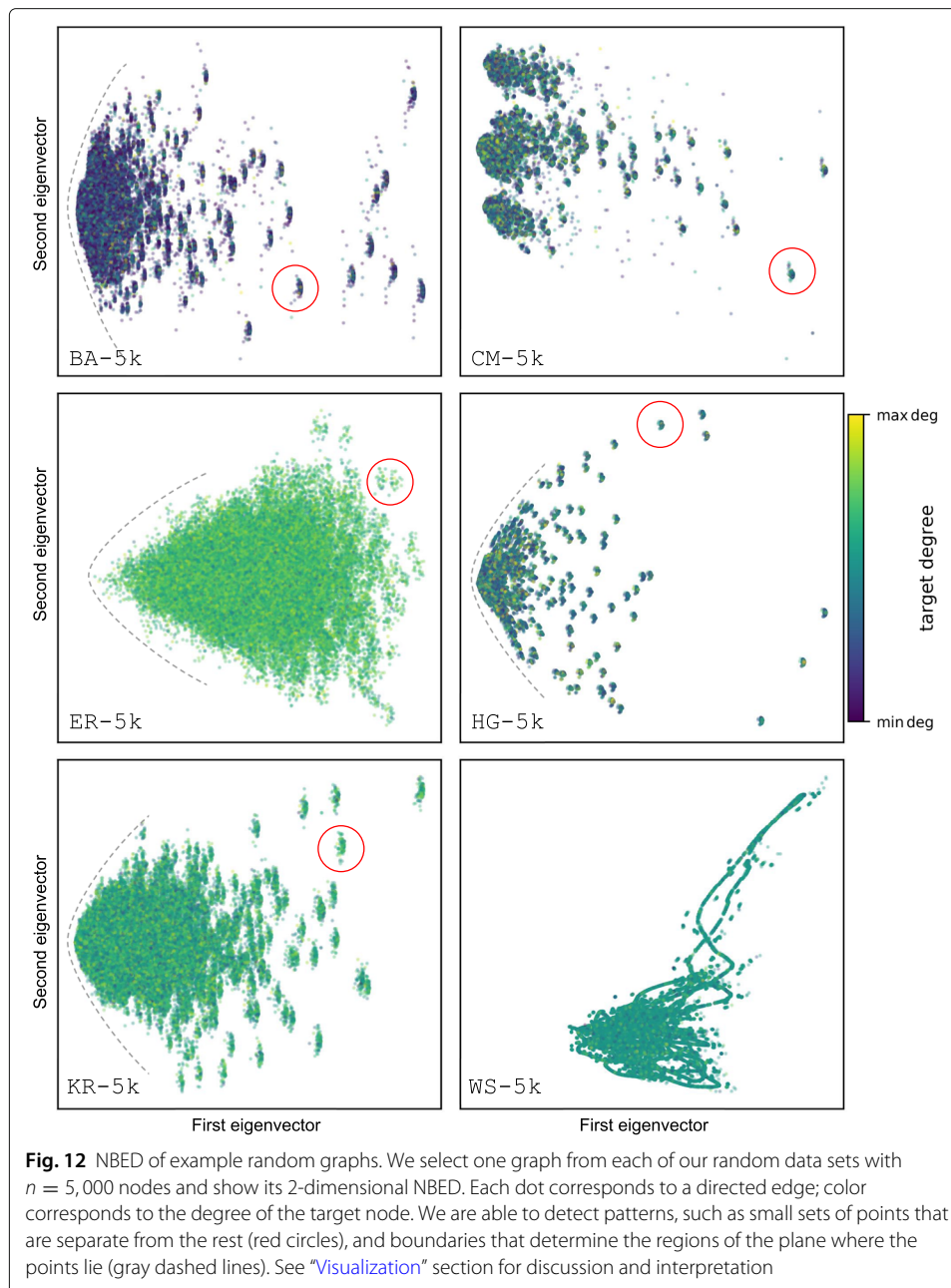
In the following we use the 2-dimensional NBED of real and random graphs. The advantages of this 2-dimensional edge embedding are as follows. First, the first and second eigenvectors of the non-backtracking matrix have been studied before and they can be interpreted in terms of edge centrality (Martin et al. 2014; Grindrod et al. 2018) and community detection (Krzakala et al. 2013; Bordenave et al. 2015; Kawamoto 2016) respectively. Second, NBED provides a deterministic graph embedding, as opposed to other methods that are stochastic due to their dependence on random sampling of subgraphs (Grover and Leskovec 2016; Perozzi et al. 2014) or due to stochastic optimization of some model (Wang et al. 2016; Cao et al. 2016). Third, NBED makes a distinction between the two possible orientations of an edge thus providing more information about it. We conjecture that these properties make NBED more robust to the presence of noise in the graph when compared to embeddings generated by popular models such as deep networks, which are in general not interpretable, are stochastic, and usually work on nodes rather than directed edges. Further, the 2-dimensional version of NBED is particularly important because our experimental evidence shows that the second non-backtracking eigenvalue of complex networks tends to be real and positive, which means that we can visualize these embeddings on the real plane without losing any information due to the linear combination f. We proceed to analyze the visual patterns we can identify in NBED visualizations; future research will be necessary for analytic characterizations of them.

Visualization

In Fig. 12 we present the 2-dimensional NBED of one network from each of our random data sets with $n = 5,000$ nodes (see "Data sets and base lines" section for description). Each edge (k, l) corresponds to two dots in this plot: $\left(v^1_{k \to l}, v^2_{k \to l}\right)$ and $\left(v^1_{l \to k}, v^2_{l \to k}\right)$. All graphs used in Fig. 12 have their two largest eigenvalues real and positive. We proceed to describe the patterns that we observe in these plots and how they relate to structural properties of the graph.

1   In all of the plots except for `WS-5k` we see small sets of dots (red circles) that are markedly separated from the bulk of the dots. Every one of these sets is made up of the embeddings of the directed edges $k \to l$ for a fixed l, and the horizontal position of this set corresponds to the degree of l. That is, the larger the degree of a fixed node l, the more likely all the embeddings of edges of the form $k \to l$ are to cluster together separate from the rest, and the further right this cluster will be.

2   Two of these sets corresponding to the same graph seem to have similar structures. That is, the relative positions of dots forming a cluster encircled in red is repeated for all such cluseters inside each graph. For example, the sets of `BA-5k` and `KR-5k` are much more vertically spread than those of `CM-5k` or `HG-5k`. We conclude that each of these sets have a particular internal structure that correlates with global graph structure, as it is repeated across sets. In `ER-5K`, the internal structure of each set is less well marked, owing to the fact that there are no global structural patterns in the ER model.

3   In `BA-5k`, `ER-5k`, `HG-5k`, and `KR-5k`, we find that the dots inhabit a certain region of the plane that is bounded by a roughly parabolic boundary (gray dashed lines). Furthermore, the cusp of this boundary always lies precisely at the origin.

Fig. 12 NBED of example random graphs. We select one graph from each of our random data sets with $n = 5,000$ nodes and show its 2-dimensional NBED. Each dot corresponds to a directed edge; color corresponds to the degree of the target node. We are able to detect patterns, such as small sets of points that are separate from the rest (red circles), and boundaries that determine the regions of the plane where the points lie (gray dashed lines). See "Visualization" section for discussion and interpretation

That is, there are never dots to the left of the origin, and the vertical spread of the dots grows at roughly the same rate both in the upper half-plane and the lower half-plane. Since the vertical axis corresponds to the entries of the second eigenvector v^2 which has been shown to describe community structure, we conjecture that the significance of this boundary is related to the existence of communities in the network. (See "Case Study: Enron emails" section for more discussion on this point.)

4. In Fig. 12, the color of the embedding of edge $k \rightarrow l$ corresponds to the degree of the source node k. In this way, we can use these plots to gather information about the degree distribution of the graph. For example, the plot for ER-5k looks like a noisy point cloud because the degree distribution is Poisson centered around

$\langle k \rangle = 15$, and the plot for `WS-5k` is very homogeneous in color because all nodes start with the same degree $k = 14$ and only a few of them are rewired. Furthermore, if one compares the color distribution of the plots for `BA-5k`, `CM-5k`, and `HG-5k`, one can differentiate that even though all have heavy-tailed degree distributions, their exponents must be different. Indeed, for `BA-5k` we have $\gamma = 3$ while the other two were generated with $\gamma = 2.3$.

5   From all previous considerations we can analyze the plot of `KR-5k` as a case study. First, it presents small sets of dots that are markedly separate from the bulk, and whose structure is similar to those of `BA-5k`. Second, its color distribution is closer to that of `ER-5k` than to any of the others. Third, it exhibits the boundary effect that we see in both `BA-5k` and `ER-5k`. From these observations we conclude that the stochastic Kronecker graph model is a "mix" between the Barabasi-Albert and Erdos-Renyi models, in the sense its plot has elements from both of the plots for BA and ER. Consider this in light of Fig. 1 where the graphs of these three data sets are clustered together, even when using the largest $r = 200$ non-backtracking eigenvalues of each graph. This means that the eigendecomposition of the non-backtracking matrix is detecting the similarities and differences between these three random graph models, which we can identify through a visual analysis of Figs. 1 and 12.

6   Lastly, we focus on the plot for `WS-5k` in Fig. 12. This plot shows none of the characteristics of the others, and in fact some of its points clearly fall on a path determined by a continuous curve. We hypothesize that this is due to the strong symmetry of the ring lattice from which it is generated. Studying the properties of the NBED of highly symmetric graphs and their randomized versions is a task we leave for a future work.

Case Study: Enron emails

In this Section we use both NBD and NBED to perform an analysis of the well-known Enron email corpus (Klimt and Yang 2004). This data set is comprised of email communications of Enron corporation employees in the years 2000–2002. From it, we form two different data sets of networks. In all networks each node represents an email address, and an edge between two email addresses represents that at least one email was sent from one account to the other. We aggregate these networks on a weekly basis, as well as on a daily basis, and we proceed to apply our methods NBD and NBED on them.

In the top panel of Fig. 13 we show the distance between a daily graph and the graph corresponding to a Sunday. In this plot we clearly see that NBD is able to detect the periodicity of email communications. In the bottom plot we show the distance between a weekly graph and the week starting on Monday January 1st 2001. In this case, we see that peaks in NBD values (or in r_0 values) correlate to several major events that occurred during the Enron bankruptcy scandal (Marks 2008; The Guardian 2006; The New York Times 2006). Not all important events are detected this way, however, as the event "1: Skilling becomes CEO" is not identified through high peaks in either NBD or r_0. Further research is necessary to clarify precisely what kind of events are being detected in this way.

In Fig. 14 we show the NBED of four weekly-aggregated graphs. The first three (top right, top left, bottom left) correspond to non-anomalous weeks, while the one on the bottom right, corresponding to the week of February 4th, 2002, is considered anomalous

Fig. 13 NBD of daily and weekly Enron graphs. *Top:* data aggregated into daily graphs and compared to one arbitrarily chosen milestone (Sunday, July 15th, 2001). Weekends are shaded blue. The periodicity of weekly communications is recovered. *Bottom:* Data aggregated into weekly graphs and compared to the week starting on Monday January 1st, 2001. Most peaks in values of NBD corresponding to events occurring during the Enron scandal

(cf. Fig. 13 where it is marked with a yellow circle and labeled "7: Lay resigns from board"). Most other weekly graphs have NBED plots that generally resemble the first three. In these plots we are able to see once again some of the features identified in Figure 12. First, they exhibit the sets of points clustered together separate from the rest with a similar internal structure to each other; like before, these correspond to the incoming edges

Fig. 14 NBED of select weekly-aggregated Enron graphs. Top left, top right, and bottom left correspond to non-anomalous weeks, according to Fig. 13. Bottom right corresponds to the anomalous week marked with a yellow circle in Fig. 13, labeled "1: Lay resigns from board". In all four we see some of the patterns that were identified in Fig. 12 and "Visualization" section
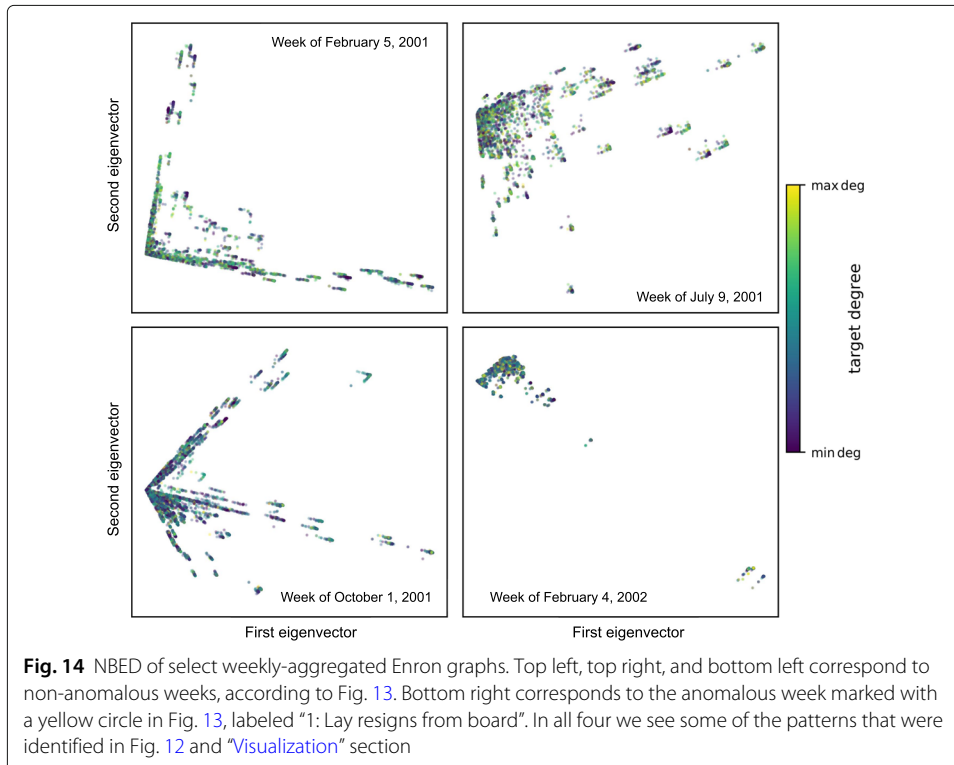
of high degree nodes. Second, the roughly parabolic boundaries of Fig. 12 have become sharply pointed with a cusp at the origin, from which two seemingly straight line segments are born in different directions. Third, the angle that these straight line segments form at the origin is different from graph to graph, and the lines are not symmetric with respect to the horizontal axis. As mentioned previously, the second eigenvector (e.g., the vertical axis in Figs, 12 and 14) is related to community structure. We hypothesize that the asymmetry seen in Enron graphs but not in random graphs is due to the presence of unevenly-sized communities; see Appendix 2. Lastly, the anomalous week of February 4th, 2002 (bottom right) looks decidedly different than the others, due to the presence of a few points very far apart from the bulk of the plot.

We have presented a visual analysis of the Enron corpus using NBD and NBED. Using these techniques, we are able to recover properties not only of the underlying network, but of the underlying data set as well, such as periodicity and temporal anomalies. Even though the analytical details of NBED require further consideration, we are still able to interpret the visualizations of Fig. 14 to mine important information about the underlying data set.

Discussion and conclusions

We have focused on the problem of deriving a notion of graph distance for complex networks based on the length spectrum function. We add to the repertoire of distance methods the Non-Backtracking Spectral Distance (NBD): a principled, interpretable, computationally efficient, and effective technique that takes advantage of the fact that one can interpret the non-backtracking cycles of a graph as its free homotopy classes. NBD is principled because it is backed by the theory of the length spectrum, which characterizes the 2-core of a graph up to isomorphism. It is interpretable because we can study its behavior in the presence of structural features such as hubs and triangles, and we can use the resulting geometric features of the eigenvalue distribution to our advantage. It is efficient relative to other similar methods that depend on the combinatorial enumeration of different kinds of subgraphs. Lastly, we have presented extensive experimental evidence to show that it is effective at discriminating between complex networks in various contexts, including visualization, clustering, and anomaly detection. Performance of NBD is better or comparable to other distance methods such as ESCAPE, S, GCD, and Lap; see "NBD: Experiments" section. We chose to compare against these methods because the first three depend on motif counts in one way or another, as NBD depends on non-backtracking cycle counts, and Lap depends on the spectral decomposition of a matrix representation of a graph, as NBD depends on the non-backtracking eigenvalues.

Motivated by the usefulness of NBD due to the connections with the homotopy of graphs and the spectrum of the non-backtracking matrix, we also presented a new embedding technique, Non-Backtracking Embedding Dimensions (or NBED for short) which provides a rich visualization full of interpretable patterns that describe the structural properties of a network. We have provided examples of these patterns as well as their application to anomaly detection. Further research will reveal the full potential of applications of NBED.

An implementation of NBD, NBED and our algorithm for computing the non-backtracking matrix, is available at (Torres 2018).

Limitations NBD relies on the assumption that the non-backtracking cycles contain enough information about the network. Accordingly, the usefulness of the NBD will decay as the 2-core of the graph gets smaller. For example, trees have an empty 2-core, and all of its non-backtracking eigenvalues are equal to zero. In order to compare trees, and more generally, those nodes outside the 2-core of the graph, the authors of (Durfee and Martin 2015) propose several different strategies, for example adding a "cone node" that connects to every other node in the graph. However, many real-world networks are not trees and we extensively showcased the utility of NBD on this class of networks.

The greatest limitation of NBED is that we are not able to provide rigorous derivations for the patterns we identify in "Visualization" section. Without a formal theory of the relationship between the eigenvectors of the non-backtracking matrix and the structural properties of graphs it is difficult to design algorithms that make use of these patterns automatically. Regardless, we have presented evidence for the usefulness of these patterns even with a visual analysis.

Future work There are many other avenues to explore in relation to how to exploit the information stored in the length spectrum and the fundamental group of a graph. As mentioned in Sec. 4, the major downside of the relaxed length spectrum $\mathcal{L}'$ is the fact that we lose information stored in the combinatorics of the fundamental group. That is, $\mathcal{L}'$ stores information about the frequency of lengths of free homotopy classes, but no information on their concatenation, i.e., the group operation in $\pi_1(G)$. One way to encapsulate this information is by taking into account not only the frequency of each possible length of non-backtracking cycles, but also the number of non-backtracking cycles of fixed lengths ℓ_1 and ℓ_2 that can be concatenated to form a non-backtracking cycle of length ℓ_3. It remains an open question whether this information can be computed using the non-backtracking matrix for all values of the parameters ℓ_1, ℓ_2, ℓ_3, and if so, how to do it efficiently. One alternative is to rely upon efficient motif counting (Pinar et al. 2017; Kolda et al. 2013).

A different research direction is to focus on the non-backtracking eigenvalues themselves, independently of the length spectrum theory. One standing question is to characterize the behavior of the eigenvalues after the network has been rewired. "Rewiring edges" section only scratches the surface of what can be said in this regard. However, spectral analysis of the non-backtracking matrix is exceedingly difficult due to the fact that it is asymmetric and non-normal and therefore most of the usual tools for spectral analysis are not applicable.

In this work, we have focused on introducing and exploiting novel theoretical concepts such as the length spectrum and the fundamental group to the study of complex networks. We are confident this work will pave the road for more research in topological and geometric data analysis in network science.

Appendix 1: UMAP parameter settings

In order to understand the visualizations of the non-backtracking eigenvalues in Fig. 1 we will now explain some of the features of the UMAP algorithm. For full detail we refer the interested reader to (McInnes et al. 2018).

UMAP stands for Uniform Manifold Approximation and Projection, authored by Leland McInnes, John Healy, and James Melville. Data are first represented in a high

dimensional euclidean space using Laplacian Eigenmaps (Belkin and Niyogi 2003), then they are approximated unifomly using fuzzy simplicial sets and patched together to form a manifold, and finally this manifold is projected to $\mathbb{R}^2$. This process reveals topological features of the data and provides flexibility for geometrical specifications to distinguish task-specific quantities of interest.

In this respect, UMAP provides a very flexible algorithm, which has various parameters. In the interest of reproducibility, we list our parameter choices of the subset of parameters that yield results of interest for network science, as they allow us to separate different random graph models from each other. A full explanation of these and other parameters is included in (McInnes et al. 2018) in Section 4.3. In Fig. 1 of this reference authors show a comparison of how visualizations change with the choice of parameters.

- `n_neighbors`: A weighted k-nearest neighbor graph is constructed from the initial data. The number of neighbors is set with this parameter. We used a value of 75.
- `metric`: Different metrics in the high dimensional space where the data are embedded can be specified. We found good overall results with the Canberra metric (shown in Fig. 1. The Chebyshev and Euclidean metrics cluster ER graphs together, for some specific values of the other parameters. However they make HG and CM clusters overlap. A clearly separated projection for the HG and CM graphs can be found using Correlation metric.
- `n_epochs`: Training optimization epochs. More epochs can provide better results, with the usual computational drawbacks. We use 1000 epochs.
- `min_dist`: Alters the minimum distance between embedded points. Smaller values increase clustering, larger values present a more uniformly distributed visualiaztion. We use a distance of 0.01.
- `repulsion_strength`: Values above 1 increase the weight of negative samples, i.e., the importance of the distances between far-apart data points. We use a value of 10.
- `negative_sample_rate`: Sets the number of negative samples to be selected per each positive sample in the optimization process. We use 50 negative samples per positive sample.

We were not able to find a specific set of parameters that improved upon the visualization shown in Fig. 1, though it is perhaps the interplay between `min_dist` and `repulsion_strength` what causes ER graphs to split into two different clusters.

Appendix 2: Asymmetric NBED embeddings

As mentioned in "Case Study: Enron emails" section, the NBED of Enron graphs present a feature that we do not see in any of the random graphs we handled for this work. Namely, the NBED of Enron graphs are asymmetric with respect to the horizontal axis (Fig. 14), whereas the NBED of all other random graphs (Fig. 12) are generally symmetric. In "Case Study: Enron emails" section we hypothesized that this is due to the fact that Enron graphs contain communities. Here, we present some evidence for this hypothesis.

In Fig. 15 we show the NBED of five random graphs generated from the stochastic block model as follows. Each graph has $n = 150$ nodes and was generated with two blocks. In all cases, the generated graphs were connected and the edge density joining the two different blocks was 0.01. We vary the size and internal density of each block to yield

Fig. 15 NBED of stochastic block model graphs. Embeddings of five random graphs generated from the stochastic block. Each graph is connected, has $n = 150$ nodes and was generated with two blocks; the edge density joining the two different blocks is 0.01. We present five different realizations, depending on the relative sizes and densities of the blocks
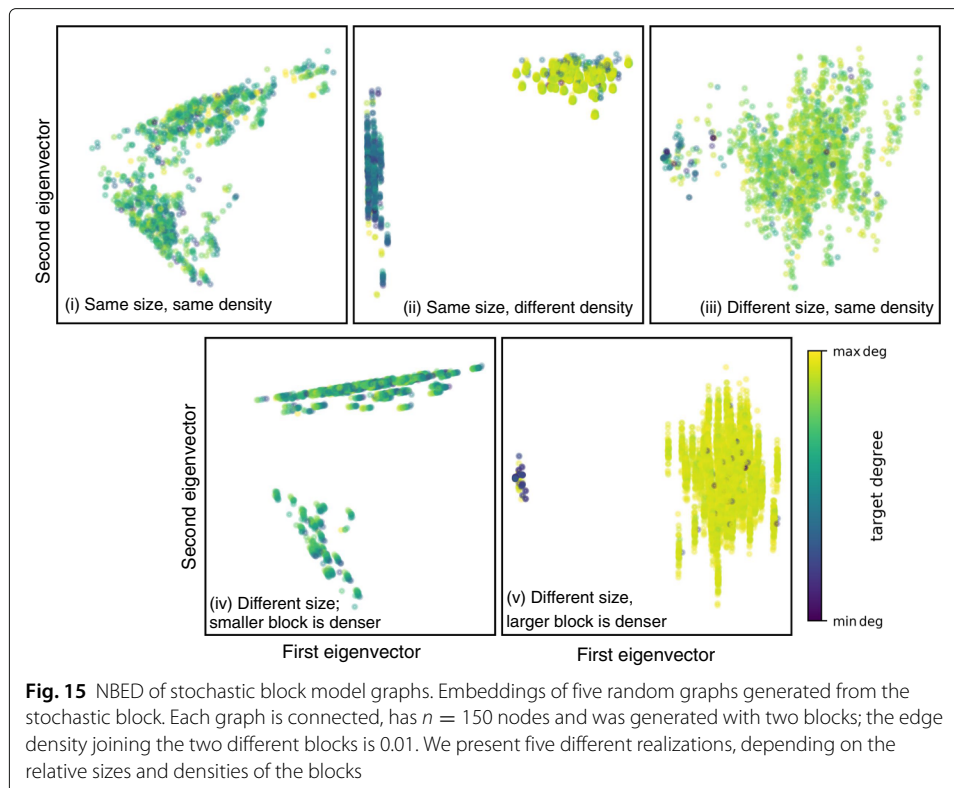
five different realizations: (i) two blocks of the same size and same internal density equal to 0.1, (ii) blocks of same size, one with density 0.5 and the other with density 0.1, (iii) blocks of different size, with the same density 0.1, (iv) blocks of different size where the larger block has density 0.1 and the smaller has density 0.5, and (v) blocks of different size where the larger block has density 0.5 and the smaller has density 0.1. In cases (ii) and (iv), we see NBED embeddings that are markedly asymmetrical with respect to the horizontal axis. Furthermore, case (iv) in particular is visually reminiscent of the Enron embeddings. Further work will elucidate more systematically the role of differently-sized commmunities with different densities on the eigenvalues and eigenvectors of the non-backtracking matrix.

Endnotes

[1] The definition presented here is also known as *marked* length spectrum. An alternative definition of the (unmarked) length spectrum does not depend on π_1; see for example (Leininger et al. 2007).

[2] This follows from G being homotopy equivalent to a bouquet of k circles, where k is the rank of the fundamental group of G. The universal covering of a bouquet of circles is contractible, which is equivalent to the space being aspherical. (See (Hatcher 2017).)

[3] Other authors reserve the use of the term *cycle* for special cases of closed paths such as the set of *simple cycles*, which are closed paths that do not intersect each other. In this work we use *cycle* and *closed path* interchangeably.

[4] In (Constantine and Lafont 2018), the authors need an isomorphism between the fundamental group of the spaces that are being compared, which is also computationally prohibitive.

[5] Here we choose both `EMD` and $r = r_0$ based on the experimental evidence of Fig. 5. An early version of NBD used `Euclidean` and a value of r set to a constant (Torres et al. 2018). Following that version, other authors have independently proposed using `EMD` (Mellor and Grusovin 2018).

[6] https://www.ipam.ucla.edu/programs/longprograms/culture-analytics/

Authors' contributions
This manuscript would not exist without LT; he was the lead contributor in all aspects of the manuscript. PSS introduced the mathematical concept of length spectra and topological data analysis to LT and TER. LT and PSS were the major contributors to the topological graph analysis and graph distance in the manuscript. LT and TER were the major contributors to the graph embedding and experimental design in the manuscript. LT developed the code and conducted all the experiments. LT was the lead contributor in writing the manuscript. All authors read and approved the final manuscript.

Availability of data and materials
With this work we also release publicly available source code that implements NBD, NBED and the related matrix computations (Torres 2018).

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Network Science Institute, Northeastern University, 360 Huntington Ave, 02115-5000, Boston, MA, USA. [2] Department of Mathematics, University of California, Santa Barbara, 552 University Rd, 93117 Isla Vista, CA, USA. [3] Instituto de Matemáticas, Universidad Nacional Autónoma de México, Área de la Investigación Científica, Circuito Exterior, Ciudad Universitaria, 04510, CDMX Ciudad de México, México. [4] Khoury College of Computer Sciences, 360 Huntington Ave, 02115-5000, Boston, MA, USA.

References
Aldecoa R, Orsini C, Krioukov D (2015) Hyperbolic graph generator. Comput Phys Commun 196:492–6

Angel O, Friedman J, Hoory S (2015) The non-backtracking spectrum of the universal cover of a graph. Trans Amer Math Soc 367(6):4287–318

Bagrow JP, Bollt EM (2018) An information-theoretic, all-scales approach to comparing networks. Preprint, arXiv:1804.03665 [cs.SI]

Barabási A-L, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–12

Bass H (1992) The Ihara-Selberg zeta function of a tree lattice. Internat J Math 3(6):717–97

Batagelj V, Zaversnik M (2011) Fast algorithms for determining (generalized) core groups in social networks. Adv Data Anal Classi 5(2):129–45

Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–96

Bento J, Ioannidis S (2018) A family of tractable graph distances. In: Proceedings of the 2018 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, San Diego, CA. pp 333–41

Berlingerio M, Koutra D, Eliassi-Rad T, Faloutsos C (2013) Network similarity via multiple social theories. In: Advances in Social Networks Analysis and Mining (ASONAM). ACM, Niagara, ON. pp 1439–40

Bollobás B (2001) Random Graphs, 2nd edn. In: Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge; New York

Bordenave C, Lelarge M, Massoulié L (2015) Non-backtracking spectrum of random graphs: community detection and non-regular Ramanujan graphs. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS) 2015. IEEE. pp 1347–57

Cao S, Lu W, Xu Q (2016) Deep neural networks for learning graph representations. In: Schuurmans D, Wellman MP (eds). Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016. AAAI, Phoenix. pp 1145–52

Chowdhury S, Mémoli F (2017) Distances and isomorphism between networks and the stability of network invariants. Preprint, arXiv:1708.04727 [cs.DM]

Chowdhury S, Mémoli F (2018) The metric space of networks. Preprint, arXiv:1804.02820 [cs.DM]

Clauset A, Tucker E, Sainz M The Colorado Index of Complex Networks. https://icon.colorado.edu/. Accessed 19 June 2018

Constantine D, Lafont J-F (2018) Marked length rigidity for one-dimensional spaces. J Topol Anal. https://doi.org/10.1142/S1793525319500250

Cooper Y (2009) Properties determined by the Ihara zeta function of a graph. Electron J Combin 16(1):14–84

Durfee C, Martin K (2015) Distinguishing graphs with zeta functions and generalized spectra. Linear Algebra Appl 481:54–82

Erdös P, Rényi A (1960) On the evolution of random graphs. Publ Math Inst Hung Acad Sci 5:17

Estrada E (1996) Spectral moments of the edge adjacency matrix in molecular graphs, 1. definition and applications to the prediction of physical properties of alkanes. J Chem Inf Comp Sci 36(4):844–9

Goyal P, Ferrara E (2018) Graph embedding techniques, applications, and performance: A survey. Knowl-Based Syst 151:78–94

Grindrod P, Higham DJ, Noferini V (2018) The deformed graph Laplacian and its applications to network centrality analysis. SIAM J Matrix Anal Appl 39(1):310–41

Grover A, Leskovec J (2016) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). In: Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R (eds). ACM. pp 855–64

Gulikers L, Lelarge M, Massoulié L (2017) Non-backtracking spectrum of degree-corrected stochastic block models. In: 8th Innovations in Theoretical Computer Science (ITCS). ITCS 2017-8th Innovations in Theoretical Computer Science, Berkeley, CA. pp 44–14427

Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: Methods and applications. IEEE Data Eng Bull 40(3):52–74

Hashimoto K (1989) Zeta functions of finite graphs and representations of p-adic groups. In: Automorphic Forms and Geometry of Arithmetic Varieties Vol. 15. pp 211–80

Hatcher A (2017) Algebraic Topology. Cambridge University Press, Cambridge; New York

Jiang F, He L, Zheng Y, Zhu E, Xu J, Yu PS (2018) On spectral graph embedding: A non-backtracking perspective and graph approximation. In: Proceedings of the 2018 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, San Diego, CA. pp 324–32

Kawamoto T (2016) Localized eigenvectors of the non-backtracking matrix. J Stat Mech Theory Exp 2:12. 023404

Klimt B, Yang Y (2004) The enron corpus: A new dataset for email classification research. In: European Conference on Machine Learning. Springer, Berlin, Heidelberg. pp 217–226

Kolda TG, Pinar A, Seshadhri C (2013) Triadic measures on graphs: The power of wedge sampling. In: Proceedings of the 13th SIAM International Conference on Data Mining (ICDM). Society for Industrial and Applied Mathematics, Austin. pp 10–8

Koutra D, Shah N, Vogelstein JT, Gallagher B, Faloutsos C (2016) DeltaCon: Principled massive-graph similarity function with attribution. TKDD 10(3):28–12843

Krioukov D, Papadopoulos F, Kitsak M, Vahdat A, Boguñá M (2010) Hyperbolic geometry of complex networks. Phys Rev E 82:036106

Krzakala F, Moore C, Mossel E, Neeman J, Sly A, Zdeborová L, Zhang P (2013) Spectral redemption in clustering sparse networks. Proc Natl Acad Sci USA 110(52):20935–40

Kunegis J (2013) KONECT: The Koblenz network collection. In: 22nd International World Wide Web Conference, (WWW). ACM, Rio de Janeiro, Brazil. pp 1343–50

Lang S (2004) Linear Algebra, 3rd edn. Springer, New York

Leininger CJ, McReynolds DB, Neumann WD, Reid AW (2007) Length and eigenvalue equivalence. Int Math Res Not IMRN 2007(24):135

Leskovec J, Chakrabarti D, Kleinberg JM, Faloutsos C, Ghahramani Z (2010) Kronecker graphs: An approach to modeling networks. J Mach Learn Res 11:985–1042

Leskovec J, Huttenlocher DP, Kleinberg JM (2010) Proceedings of the 28th International Conference on Human Factors in Computing Systems. In: Mynatt ED, Schoner D, Fitzpatrick G, Hudson SE, Edwards WK, Rodden T (eds). CHI 2010, Atlanta, Georgia. pp 1361–70. April 10-15, 2010

Leskovec J, Kleinberg JM, Faloutsos C (2005) Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. In: Grossman R, Bayardo RJ, Bennett KP (eds). ACM, Chicago, Illinois. pp 177–87. August 21-24, 2005

Leskovec J, Kleinberg JM, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. TKDD 1(1):2

Leskovec J, Krevl A (2014) SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. Accessed 9 Feb 2019

Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Math 6(1):29–123

Marks R (2008) Enron Timeline. http://www.agsm.edu.au/bobm/teaching/BE/Enron/timeline.html. Accessed 2018-06-06

Martin T, Zhang X, Newman MEJ (2014) Localization and centrality in networks. Phys Rev E 90:052808

McAuley JJ, Leskovec J (2012) Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a Meeting Held December 3-6, 2012. In: Bartlett PL, Pereira FCN, Burges CJC, Bottou L, Weinberger KQ (eds). Neural information processing systems Foundation, Lake Tahoe, Nevada. pp 548–56

McInnes L, Healy J, Melville J (2018) Umap: Uniform manifold approximation and projection for dimension reduction. Preprint arXiv:1802.03426

Mellor A, Grusovin A (2018) Graph comparison via the non-backtracking spectrum. Preprint arXiv:1812.05457

Munkres JR (2000) Topology, 2nd edn. Prentice Hall, Englewood Cliffs, NJ

Newman MEJ (2003) The structure and function of complex networks. SIAM Rev. 45(2):167–256

Onnela J-P, Fenn DJ, Reid S, Porter MA, Mucha PJ, Fricker MD, Jones NS (2012) Taxonomies of networks from community structure. Phys Rev E 86:036104

Perozzi B, Al-Rfou R, Skiena S (2014) The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (KDD). In: Macskassy SA, Perlich C, Leskovec J, Wang W, Ghani R (eds). ACM. pp 701–10

Pinar A, Seshadhri C, Vishal V (2017) ESCAPE: efficiently counting all 5-vertex subgraphs. In: Proceedings of the 26th International Conference on World Wide Web (WWW) 2017. ACM, Perth. pp 1431–40. April 3-7, 2017

Preciado VM, Jadbabaie A, Verghese GC (2013) Structural analysis of Laplacian spectral properties of large-scale networks. IEEE Trans Automat Contr 58(9):2338–43

Ren P, Wilson RC, Hancock ER (2011) Graph characterization via Ihara coefficients. IEEE T Neural Nerwor 22(2):233–45

Richardson M, Agrawal R, Domingos PM (2003) Trust management for the semantic web. In: Fensel D, Sycara KP, Mylopoulos J (eds). The Semantic Web - ISWC 2003, Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003, Proceedings. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg Vol. 2870. pp 351–68

Rosenberg A, Hirschberg J (2007) Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), June 28-30, 2007. In: Eisner J (ed). Association for Computational Linguistics, Prague. pp 410–20

Rubner Y, Tomasi C, Guibas LJ (1998) A metric for distributions with applications to image databases. In: ICCV. IEEE. pp 59–66

Saade A, Krzakala F, Zdeborová L (2014) Spectral density of the non-backtracking operator on random graphs. EPL (Europhys Lett) 107(5):50005

Schieber TA, Carpi L, Díaz-Guilera A, Pardalos PM, Masoller C, Ravetti MG (2017) Quantification of network structural dissimilarities. Nat Commun 8:13928

Seshadhri C, Pinar A, Kolda TG (2013) An in-depth analysis of stochastic Kronecker graphs. J ACM 60(2):13–11332

Soundarajan S, Eliassi-Rad T, Gallagher B (2014) A guide to selecting a network similarity method. In: Proceedings of the 2014 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, Philadelphia, PA. pp 1037–45

Terras A (2011) Zeta Functions of Graphs: A Stroll Through the Garden. In: Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, Cambridge; New York. p 239

The Guardian (2006) Timeline: Enron. https://www.theguardian.com/business/2006/jan/30/corporatefraud.enron. Accessed 2018-06-06

The New York Times (2006) Timeline: A chronology of Enron Corp. https://www.nytimes.com/2006/01/18/business/worldbusiness/timeline-a-chronology-of-enron-corp.html. Accessed 2018-06-06

Torres L (2018) SuNBEaM: Spectral Non-Backtracking Embedding And pseudo-Metric. GitHub. https://github.com/leotrs/sunbeam. Accessed 5 Mar 2019

Torres L, Suarez-Serrato P, Eliassi-Rad T (2018) Graph distance from the topological view of non-backtracking cycles. arXiv preprint arXiv:1807.09592

von Luxburg U (2007) A tutorial on spectral clustering. Stat Comput 17(4):395–416

Wang D, Cui P, Zhu W (2016) Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). In: Krishnapuram B, Shah M, Smola AJ, Aggarwal CC, Shen D, Rastogi R (eds). ACM. pp 1225–34

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440

Wood PM, Wang K (2017) Limiting empirical spectral distribution for the non-backtracking matrix of an Erdös-Rényi random graph. Preprint, arXiv:1710.11015 [math.PR]

Yaveroğlu ÖN, Malod-Dognin N, Davis D, Levnajic Z, Janjic V, Karapandza R, Stojmirovic A, Pržulj N (2014) Revealing the hidden language of complex networks. Sci Rep 4:4547

Yaveroğlu ÖN, Milenković T, Pržulj N (2015) Proper evaluation of alignment-free network comparison methods. Bioinformatics 31(16):2697–704

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.