nature communications

Article

https://doi.org/10.1038/s41467-024-52355-w

Network community detection via neural embeddings

Received: 28 August 2023

Accepted: 3 September 2024

Published online: 01 November 2024

Check for updates

Sadamori Kojaku[1,2], Filippo Radicchi[2], Yong-Yeol Ahn[2] & Santo Fortunato[2]

Recent advances in machine learning research have produced powerful neural graph embedding methods, which learn useful, low-dimensional vector representations of network data. These neural methods for graph embedding excel in graph machine learning tasks and are now widely adopted. However, how and why these methods work—particularly how network structure gets encoded in the embedding—remain largely unexplained. Here, we show that node2vec—shallow, linear neural network—encodes communities into separable clusters better than random partitioning down to the information-theoretic detectability limit for the stochastic block models. We show that this is due to the equivalence between the embedding learned by node2vec and the spectral embedding via the eigenvectors of the symmetric normalized Laplacian matrix. Numerical simulations demonstrate that node2vec is capable of learning communities on sparse graphs generated by the stochastic blockmodel, as well as on sparse degree-heterogeneous networks. Our results highlight the features of graph neural networks that enable them to separate communities in the embedding space.

Networks represent the structure of complex systems as sets of nodes connected by edges[1–3] and are ubiquitous across diverse domains, including social sciences[4,5], transportation[6,7], finance[8,9], science of science[10,11], neuroscience[12,13], and biology[14–16]. Networks are complex, high-dimensional, and discrete objects, making it highly non-trivial to obtain useful representations of their structure. For instance, recommendation systems for social networks typically require informative variables (or "features") that capture the most important structural characteristics. Often, these features are designed through trial and error, and may not be generalizable across networks.

Graph embeddings automatically identify useful structural features for network elements, most commonly for the nodes[17,18]. Each node is represented as a point in a compact and continuous vector space. Such a vector representation enables the direct application of powerful machine learning methods, capable of solving various tasks, such as visualization[19,20], clustering[21,22], and prediction[18,23,24]. This representation can facilitate the operationalization of abstract concepts using vectorial operations[20,25–28]. Graph embeddings have been studied in various contexts. For example, spectral embedding stems from the spectral analysis of networks[17,29]. A closely related formulation is matrix factorization[30,31]. Recent years have witnessed a substantial shift towards a new paradigm of graph embeddings based on neural networks[20,22,32–40], which have demonstrated remarkable effectiveness across many computational tasks[23,34,35,38,39,39,40]. Yet, due to the inherent black-box nature of neural networks, how and why these methods work is still largely unknown; we lack a clear understanding of the process of encoding certain network structures onto embeddings.

One of the fundamental and ubiquitous features of networks is community structure, i.e., the existence of cohesive groups of nodes, characterized by a density of within-group edges that is higher than the density of edges between them[41–43]. In practice, neural graph embedding methods are widely used to discover communities from networks[26,31,34,38].

The stochastic block model (SBM) is a basic generative model of networks with community structure[44,45] and is regularly used as a benchmark for community detection algorithms. Some community detection methods are able to correctly classify all nodes into communities in large and dense networks generated by the SBM, provided

[1]School of Systems Science and Industrial Engineering, Binghamton University, Binghamton, NY, USA. [2]Center for Complex Networks and Systems Research, Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA. e-mail: santo@iu.edu

that the average degree increases as the number of nodes increases[21,46–50]. However, most networks of interest in applications are sparse[1,51], in that their average degree is usually much smaller than the network size. The task of community detection is particularly hard on sparse networks. For instance, the performance of many spectral methods significantly worsens as the graph gets sparser[52,53], which has led to the development of remedies such as non-backtracking walks[52–54] and consensus clustering[55]. However, it remains unclear how neural graph embeddings perform on sparse networks, how much edge sparsity hampers their ability to detect communities, and how they fare for traditional clustering techniques, especially spectral methods.

Here, we prove that graph embedding methods based on a shallow neural network without non-linear activation—such as DeepWalk[38], LINE[39], and node2vec[34]—can resolve communities all the way down to the information-theoretical limit on graphs generated by the SBM[56]. Our results imply that two common components of deep learning—multiple deep layers and non-linear activation—are not necessary to achieve the optimal limit of community detectability. Numerical experiments demonstrate that the communities embedded by node2vec can be effectively identified by the K-means algorithm, with accuracy close to the performance of the optimal belief propagation (BP) method[56] when the true number of communities is given to the K-means and BP. Additional numerical tests reveal that node2vec is also able to learn communities in the presence of heterogeneity of degree and community size. In this case, the two-step approach, combining embedding and clustering, is underperforming in certain settings, but this may be due to the fact that K-means clustering struggles when clusters have widely different sizes. We expect that addressing this shortcoming of K-means would lead to much better results.

Our work might help to inform powerful community detection algorithms and improve our theoretical understanding of clustering via neural embeddings. The code to reproduce all the results is available at ref. 57.

Results

Planted partition model

We first consider the standard setting studied in papers concerning community detectability[52,53,58]. We focus on undirected and unweighted networks with community structure generated according to the planted partition model (PPM)[59], a special case of the SBM where nodes are divided into q equal-sized communities, and two nodes are connected with probability p_{in} if they are in the same community and with probability p_{out} if they are in different communities. We assume that the networks are sparse, i.e., p_{in} and p_{out} are inversely proportional to the number n of nodes. The average degree $\langle k\rangle$ and the ratio of edge probabilities $p_{\mathrm{in}}/p_{\mathrm{out}}$ do not depend on n. We specify the edge probabilities via the mixing parameter $\mu = np_{\mathrm{out}}/\langle k\rangle$. The mixing parameter indicates how blended communities are with each other. As $\mu \to 0$, communities are well separated and easily detectable. For larger values of μ, community detection becomes harder. For $\mu = 1$, which corresponds to $p_{\mathrm{in}} = p_{\mathrm{out}}$, the network is an Erdős-Rényi random graph and, as such, has no community structure. We note that the mixing parameter μ is slightly different from the traditional mixing parameter μ_{LFR} used in the Lancichinetti-Fortunato-Radicchi (LFR) benchmark, which is defined as $\mu_{\mathrm{LFR}} = (1-\frac{1}{q})np_{\mathrm{out}}/\langle k\rangle$. The difference between μ and μ_{LFR} is negligible for large q.

Detectability limit of communities

The goal of community detection in the PPM is to recover the block membership of the model based on the structure of the specific networks generated by it. When communities are well separated, an algorithm is likely to recover these communities perfectly. However, as the number of inter-community edges increases, thereby reducing the difference between the densities of inter-community and intra-community edges, the algorithm may fail to correctly classify some nodes, and eventually, communities cannot be detected better than random guessing. The level of community mixing above which no algorithm can recover communities better than random guessing is the information-theoretic detectability limit[56,58].

Operationally, with the PPM, the level of mixing is quantified by μ. Communities are present for all μ-values in the range $[0, 1)$, because the edges are more densely distributed within communities than between them. In the regime above the information-theoretical limit (i.e., $\mu^* \le \mu < 1$), communities are not detectable because their inter-community/intra-community edge densities are indistinguishable from the corresponding edge densities of random partitions.

Detectability limit of node2vec

We first give a high-level description of our derivation of the algorithmic detectability limit for node2vec. We note that our derivation can be directly applied to other neural graph embeddings, such as DeepWalk[38] and LINE[39]. See the Methods section for the step-by-step derivations.

Our analysis is based on the fact that node2vec generates its embedding by effectively factorizing a matrix when the number of dimensions is sufficiently large[30]. This insight enables us to study node2vec as a spectral method (see Methods). Spectral algorithms identify communities by computing the eigenvectors associated with the largest or smallest eigenvalues of a reference operator, such as the combinatorial and normalized Laplacian matrices. When using eigenvectors to represent the network in vector space, nodes in the same community are projected onto points in space lying close to each other so that a data clustering algorithm can separate them[17].

The existence of such localized eigenvectors can be inferred by analyzing the spectrum of the reference operator using random matrix theory. For instance, this approach has been applied to determine the detectability limit of the normalized Laplacian matrix generated by the PPM[60]. We find that, under some mild conditions, the spectrum of the node2vec matrix is equivalent to that of the normalized Laplacian matrix. Hence, the detectability limit of node2vec matches that of the spectral embedding with the normalized Laplacian matrix[60]:

$$\mu^*_{\mathrm{n2v}} = \mu^* = 1 - \frac{1}{\sqrt{\langle k\rangle}}. \tag{1}$$

See Supporting Information Section 2 for the expression of the detectability limit in terms of the mixing parameter μ. This threshold exactly corresponds to the information-theoretical detectability limit μ^* of the PPM[55,58]. In other words, node2vec has the ability to detect communities down to the information-theoretic limit in principle. However, like in the case of spectral modularity maximization[58], our analysis is only valid when the average degree is sufficiently large. Nevertheless, as we shall see, our numerical simulations show that node2vec performs well even if the average degree is small.

Experiment setup

As baselines, we use two spectral embedding methods whose detectability limit matches the information-theoretical one: spectral modularity maximization[58] and the spectral embedding based on the normalized Laplacian matrix (Laplacian EigenMap)[61]. In addition, we use two other neural embeddings, DeepWalk[38] and LINE[39]. DeepWalk and LINE share the same architecture as node2vec but are trained with different objective functions[30,62]. Furthermore, we employ the spectral algorithm based on the leading eigenvectors of the non-backtracking matrix, which reaches the information-theoretical limit even in the sparse case for networks generated by the PPM[52]. For all embedding methods, we set the number of dimensions, C, to 64. Finally, we employ two community detection algorithms: statistical inference of

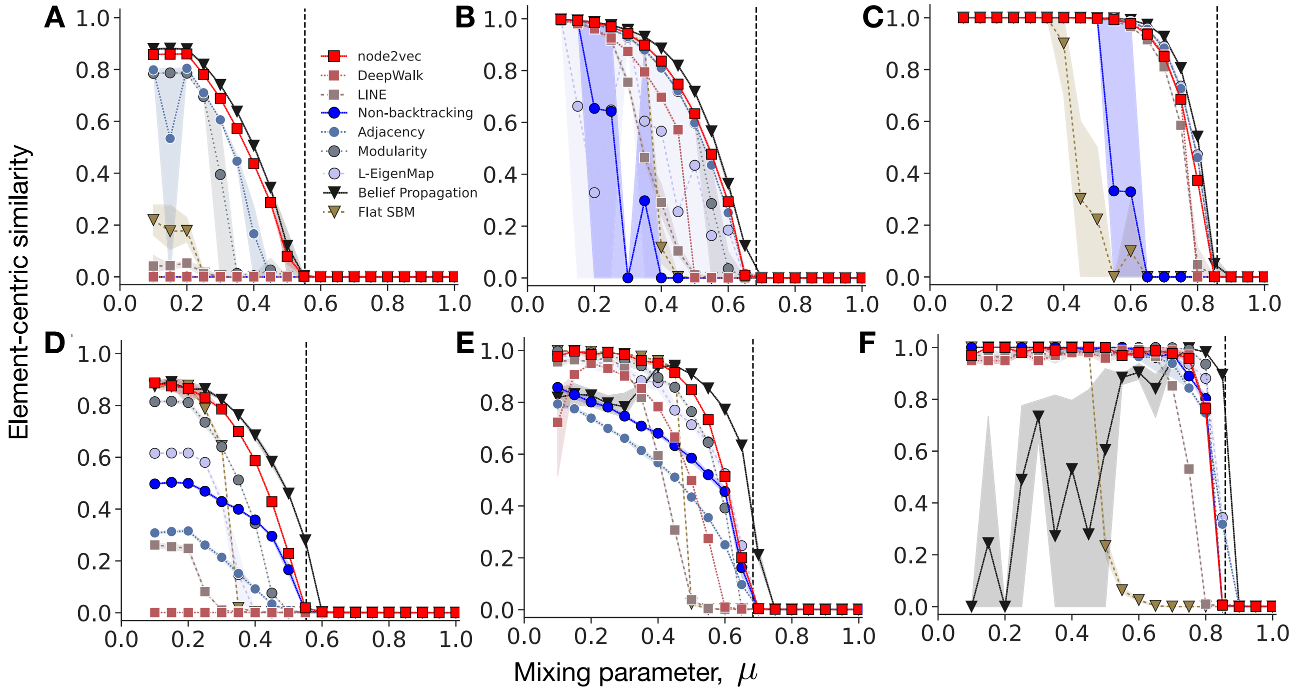

Fig. 1 | Performance of community detection methods on PPM networks. We generated networks with $n = 10^5$ nodes, different edge sparsity ($\langle k \rangle = 5$ in (**A**, **D**), $\langle k \rangle = 10$ in (**B**, **E**), $\langle k \rangle = 50$ in (**C**, **F**), and the different number of communities ($q = 2$ for **A**–**C** and $q = 50$ for **D**–**F**). The dashed vertical line indicates the theoretical detectability limit μ^* given by (1): communities are detectable (i.e., $S > 0$), in principle, below μ^*. Spectral embedding methods detect communities up to the theoretical limit for dense networks (**C**, **F**), supporting the detectability limit derived from previous studies[58,60]. However, for sparse networks, they fall short even at low μ-values (**A**, **D**). node2vec and the spectral embedding based on the non-backtracking matrix outperform other spectral methods, with the performance curves close to that of the BP algorithm. Note that even the BP algorithm falls short of the exact recovery of some easily detectable communities in the case of $q = 50$ communities, with the initial parameters set according to the ground-truth communities. The error bands represent the 90% confidence interval by a bootstrapping with 10^4 resample.

the microscopic degree-corrected SBM[44], and the BP algorithm[56]. The BP algorithm is theoretically optimal for PPM networks and serves as an ideal baseline for assessing graph embeddings. However, achieving optimal performance with BP in practice requires non-trivial parameter tuning. Therefore, we initialized the BP using the information about the true communities, namely the number of nodes in each true community and the number of edges between the communities. See Supporting Information Section 4 for the parameter choices of the models and the implementations we used.

Community detection via graph embedding is a two-step process:

- First, the network is embedded, which yields a projection of nodes onto points in a vector space.
- Second, the points are divided into groups using a data clustering method (e.g., *K*-means clustering).

Thus, the performance of community detection depends on both the quality of the embedding and the performance of the subsequent data clustering procedure. We use the *K*-means clustering algorithm in the second step. We set the number of clusters to the number of true communities, run the *K*-means algorithm 10 times with different random seeds, and select the best clustering in terms of the objective of the *K*-means algorithm (i.e., the mean squared distance between the nodes and their assigned cluster centroids). Additionally, we also test an alternative data clustering method, Voronoi clustering, which assigns each node to the cluster with the closest centroid in the embedding space, with the cluster centroids being the ones of the true communities. Because the Voronoi clustering method has access to additional information about the locations of true communities, it provides the best-case scenario for the *K*-means algorithm. The results for Voronoi clustering are presented in the Supporting Information 7.

We assess the performance by comparing the similarity between the planted partition of the network and the detected partition of the algorithm. We used the element-centric similarity[63], denoted by S, with an adjustment such that a random shuffling of the community memberships for the two partitions yields $S = 0$ on expectation (See Supporting Information Section 1). This way, for planted divisions into equal-sized communities, $S = 0$ represents the baseline performance of the trivial algorithm, while $S > 0$ indicates that communities are detectable by the given algorithm.

Simulations: PPM

We test the graph embedding and community detection algorithms on networks of $n = 100{,}000$ nodes generated by the PPM, with $q \in \{2, 50\}$ communities of equal size and average degree $\langle k \rangle \in \{5, 10, 50\}$ (Fig. 1). Spectral methods find communities better than random guessing below the detectability limit μ^*, i.e., $S > 0$, for $\mu < \mu^*$ and $\langle k \rangle = 50$ (Fig. 1C, F). However, their performance is much worse when the average degree is small ($\langle k \rangle = 5$, Fig. 1A, D). For example, Laplacian EigenMap falls short below the detectability limit ($\mu < \mu^*$), despite having the optimal detectability limit when the average degree is sufficiently large[64]. All techniques, including BP that is supposed to be optimal for sparse networks, fail the exact recovery of the clusters for sparse networks even if the value of μ is low ($\langle k \rangle = 5$, Fig. 1A, D). We find that misclassifications are inevitable for these highly sparse networks because some nodes end up being connected with other communities more densely than with their own community by random chance. Notably, the poor performance of the BP algorithm is mainly observed in the networks with 50 communities ($q = 50$; Fig. 1F), where the prevalence of many local minima may exacerbate the limitations of the greedy optimization used to optimize the objective of the BP algorithm.

On the other hand, node2vec is substantially better than the spectral methods, and its performance is the closest to that of the BP algorithm for sparse networks (Fig. 1A, D). The results are striking, given that the K-means algorithm can significantly worsen the performance of node2vec. Crucially, the information-theoretical limit of community detectability sharply separates the detectable and undetectable regime of communities for node2vec, demonstrating the validity of our theoretical result. node2vec consistently achieves a good performance across different numbers of communities and different network sparsity. Furthermore, node2vec performs well even if we reduce the embedding dimension C from 64 to 16, which is smaller than the number of communities in the cases where $q = 50$ (Supporting Information Section 5). We also confirmed that the effectiveness of node2vec is robust for different sets of hyperparameter values (Supporting Information Section 6).

Simulations: LFR benchmark

The PPM is a stylized model that lacks key characteristics of empirical community structure. We test the graph embedding using more realistic networks generated by the LFR model[65], which produces networks with heterogeneous degree and community-size distributions, to assess the performance of the methods in a more practical context. Unlike the PPM, however, the theoretical detectability limit of communities in LFR networks is not known. We build the LFR networks by using the following parameter values: number of nodes $n = 10{,}000$, degree exponent $\tau_1 \in \{2.1, 3\}$, average degree $\langle k \rangle \in \{5, 10, 50\}$, maximum degree $\sqrt{10n}$, community-size exponent $\tau_2 = 1$, community-size range $[50, \sqrt{10n}]$.

In LFR networks, the BP algorithm and the non-backtracking embedding—which have an excellent performance on the PPM networks, at least in theory—underperform noticeably (Fig. 2), suggesting that optimal methods for the standard PPM may not perform well in practice. The underperformance is likely due to the violation of the assumption in the BP algorithm that loops are negligible in the network. Even if the network is highly sparse, loops are likely to be formed when the degree distribution is highly heterogeneous[66,67]. As a result, the BP falls short of the LFR networks. node2vec struggled to recover the planted communities perfectly, even when they were well separated, as previously noted[22]. We note that the substandard performance of node2vec on the LFR networks may be attributed to the heterogeneity in community sizes, as the K-means algorithm tends to detect communities of nearly equal sizes[68] (Fig. 2A, B). In fact, when the Voronoi clustering method is used, the performance of node2vec is significantly improved, suggesting that the substandard performance of node2vec is attributed to the clustering algorithm, not to the embedding itself. Laplacian EigenMap outperformed other methods, except in extremely sparse networks (Fig. 2A, B). It is worth noting that Laplacian EigenMap is highly sensitive to the number of dimensions. When the number of dimensions is set to 16, Laplacian EigenMap underperforms considerably. On the other hand, node2vec consistently performs well even across different number of dimensions (Supporting Information Section 7). Even with the smaller embedding dimension $C = 16$, node2vec performs comparably well with the flat SBM (Supporting Information Section 5). We also confirmed that the effectiveness of node2vec is robust for different sets of hyperparameter values (Supporting Information 6).

Empirical networks

We evaluated graph embedding methods using six empirical networks from various domains. Since communities in empirical networks are unknown, we relied on node metadata labels to define community memberships. We note that node attributes do not necessarily align

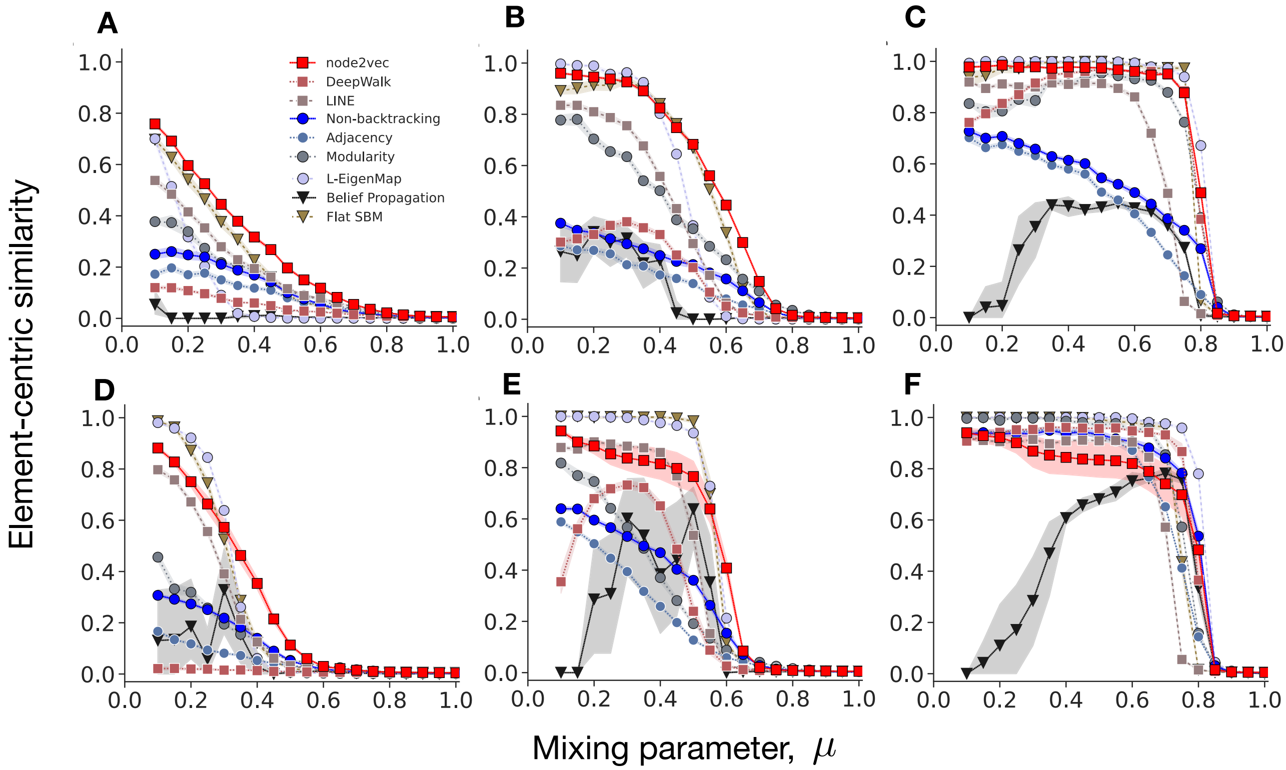

Fig. 2 | Performance of community detection methods on the LFR benchmark networks, as a function of the mixing parameter μ. We generated networks with $n = 10^4$ nodes with different edge sparsity ($\langle k \rangle = 5$ in **A**, **D**, $\langle k \rangle = 10$ in (**B**, **E**), $\langle k \rangle = 50$ in **C**, **F**). The degree exponent $\tau_1 = 2.1$ in **A**–**C**, and $\tau_1 = 3$ in **D**–**F**. node2vec consistently performs well across different sparsity regimes for most μ-values, with a larger margin for sparser networks. The BP algorithm, which is provably optimal for networks generated by the PPM, fails to identify some easily detectable communities, even with the initial parameters set according to the ground-truth communities. The error bands represent the 90% confidence interval by a bootstrapping with 10^4 resample.

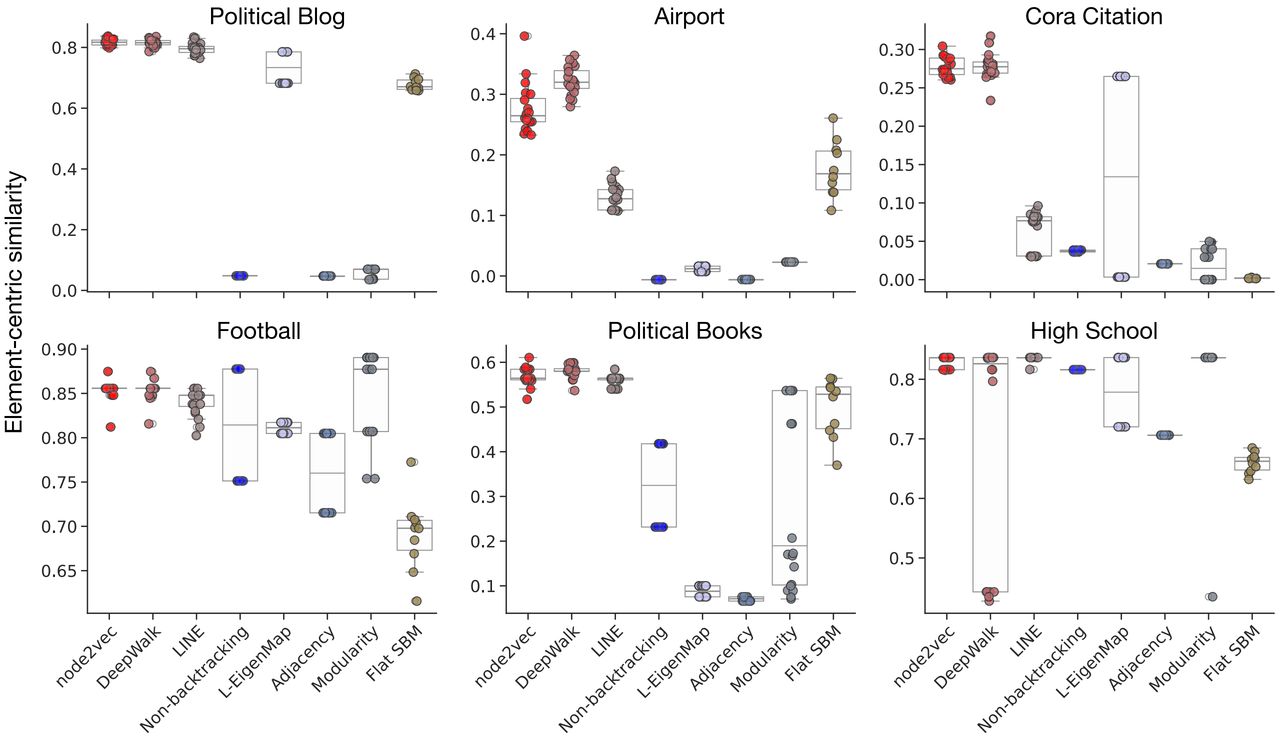

Fig. 3 | Performance of community detection methods on empirical networks. Each panel illustrates the distribution of element-centric similarities for the community detection and graph embedding methods. Each circle denotes the outcome of a single run. The boxes indicate the quartiles of this distribution. The whiskers extend to the farthest data point within 1.5 times the interquartile range from the nearest hinge.

with the detected structural communities, hence community detection methods may fail to identify the node groups based on node attributes[69,70]. Keeping this potential issue in mind, we focus on the following networks, where node attributes align relatively well with the community structures, to shed light on the practical performance of graph embedding methods. `Political blog network` represents hyperlinks between U.S. political blogs related to the 2004 U.S. presidential election[71]. The network consists of 1222 nodes (blogs) and 16,714 edges, where an edge represents a citation from one blog to another on its front page. As the community membership of the blogs, we use the blog categorization into liberal or conservative identified by an automated classification from several weblog directories. `World-wide airport network` consists of 2939 nodes representing airports in the world and 15,677 edges representing direct scheduled flights between the airports[72]. As the community membership of the airports, we use the geographical classification into four regions (Africa, Americas, Asia & Oceania, and Europe). `Cora citation network` consists of 2708 scientific publications and 5429 citations among the publications[73]. As the community membership of the publications, we use the scientific field classification into seven fields of study (computer science, mathematics, physics, statistics, engineering, materials science, and medicine). `Football network` represents American football games between Division IA colleges during regular season Fall 2000. The nodes represent football teams, and the edges represent the matches between the two teams. Each team belongs to one of 12 conferences, and we use the conference classification as the community membership[74]. `Political book network` represents a network of books on US politics published around the time of the 2004 presidential election. Each node represents a book, and two books are connected if they are frequently copurchased by the same buyers[75]. We use the political leaning of the books as the community membership. `High-school network` represents a contact network of students in a high school in Marseilles, France. Each node represents a student and an edge between two students indicates a contact between them during 4 days in Dec. 2011[76]. The community membership of the students is the year of their high school entrance.

We consider a scenario where the number q of communities is not known. We estimate q by using the silouette score[77]. Specifically, we identify the clusters with the K-means algorithm, by imposing a number of clusters q going from 2 to 20, and choose the value of q with the highest silouette score. We use the same parameter set to generate the embeddings and identify the communities. We run the whole process of community detection—from graph embedding, the estimation of the number of communities, and clustering—10 times with different random seeds, and report the agreement between the ground-truth communities and the detected communities in terms of the element-centric similarity for each run (Fig. 3).

node2vec and DeepWalk performed the best in four out of the six networks (`Political Blog`, `Airport`, `Cora Citation`, `Political Books`), and at least on par with the top-performing method in `High School`, suggesting that they performed consistently well across different networks (Fig. 3). Another neural embedding method—LINE—performed similarly with node2vec and DeepWalk except for two networks (`Airport` and `Cora`). The performance of the spectral embedding methods is less consistent across networks. For example, L-EigenMap can perform on par with the top-performing methods in three networks (`Political Blog`, `Cora`, and `High School`) but underperform on the other four networks. Similary, Modularity embedding performed particularly well on `Football` but substantially underperformed on the other networks.

Discussion

We investigated the ability of neural graph embeddings to encode communities by focusing on shallow linear graph neural networks—node2vec, DeepWalk, and LINE—and comparing them with traditional spectral approaches. We proved that, for not-too-sparse networks created by the PPM, node2vec is an optimal method to encode their community structure in that the algorithmic detectability limit coincides with the information-theoretic limit. Our results elucidate how and why node2vec works for community detection by demonstrating the equivalence between the embedding learned by node2vec and the spectral embedding based on the eigenvectors of the normalized Laplacian matrix. This equivalence provided insights into how communities in a network are embedded and the effectiveness of node2vec in learning network communities.

Our theoretical framework shows that graph embeddings based on simple neural networks can achieve optimal community detection. This finding provides guiding principles for developing effective neural embedding methods that are able to resolve communities in embedding space. In neural graph embeddings, deep neural structures and non-linear activation are considered indispensable in order to achieve high performance. The neural network architecture is also critical for graph neural networks for the community detection task[78]. Our findings instead demonstrate that a simple neural network with only one hidden layer and no non-linear activation can achieve the information-theoretical detectability limit of communities.

DeepWalk[38] and LINE[39] are also optimal in terms of the detectability limit of communities (Supplementary Information Section 2). However, node2vec surpasses both DeepWalk and LINE in numerical tests, owing to two key features. First, node2vec learns degree-agnostic embeddings, which are highly robust against degree heterogeneity[62]. By contrast, DeepWalk tends to learn node degree as the primary dimension in the embedding space[62]. Consequently, degree heterogeneity introduces considerable noise to the community structure in the DeepWalk embedding. Second, LINE is a specific instance of node2vec with window size $T = 1$[30], and thus learns the dyadic relationships between nodes. As is the case for node2vec, LINE is resilient to degree heterogeneity, and performed closely to node2vec for some networks in our simulations. However, it did not perform as well as node2vec, and this discrepancy may be attributed to LINE's emphasis on learning stochastic and noisy dyadic relationships, as opposed to the indirect relationships that node2vec captures.

Our results come with caveats. First, our numerical results do not report the limiting performance of the embedding methods, rather the lower bound of the performance limited by the K-means algorithm. With graph embedding methods, the performance of community detection depends on both the quality of the embedding and the performance of the subsequent data clustering procedure. Consequently, the performance of the graph embedding methods can be limited by the K-means algorithm. For instance, a previous study[22] using the K-means algorithm demonstrated that node2vec did not perform as well as standard community detection methods for the LFR networks even if its hyperparameters are fine-tuned. Consistently with this result, the performance of node2vec was suboptimal for the LFR networks in our analysis. However, we note that the LFR networks—that produce communities of different sizes—are challenging for the K-means algorithm—that tends to detect communities of nearly equal sizes[68]. In fact, communities in LFR networks are still well separated in the embedding of node2vec, as knowing the position of the centroids of the planted communities leads to a very good performance (Supplementary Information 7). Nevertheless, the clustering step is a critical limitation when using graph embedding for community detection. An extended K-means algorithm that can handle imbalanced cluster sizes could be a potential solution to this issue[79]. Our results reveal that communities are accurately represented in the embeddings, which might be sufficient for applications that can benefit from community structure but do not require the clustering step, such as link prediction[80] and node classification[20,62].

Second, in our analytical derivations, we assumed that the average degree is sufficiently large, as is the case for the corresponding analysis of spectral modularity maximization[58]. Thus, the optimality may not hold if networks are substantially sparse. However, our simulations suggest that node2vec is resilient to network sparsity compared with traditional spectral embedding methods. Understanding the factor inducing such resilience is left to future work.

Third, while we restricted ourselves to the community detection task, graph embeddings have been used for other tasks, including link prediction, node classification, and anomaly detection. Investigating the theoretical foundation behind the performance of neural embeddings in other tasks is a promising research direction.

We believe that our study will provide the foundation for future studies that uncover the inner workings of neural embedding methods and bridge the study of artificial neural networks to network science.

Methods

node2vec as spectral embedding

node2vec learns the structure of a given network based on random walks. A random walk traverses a given network by following randomly chosen edges and generates the sequence of nodes $x^{(1)}$, $x^{(2)}$, ... The sequence is then fed into skip-gram word2vec[81], which learns how likely it is that a node j appears in the surrounding of another node i up to a certain time lag T (i.e., window length) through the conditional probability

$$P(x^{(t+\tau)} = j | x^{(t)} = i, 1 \leq |\tau| \leq T) = \frac{1}{Z}\exp(\boldsymbol{u}_i^\top \boldsymbol{v}_j), \tag{2}$$

where $\boldsymbol{u}_i \in \mathbb{R}^{C\times 1}$, $\boldsymbol{v}_j \in \mathbb{R}^{C\times 1}$, and Z is a normalization constant. Each node i is associated with two vectors: vector $\boldsymbol{u}_i$ represents the embedding of node i; $\boldsymbol{v}_i$ represents node i as a context of other nodes. Because the normalization constant is computationally expensive, node2vec uses a heuristic training algorithm, i.e., negative sampling[81]. When trained with negative sampling, skip-gram word2vec is equivalent to a spectral embedding that factorizes matrix $\mathbf{R}^{\mathrm{n2v}}$ with elements[30,82]:

$$R_{ij}^{\mathrm{n2v}} = \log \frac{1}{T}\sum_{\tau=1}^{T}\left[\frac{P(x^{(t+\tau)} = j | x^{(t)} = i)}{P(x^{(t)} = j)}\right], \tag{3}$$

in the limit of $C \to n$ with T greater than or equal to the network diameter, where $P(x^{(t)} = i)$ is the probability that the t^{th} node in the given sequence is node i (see Supporting Information Section 3 for the step-by-step derivation). Note that the two embedding vectors $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$ generated by node2vec are parallel to each other because $\mathbf{R}^{\mathrm{n2v}}$ is symmetric[30,62,82].

Leveraging this equivalence, we take another step forward to connect the result with random matrix theory, deriving the detectability limit of these methods for community detection. While previous studies demonstrated that node2vec factorizes $\mathbf{R}^{\mathrm{n2v}}$, it remains unclear about its spectral properties, which is crucial to derive the detectability limit. Deriving the spectrum of $\mathbf{R}^{\mathrm{n2v}}$ in a closed form is challenging. In fact, to identify the spectral density analytically, we need to decompose $\mathbf{R}^{\mathrm{n2v}}$ into a linear combination of matrices (e.g., $\frac{1}{T}\sum_{\tau=1}^{T}\left[\frac{P(x^{(t+\tau)}=j|x^{(t)}=i)}{P(x^{(t)}=j)}\right]$), which is not straightforward due to the non-linear element-wise logarithmic transformation. Here, we derive the spectral properties of $\mathbf{R}^{\mathrm{n2v}}$ by approximating the element-wise logarithm with a linear function based on the assumption that the window length T is sufficiently large. To demonstrate our argument, let us describe R_{ij}^{n2v} in the language of random walks. Given that the network is undirected and unweighted, the probability $P(x^{(t)} = j)$ corresponds to the long-term probability of finding the random walker at node j. The probability $P(x^{(t+\tau)} = j | x^{(t)} = i)$ refers to the transition of a walker from

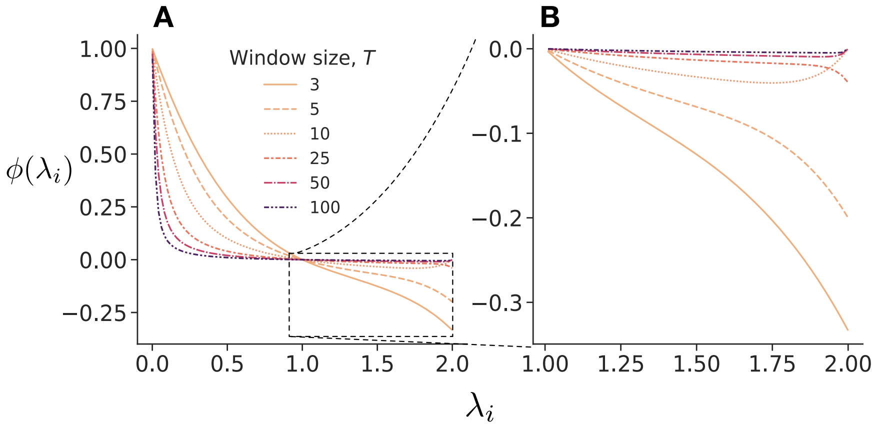

Fig. 4 | Matrix factorization of node2vec. Graph kernel $\phi(\lambda_i; T)$ of node2vec matrix $\hat{\mathbf{R}}^{\text{n2v}}$ across different T values. **A** The plot for all eigenvalues. **B** A zoom-in plot for the values between one and two. The function $\phi(\lambda_i)$ is non-negative and monotonically decreasing for $0 < \lambda_i \leq 1$ and $\phi(\lambda_i) \leq 0$ for $1 < \lambda_i \leq 2$.

node i to node j after τ steps. In the limit $\tau \to \infty$, the walker reaches the stationary state, and $P(x^{(t+\tau)} = j | x^{(t)} = i)$ approaches $P(x^{(t)} = j)$. Thus, in the regime of a sufficiently large T, we take the Taylor expansion of $R_{ij}^{\text{n2v}} = \log\left(1 + \epsilon_{ij}\right)$ around $\epsilon_{ij} = \sum_{\tau=1}^{T} P(x^{(t+\tau)} = j | x^{(t)} = i)/[TP(x^{(t)} = j)] - 1$ and obtain

$$R_{ij}^{\text{n2v}} \simeq \hat{R}_{ij}^{\text{n2v}} \quad : = \tfrac{1}{T} \sum_{\tau=1}^{T} \left[\tfrac{P(x^{(t+\tau)} = j | x^{(t)} = i)}{P(x^{(t)} = j)} \right] - 1. \tag{4}$$

In matrix form,

$$\hat{\mathbf{R}}^{\text{n2v}} = \frac{2m}{T} \left[\sum_{\tau=1}^{T} \left(\mathbf{D}^{-1}\mathbf{A} \right)^{\tau} \right] \mathbf{D}^{-1} - \mathbf{1}_{n \times n}, \tag{5}$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{D}$ is a diagonal matrix whose diagonal element D_{ii} is the degree k_i of node i, m is the number of edges in the network, and $\mathbf{1}_{n \times n}$ is the $n \times n$ all-one matrix. We used $P(x^{(t)} = j) = k_j/2m$ and $(\mathbf{D}^{-1}\mathbf{A})_{ij}^{\tau} = P(x^{(t+\tau)} = j | x^{(t)} = i)$, derived from the fact that $P(x^{(t)} = j)$ is proportional to degree in undirected networks; $\mathbf{D}^{-1}\mathbf{A}$ is the transition matrix, whose τth power represents the random walk transition probability after τ steps.

The node2vec matrix $\hat{\mathbf{R}}^{\text{n2v}}$ has a connection to the normalized Laplacian matrix, $\mathbf{L}$, which is tightly related to the characteristics of random walks and network communities[83]. The normalized Laplacian matrix is defined by $\mathbf{L} : = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$. By using an alternative expression of the transition probability, i.e., $(\mathbf{D}^{-1}\mathbf{A})^{\tau} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})^{\tau}\mathbf{D}^{\frac{1}{2}}$, we rewrite $\hat{\mathbf{R}}^{\text{n2v}}$ as

$$\begin{aligned} \hat{\mathbf{R}}^{\text{n2v}} &= \frac{2m}{T} \left[\sum_{\tau=1}^{T} \mathbf{D}^{-\frac{1}{2}} \left(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \right)^{\tau} \mathbf{D}^{-\frac{1}{2}} \right] - \mathbf{1}_{n \times n} \\ &= 2m\mathbf{D}^{-\frac{1}{2}} \left[\frac{1}{T} \sum_{\tau=1}^{T} (\mathbf{I} - \mathbf{L})^{\tau} - \frac{\mathbf{D}^{\frac{1}{2}}\mathbf{1}_n}{\sqrt{2m}} \frac{\mathbf{1}_n^{\top}\mathbf{D}^{\frac{1}{2}}}{\sqrt{2m}} \right] \mathbf{D}^{-\frac{1}{2}}, \end{aligned} \tag{6}$$

where $\mathbf{1}_n$ is a column vector of length n. We note that vector $\mathbf{D}^{1/2}\mathbf{1}_n/\sqrt{2m}$ is a trivial eigenvector of $\mathbf{L}$ associated with the null eigenvalue, $\lambda_1 = 0$. Furthermore, $(\mathbf{I} - \mathbf{L})^{\tau}$ changes the eigenvalues while keeping the eigenvectors intact. This means that $\hat{\mathbf{R}}^{\text{n2v}}$ can be specified by using the spectrum of $\mathbf{L}$, i.e.,

$$\hat{\mathbf{R}}^{\text{n2v}} = \mathbf{D}^{-\frac{1}{2}}\boldsymbol{\Gamma} \begin{bmatrix} \phi(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & \phi(\lambda_n) \end{bmatrix} \boldsymbol{\Gamma}^{\top}\mathbf{D}^{-\frac{1}{2}}, \tag{7}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times n}$ is the matrix of the eigenvectors of $\mathbf{L}$, and ϕ is a *graph kernel*[18] that transforms the eigenvalues λ_i ($i = 1, 2, \ldots, n$) of $\mathbf{L}$ by

$$\phi(\lambda_i) = \begin{cases} \frac{2m(1-\lambda_i)\left[1-(1-\lambda_i)^T\right]}{T\lambda_i} & (\lambda_i \neq 0), \\ 0 & (\lambda_i = 0), \end{cases} \tag{8}$$

or equivalently $\phi(\lambda_i) = \frac{2m}{T}\sum_{\tau=1}^{T}(1 - \lambda_i)^{\tau}$ if $\lambda_i \neq 0$ (Fig. 4). Equation (7) tells us that the eigenvectors $\mathbf{U}$ of $\hat{\mathbf{R}}^{\text{n2v}}$ are equivalent to the eigenvectors $\mathbf{\Gamma}$ of the normalized Laplacian matrix, up to a linear transformation given by

$$\mathbf{U} : = \mathbf{D}^{\frac{1}{2}}\boldsymbol{\Gamma}. \tag{9}$$

Building on the correspondence between the normalized Laplacian $\mathbf{L}$ and the node2vec matrix $\hat{\mathbf{R}}^{\text{n2v}}$, we derive the algorithmic community detectability limit of node2vec. Following[58,60,64], we assume that the network consists of two communities generated by the PPM. Then, the non-trivial eigenvector of $\mathbf{L}$ encodes the communities and has the optimal detectability limit of communities, provided that the average degree is large (1)[58,60,64]. This non-trivial eigenvector of $\mathbf{L}$ corresponds to the *principal* eigenvector of $\hat{\mathbf{R}}^{\text{n2v}}$. Specifically, the non-trivial eigenvector of $\mathbf{L}$ is associated with the smallest non-zero eigenvalue λ_2, which is $\lambda_2 < 1$ when each community is densely connected within itself and sparsely with other communities[17]. The eigenvalues are mirrored in the eigenvalues $\phi(\lambda_i)$ of $\hat{\mathbf{R}}^{\text{n2v}}$, and λ_2—the smallest non-zero eigenvalue—yields the maximum ϕ-value (Fig. 4).

This correspondence of non-trivial eigenvectors between $\hat{\mathbf{R}}^{\text{n2v}}$ and $\mathbf{L}$ suggests that communities detectable by $\mathbf{L}$ are also detectable by $\hat{\mathbf{R}}^{\text{n2v}}$ and vice versa. Thus, spectral embedding with $\hat{\mathbf{R}}^{\text{n2v}}$ has the same information-theoretic detectability limit as spectral methods relying on eigenvectors of $\mathbf{L}$, for networks with sufficiently high degree.

Detectability limit of DeepWalk

We expand our argument to include DeepWalk[38]. Similar to node2vec, DeepWalk also trains word2vec but with a different objective function. Previous studies have demonstrated that DeepWalk is a matrix factorization method[30,62]. However, it remains unclear about the spectral properties of the matrix to be factorized. Furthermore, deriving the spectral properties of the matrix is challenging due to the elementwise logarithm involved in the matrix to be factorized. More specifically, DeepWalk generates an embedding by factorizing a matrix with entries[62]:

$$R_{ij}^{\mathrm{DW}} := \log\left(\frac{1}{T}\sum_{\tau=1}^{T}\frac{P(x^{(t)}=i, x^{(t+\tau)}=j)}{P(x^{(t)}=i)\cdot\frac{1}{n}}\right), \tag{10}$$

in the limit of $C \to n$ with T being greater than the network diameter. The element-wise logarithm in Eq. (10) makes it challenging to derive the spectral properties of $\mathbf{R}_{ij}^{\mathrm{DW}}$. Here, we employ a linear approximation by assuming that T is sufficiently large. When T is large, the random walker reaches the stationary state, which is independent of where the walker starts from ref. 83. Thus, we have

$$\begin{aligned}\lim_{\tau\to\infty} P(x^{(t)}=i, x^{(t+\tau)}=j) &= P(x^{(t)}=i)P(x^{(t)}=j)\\ &= P(x^{(t)}=i)\cdot\frac{k_j}{n\langle k\rangle}\end{aligned} \tag{11}$$

In particular, if the degree distribution is Poisson and the average degree is sufficiently large,

$$\frac{k_j}{n\langle k\rangle} \simeq \frac{1}{n}, \tag{12}$$

which is true for the PPM. By substituting (12) into (11), we obtain

$$P(x^{(t)}=i, x^{(t+\tau)}=j) \simeq P(x^{(t)}=i)\cdot\frac{1}{n} \quad \text{for } \tau \gg 1. \tag{13}$$

Armed with this result, we demonstrate the detectability limit of DeepWalk as follows. Assuming that the window length T is large, we take the Taylor expansion of (10) around $\epsilon'_{ij} = \sum_{\tau=1}^{T} P(x^{(t)}=i, x^{(t+\tau)}=j)/[T(P(x^{(t)}=i)\cdot 1/n)] - 1$ and obtain

$$R_{ij}^{\mathrm{DW}} \simeq \hat{R}_{ij}^{\mathrm{DW}} := \left(\frac{1}{T}\sum_{\tau=1}^{T}\frac{P(x^{(t)}=i, x^{(t+\tau)}=j)}{P(x^{(t)}=i)\cdot\frac{1}{n}}\right) - 1. \tag{14}$$

In matrix form,

$$\hat{\mathbf{R}}^{\mathrm{DW}} := \frac{n}{T}\left[\sum_{\tau=1}^{T}\left(\mathbf{D}^{-1}\mathbf{A}\right)^{\tau}\right] - \mathbf{1}_{n\times n}. \tag{15}$$

Note that $\hat{\mathbf{R}}^{\mathrm{DW}}$ is similar to the node2vec matrix $\hat{\mathbf{R}}^{\mathrm{n2v}}$ (5). The right/left eigenvectors of $\hat{\mathbf{R}}^{\mathrm{DW}}$ are obtained from those of the normalized Laplacian by simple multiplications by the operators $\mathbf{D}^{1/2}$ and $\mathbf{D}^{-1/2}$, respectively. Therefore, DeepWalk has the information-theoretical detectability limit as well.

Detectability limit of LINE

LINE[39] is a special version of node2vec with the window length being $T=1$. The corresponding matrix factorized by LINE is given by ref. 30:

$$R_{ij}^{\mathrm{LINE}} := \log\left(\frac{A_{ij}}{k_i k_j} + a_0\right) + \log 2m. \tag{16}$$

For LINE, although ref. 30 shows $\log\left(\frac{A_{ij}}{k_i k_j}\right) + \log 2m$, we introduce a small positive value a_0 ($a_0 > 0$) to prevent the matrix elements from being infinite for $A_{ij} = 0$. To obtain the spectrum of $\mathbf{R}^{\mathrm{LINE}}$, we exploit the Taylor expansion $\log(x + a_0) \simeq \frac{x}{a_0} + \log a_0$ around $x = 0$, where $a_0 > 0$. Specifically, assuming that the average degree is sufficiently large, we obtain

$$\hat{R}_{ij}^{\mathrm{LINE}} = \frac{A_{ij}}{a_0 k_i k_j} + \log a_0 + \log 2m, \tag{17}$$

or equivalently in matrix form

$$\begin{aligned}\hat{\mathbf{R}}^{\mathrm{LINE}} &= \frac{1}{a_0}\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1} + a_1\mathbf{1}_{n\times n}\\ &= \frac{1}{a_0}\mathbf{D}^{-1/2}(\mathbf{I}-\mathbf{L})\mathbf{D}^{-1/2} + a_1\mathbf{1}_{n\times n}\\ &= \frac{1}{a_0}\mathbf{D}^{-1/2}\left(\mathbf{I}-\mathbf{L}+2a_0a_1m\frac{\mathbf{D}^{\frac{1}{2}}\mathbf{1}_n}{\sqrt{2m}}\frac{\mathbf{1}_n^{\top}\mathbf{D}^{\frac{1}{2}}}{\sqrt{2m}}\right)\mathbf{D}^{-1/2}.\end{aligned} \tag{18}$$

where $a_1 := \log a_0 + \log(2m)$. Equation (18) is reminiscent of (6) for node2vec. Comparing Eqs. (18) and (6), it immediately follows that they share the same eigenvectors, and thus node2vec and LINE have the same detectability threshold.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The dataset used in this study is available in the Figshare database under accession code 10.6084/m9.figshare.26808775. The data can be obtained at ref. 84.

Code availability

We made available the code and documentations to reproduce all results. See our archived code at ref. 85 for reproducing our results and the up-to-date version at ref. 57 for replications.

References

1. Barabási, A.-L. & Pósfai, M. Network science, 1st edn. (Cambridge University Press, Cambridge, United Kingdom, 2016).
2. Menczer, F., Fortunato, S. & Davis, C. A. A first course in network science, 1st edn. (Cambridge University Press, Cambridge, 2020).
3. Newman, M. Networks, 2nd edn. (Oxford University Press, Oxford, United Kingdom; New York, NY, United States of America, 2018).
4. Mislove, A., Lehmann, S., Ahn, Y.-Y., Onnela, J.-P. & Rosenquist, J. Understanding the demographics of Twitter users. *Proc. Int. AAAI Conf. Web Soc. Media* **5**, 554–557 (2011).
5. Kojaku, S., Hébert-Dufresne, L., Mones, E., Lehmann, S. & Ahn, Y.-Y. The effectiveness of backward contact tracing in networks. *Nat. Phys.* **17**, 652–658 (2021).
6. Barthélemy, M. Spatial networks. *Phys. Rep.* **499**, 1–101 (2011).
7. Colizza, V., Barrat, A., Barthélemy, M. & Vespignani, A. The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc. Natl. Acad. Sci.* **103**, 2015–2020 (2006).
8. Bardoscia, M. et al. The physics of financial networks. *Nat. Rev. Phys.* **3**, 490–507 (2021).
9. Barucca, P. et al. Network valuation in financial systems. *Math. Finance* **30**, 1181–1204 (2020).
10. Newman, M. E. J. The structure of scientific collaboration networks. *Proc. Natl. Acad. Sci.* **98**, 404–409 (2001).
11. Clauset, A., Arbesman, S. & Larremore, D. B. Systematic inequality and hierarchy in faculty hiring networks. *Sci. Adv.* **1**, e1400005 (2015).
12. Bassett, D. S. & Sporns, O. Network neuroscience. *Nat. Neurosci.* **20**, 353–364 (2017).
13. Bassett, D. S. & Bullmore, E. Small-world brain networks. *Neuroscientist* **12**, 512–523 (2006).
14. Kim, J., Park, S.-M. & Cho, K.-H. Discovery of a kernel for controlling biomolecular regulatory networks. *Sci. Rep.* **3**, 2223 (2013).

15. Samaga, R. & Klamt, S. Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Commun. Signal.* **11**, 1–19 (2013).
16. Rozum, J. C., Zañudo, J. G. T., Gan, X., Deritei, D. & Albert, R. Parity and time reversal elucidate both decision-making in empirical models and attractor scaling in critical boolean networks. *Sci. Adv.* **7**, eabf8124 (2021).
17. von Luxburg, U. A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416 (2007).
18. Kunegis, J. & Lommatzsch, A. Learning spectral graph transformations for link prediction. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 561–568 (Association for Computing Machinery, New York, NY, USA, 2009).
19. Nickel, M. & Kiela, D. Poincaré embeddings for learning hierarchical representations. In: Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. **30** (Curran Associates, Inc., 2017).
20. Peng, H., Ke, Q., Budak, C., Romero, D. M. & Ahn, Y.-Y. Neural embeddings of scholarly periodicals reveal complex disciplinary organizations. *Sci. Adv.* **7**, eabb9004 (2021).
21. Barot, A., Bhamidi, S. & Dhara, S. Community detection using low-dimensional network embedding algorithms. *arXiv* https://arxiv.org/abs/2111.05267 (2021).
22. Tandon, A. et al. Community detection in networks using graph embeddings. *Phys. Rev. E* **103**, 022316 (2021).
23. Chen, H. et al. PME: projected metric embedding on heterogeneous networks for link prediction. In: *Proceedings of the 24th ACM SIGKDD International Conference on KDD*, KDD '18, 1177–1186 (Association for Computing Machinery, New York, NY, USA, 2018).
24. Masrour, F., Wilson, T., Yan, H., Tan, P.-N. & Esfahanian, A. Bursting the filter bubble: fairness-aware network link prediction. *Proc. AAAI Conf. Artif. Intell.* **34**, 841–848 (2020).
25. Kwak, H., An, J., Jing, E. & Ahn, Y.-Y. FrameAxis: characterizing microframe bias and intensity with word embedding. *PeerJ Comput. Sci.* **7**, e644 (2021).
26. Murray, D. et al. Unsupervised embedding of trajectories captures the latent structure of scientific migration. *Proc. Natl. Acad. Sci.* **120**, e2305414120 (2023).
27. Sourati, J. & Evans, J. A. Accelerating science with human-aware artificial intelligence. *Nat. Hum. Behav.* **7**, 1682–1696 (2023).
28. Tshitoyan, V. et al. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95–98 (2019).
29. Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* **74**, 036104 (2006).
30. Qiu, J. et al. Network embedding as matrix factorization: unifying DeepWalk, LINE, PTE, and node2vec. In: *Proceedings of the Eleventh ACM International Conference on WSDM*, WSDM '**18**, 459–467 (Association for Computing Machinery, New York, NY, USA, 2018).
31. Pennington, J., Socher, R. & Manning, C. GloVe: global vectors for word representation. In: *Proceedings of the 2014 Conference on EMNLP*, 1532–1543 (Association for Computational Linguistics, Doha, Qatar, 2014).
32. Agarwal, C., Lakkaraju, H. & Zitnik, M. Towards a unified framework for fair and stable graph representation learning. In: de Campos, C. & Maathuis, M. H. (eds.) *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, vol. **161**, *Proceedings of Machine Learning Research*, 2114–2124 (PMLR, 2021).
33. Dehghan-Kooshkghazi, A., Kamiński, B., Kraínski, L., Prałat, P. & Théberge, F. Evaluating node embeddings of complex networks. *J. Complex Netw.* **10**, cnac030 (2022).
34. Grover, A. & Leskovec, J. Node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on KDD*, KDD '16, 855–864 (Association for Computing Machinery, New York, NY, USA, 2016).
35. Hamilton, W., Ying, Z. & Leskovec, J. Inductive representation learning on large graphs. In: Guyon, I. et al. (eds.) *Advances in Neural Information Processing Systems*, vol. **30** (Curran Associates, Inc., 2017).
36. Liu, L. et al. Understanding the onset of hot streaks across artistic, cultural, and scientific careers. *Nat. Commun.* **12**, 5392 (2021).
37. Meng, L. & Masuda, N. Analysis of node2vec random walks on networks. *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **476**, 20200447 (2020).
38. Perozzi, B., Al-Rfou, R. & Skiena, S. DeepWalk: online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on KDD*, KDD '14, 701–710 (Association for Computing Machinery, New York, NY, USA, 2014).
39. Tang, J. et al. LINE: large-scale information network embedding. In: *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, 1067–1077 (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2015).
40. Veličković, P. et al. Graph attention networks. In: *International Conference on Learning Representations* (Poster) (2018).
41. Fortunato, S. Community detection in graphs. *Phys. Rep.* **486**, 75–174 (2010).
42. Fortunato, S. & Hric, D. Community detection in networks: a user guide. *Phys. Rep.* **659**, 1–44 (2016).
43. Fortunato, S. & Newman, M. E. J. 20 years of network community detection. *Nat. Phys.* **18**, 848–850 (2022).
44. Peixoto, T. P. Parsimonious module inference in large networks. *Phys. Rev. Lett.* **110**, 148701 (2013).
45. Karrer, B. & Newman, M. E. J. Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107 (2011).
46. Zhang, Y. & Tang, M. Exact recovery of community structures using DeepWalk and Node2vec. *arXiv* https://arxiv.org/abs/2101.07354 (2022).
47. Chen, P.-Y. & Hero, A. O. Universal phase transition in community detectability under a stochastic block model. *Phys. Rev. E* **91**, 032804 (2015).
48. Chen, P.-Y. & Hero, A. O. Phase transitions in spectral community detection. *IEEE Trans. Signal Process.* **63**, 4339–4347 (2015).
49. Zhang, Y. & Tang, M. Consistency of random-walk based network embedding algorithms. *arXiv* https://arxiv.org/abs/2101.07354 (2021).
50. Abbe, E. & Sandon, C. Community detection in general stochastic block models: fundamental limits and efficient algorithms for recovery. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science,* 670–688 (2015).
51. Newman, M. E. J. *Networks: an introduction* (Oxford University Press, Oxford; New York, 2010).
52. Krzakala, F. et al. Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci.* **110**, 20935–20940 (2013).
53. Benaych-Georges, F., Bordenave, C. & Knowles, A. Spectral radii of sparse random matrices. *Ann. inst. Henri Poincare (B) Probab. Stat.* **56**, 2141 – 2161 (2020).
54. Newman, M. E. J. Spectral community detection in sparse networks. *arXiv* https://arxiv.org/abs/1308.6494 (2013).
55. Zhang, P. & Moore, C. Scalable detection of statistically significant communities and hierarchies, using message passing for modularity. *PNAS* **111**, 18144–18149 (2014).
56. Decelle, A., Krzakala, F., Moore, C. & Zdeborová, L. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E* **84**, 066106 (2011).
57. Code: network clustering via neural embedding. https://github.com/skojaku/community-detection-via-neural-embedding (2024).
58. Nadakuditi, R. R. & Newman, M. E. J. Graph spectra and the detectability of community structure in networks. *Phys. Rev. Lett.* **108**, 188701 (2012).

59. Condon, A. & Karp, R. M. Algorithms for graph partitioning on the planted partition model. In: Hochbaum, D. S., Jansen, K., Rolim, J. D. P. & Sinclair, A. (eds.) *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*, Lecture Notes in Computer Science, 221–232 (Springer, Berlin, Heidelberg, 1999).
60. Radicchi, F. Detectability of communities in heterogeneous networks. *Phys. Rev. E* **88**, 010801 (2013).
61. Belkin, M. & Niyogi, P. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15**, 1373–1396 (2003).
62. Kojaku, S., Yoon, J., Constantino, I. & Ahn, Y.-Y. Residual2Vec: debiasing graph embedding with random graphs. In: *Advances in Neural Information Processing Systems*, vol. **34**, 24150–24163 (Curran Associates, Inc., 2021).
63. Gates, A. J., Wood, I. B., Hetrick, W. P. & Ahn, Y.-Y. Element-centric clustering comparison unifies overlaps and hierarchy. *Sci. Rep.* **9**, 8574 (2019).
64. Radicchi, F. A paradox in community detection. *EPL (Europhys. Lett.)* **106**, 38001 (2014).
65. Lancichinetti, A., Fortunato, S. & Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**, 046110 (2008).
66. Bianconi, G. & Marsili, M. Loops of any size and Hamilton cycles in random scale-free networks. *J. Stat. Mech.: Theory Exp.* **2005**, P06005 (2005).
67. Cantwell, G. T., Kirkley, A. & Radicchi, F. Heterogeneous message passing for heterogeneous networks. *Phys. Rev. E* **108**, 034310 (2023).
68. Wu, J., Xiong, H., Chen, J. & Zhou, W. A generalization of proximity functions for k-means. In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)* 361–370 (2007).
69. Hric, D., Darst, R. K. & Fortunato, S. Community detection in networks: structural communities versus ground truth. *Phys. Rev. E* **90**, 062805 (2014).
70. Peel, L., Larremore, D. B. & Clauset, A. The ground truth about metadata and community detection in networks. *Sci. Adv.* **3**, e1602548 (2017).
71. Adamic, L. A. & Glance, N. The political blogosphere and the 2004 us election: divided they blog. In: *Proceedings of the 3rd international workshop on Link discovery*, 36–43 (2005).
72. Opsahl, T. Why anchorage is not (that) important: binary ties and sample selection. https://toreopsahl.com/2011/08/12/why-anchorage-is-not-that-important-binary-ties-and-sample-selection/ (2011).
73. McCallum, A. K., Nigam, K., Rennie, J. & Seymore, K. Automating the construction of internet portals with machine learning. *Inf. Retr.* **3**, 127–163 (2000).
74. Girvan, M. & Newman, M. E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**, 7821–7826 (2002).
75. Newman, M. E. J. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**, 8577–8582 (2006).
76. Fournet, J. & Barrat, A. Contact patterns among high school students. *PloS One* **9**, e107878 (2014).
77. Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
78. Kawamoto, T., Tsubaki, M. & Obuchi, T. Mean-field theory of graph neural networks in graph partitioning. In: *Adv. Neural Inf. Process. Syst.*, vol. **31** (Curran Associates, Inc., 2018).
79. Liang, J., Bai, L., Dang, C. & Cao, F. The *k*-means-type algorithms versus imbalanced data distributions. *IEEE Trans. Fuzzy Syst.* **20**, 728–745 (2012).
80. Ghasemian, A., Hosseinmardi, H., Galstyan, A. G., Airoldi, E. M. & Clauset, A. Stacking models for nearly optimal link prediction in complex networks. *Proc. Natl. Acad. Sci.* **117**, 23393–23400 (2019).
81. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. In: *Adv. Neural Inf. Process. Syst.*, vol. **26** (Curran Associates, Inc., 2013).
82. Levy, O. & Goldberg, Y. Neural word embedding as implicit matrix factorization. In: *Adv. Neural Inf. Process. Syst.*, vol. **27** (Curran Associates, Inc., 2014).
83. Masuda, N., Porter, M. A. & Lambiotte, R. Random walks and diffusion on networks. *Phys. Rep.* **716–717**, 1–58 (2017).
84. Kojaku, S., Radicchi, F., Ahn, Y.-Y. & Fortunato, S. Dataset for network community detection via neural embeddings (2023).
85. *Archived code: network clustering via neural embedding*. https://doi.org/10.5281/zenodo.13362073 (2023).

Acknowledgements

This project was partially supported by the Army Research Office under contract number W911NF-21-1-0194, by the Air Force Office of Scientific Research under award numbers FA9550-19-1-0391, FA9550-21-1-0446 and FA9550-24-1-0039, by the National Science Foundation under award number 1927418, and by the National Institutes of Health under awards U01 AG072177 and U19 AG074879.

Author contributions

S.K. and F.R. performed the analysis and experiments. S.K., F.R., Y.A., and S.F. conceived the research, discussed, and wrote the manuscript.

Competing interests

The authors have no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-024-52355-w.

Correspondence and requests for materials should be addressed to Santo Fortunato.

Peer review information *Nature Communications* thanks the anonymous reviewers for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2024