7-2022

On measuring network robustness for weighted networks

Jianbing ZHENG

Ming GAO

Ee-peng LIM
Singapore Management University, eplim@smu.edu.sg

David LO
Singapore Management University, davidlo@smu.edu.sg

Cheqing JIN

See next page for additional authors

Citation

ZHENG, Jianbing; GAO, Ming; LIM, Ee-peng; LO, David; JIN, Cheqing; and ZHOU, Aoying. On measuring network robustness for weighted networks. (2022). *Knowledge and Information Systems.* 64, (7), 1967-1996.
Available at: https://ink.library.smu.edu.sg/sis_research/7233

Author

Jianbing ZHENG, Ming GAO, Ee-peng LIM, David LO, Cheqing JIN, and Aoying ZHOU

Knowledge and Information Systems (2022) 64:1967–1996
https://doi.org/10.1007/s10115-022-01670-z

REGULAR PAPER

On measuring network robustness for weighted networks

Jianbing Zheng[1] · Ming Gao[2] · Ee-Peng Lim[3] · David Lo[3] · Cheqing Jin[1] · Aoying Zhou[1]

Received: 9 April 2021 / Revised: 21 February 2022 / Accepted: 26 February 2022 /
Published online: 1 July 2022

Abstract

Network robustness measures how well network structure is strong and healthy when it is under attack, such as vertices joining and leaving. It has been widely used in many applications, such as information diffusion, disease transmission, and network security. However, existing metrics, including node connectivity, edge connectivity, and graph expansion, can be suboptimal for measuring network robustness since they are inefficient to be computed and cannot directly apply to the weighted networks or disconnected networks. In this paper, we define the $\mathcal{R}$-energy as a new robustness measurement for weighted networks based on the method of spectral analysis. $\mathcal{R}$-energy can cope with disconnected networks and is efficient to compute with a time complexity of $O(|V| + |E|)$, where V and E are sets of vertices and edges in the network, respectively. Our experiments illustrate the rationality and efficiency of computing $\mathcal{R}$-energy: (1) Removal of high degree vertices reduces network robustness more than that of random or small degree vertices; (2) it takes as little as 120 s to compute for a network with about 6M vertices and 33M edges. We can further detect events occurring in a dynamic Twitter network with about 130K users and discover interesting weekly tweeting trends by tracking changes to $\mathcal{R}$-energy.

✉ Ming Gao
mgao@dase.ecnu.edu.cn

Ee-Peng Lim
eplim@smu.edu.sg

David Lo
davidlo@smu.edu.sg

Cheqing Jin
cqjin@dase.ecnu.edu.cn

Aoying Zhou
ayzhou@dase.ecnu.edu.cn

[1] Shanghai Engineering Research Center of Big Data Management on the School of Data Science and Engineering, East China Normal University, Shanghai 200062, China

[2] School of Data Science and Engineering, KLATASDS-MOE on the School of Statistics, and Shanghai Key Laboratory of Mental Health and Psychological Crisis Intervention on the School of Psychology and Cognitive Science, East China Normal University, Shanghai 200062, China

[3] School of Information Systems, Singapore Management University, Singapore, Singapore

Keywords Weighted network · Normalized Laplacian matrix · 2-step commute probability · R-energy

1 Introduction

The popularity of web, mobile phones, and other portable devices has propelled the growth of large-scale networks such as Facebook and Twitter, as well as a wide range of online content sharing and crowdsourcing services. These networks are dynamically evolving as users join and leave, and as their traffic of interactions varies. To characterize the strength or health of these large-scale networks, we need some measures to tell us their robustness.

Network robustness measures how well network structure is strong and healthy when it is under attack, such as vertices joining and leaving. The ability to measure the robustness of networks can benefit several useful applications. For example, in a phone call network, dense and frequent calls among users reduce the likelihood of churn. A similar comment can be made for online social networks. For another example, the robustness of IP networks affects service quality and security. Service providers therefore aim to monitor, manage, and optimize their networks to keep their networks robust. Network robustness is also studied in other applications, such as disease transmission [3, 12] and network security [17].

Evaluating the robustness of a weighted network is a natural problem because edges of the network may associate with attached information, such as times of interactions in Telcom network, # retweets between Twitterers, and # adoptions between a user and an item for online shopping web site. As today's networks are usually of very large scale, efficiently measuring robustness of weighted networks is therefore a challenge. The naive way is to extend existing robustness measures to evaluate the robustness of weighted networks. They include *node connectivity* [11], *edge connectivity* [11], and *algebraic connectivity* [13]. Node (or edge) connectivity $\upsilon(G)$ (or $\varepsilon(G)$) of a weighted network G may be defined by the weights of nodes (or edges) that may be removed to break the networks into multiple connected components. Large node and edge connectivity values suggest that a network is robust. Algebraic connectivity $\lambda(G)$ may be defined by the second smallest eigenvalue of the Laplacian matrix (defined in Sect. 3) of the weighted network.

In combinatorics, an expander network is a connected and undirected network in which every small subset of the vertex set has a large boundary. The goodness (or robustness) of an expander network can be measured by *Cheeger constant* [29], *vertex expansion* [5] and *edge expansion* [18]. Let $G = (V, E, W)$ be a connected, undirected, and weighted network. Cheeger constant $h(G)$, vertex expansion $h_v(G)$, and edge expansion $h_e(G)$ may be defined in Eqs. (1), (2) and (3).

$$h(G) = \min_{S \subset V} \frac{|\partial(S)|}{\min\{vol(S), vol(\overline{S})\}}, \tag{1}$$

$$h_v(G) = \min_{S \subset V, 0<|S| \le \frac{|V|}{2}} \frac{|\partial_{out}(S)|}{|S|}, \tag{2}$$

$$h_e(G) = \min_{S \subset V, 0<|S| \le \frac{|V|}{2}} \frac{|\partial(S)|}{|S|}, \tag{3}$$

where the symbols can be found in Table 1. $vol(S)$ ($vol(\overline{S})$) is the total weighted degrees of vertices in S (complement of S), $\partial(S)$ is the edge boundary of S (i.e., the set of edges

Table 1 The description of symbols in Eqs. (1), (2) and (3)

Symbol	Description
$vol(S)$	The total weights of vertices in S
$vol(\overline{S})$	The total weights of vertices in complement of S
$\partial(S)$	The edge boundary of S
$\partial_{out}(S)$	The outer vertex boundary of S
$\|\partial(S)\|$	The total weights of edges in $\partial(S)$
$\|\partial_{out}(S)\|$	The total weights of vertices in $\partial_{out}(S)$

with exactly one endpoint in S), and $\partial_{out}(S)$ is the outer vertex boundary of S (i.e., the set of vertices in $V \backslash S$ with at least one neighbor in S).

The existing measures may have the following shortcomings:

- They are only applicable to connected networks. Even though a highly robust giant component exists in a network with very few connected components, the network is considered not robust at all as all these measures return zero values.
- They quantify robustness using specific (optimal) combinations of vertices (for node connectivity), specific combination of edges (for edge connectivity), and specific eigenvalue (for algebraic connectivity).
- Even for connected network, they are difficult to scale for large networks of millions vertices. For algebraic connectivity, we need to compute the second smallest eigenvalue of the Laplacian matrix. For node connectivity, edge connectivity, Cheeger constant, vertex expansion, and edge expansion, we have to check all cuts of the weighted network. These are all time-consuming measurements.

In this paper, we aims to extend our proposed $\mathcal{R}$-energy in [14] to measure robustness for the weighted networks. Comparing to the original work, the main contributions of this version can be summarized as follows:

- We propose $\mathcal{R}$-energy as an efficient measure for weighted network robustness. The new measure, defined based on the normalized Laplacian matrix, demonstrates several nice properties. It can also handle networks with multiple connected components and can be computed with good time complexity $O(|V| + |E|)$, where V and E are sets of vertices and edges in a weighted network.
- We find that R-energy can be used to monitor the robustness for dynamic networks. In Theorem 3, we have proved that we can incrementally and efficiently compute the R-energy for dynamic networks with vertex or edge modification, such as insert and delete.
- We further apply $\mathcal{R}$-energy to a dynamic Twitter community with about 130K users to detect events and regular trend patterns that affect the weighted network robustness. We empirically show that more events can be detected from the reply network in this extended version. This points to the positive effect of defining R-energy on a weighted network.

The remainder of the paper is organized as follows. We first review related work in Sect. 2. We then introduce some basic notations in Sect. 3, before presenting $\mathcal{R}$-energy and its algorithm in Sect. 4. We illustrate some important properties of $\mathcal{R}$-energy in Sect. 5 and demonstrate some observations and the performance of $\mathcal{R}$-energy on both synthetic and real networks in Sect. 6. Before we conclude this paper in Sect. 7, we illustrate some patterns and events found using $\mathcal{R}$-energy on a dynamic Twitter user community.

2 Related work

2.1 Robustness

The traditional network robustness measures, node connectivity, and edge connectivity were proposed by Dekker and Colbert [11]. Network expansion can also be used to measure network robustness. Different formulations of expander give rise to different measures of expander, e.g., edge expansion [18], vertex expansion [5], and spectral expansion [23]. Larger edge or vertex expansions indicate less bottleneck inside a network.

Jamakovic and Mieghem proposed to use the second smallest eigenvalue of the Laplacian matrix also known as algebraic connectivity to measure network robustness [13, 19]. Malliaros et al. [23] described the relationship between algebraic connectivity and node/edge connectivities. According to Cheeger's inequality, Chung found that the expansion of a network is closely related to the spectral gap between the largest and the second largest eigenvalues of adjacency matrix. Malliaros et al. confirmed the findings of Chung in [23]. This measure is, however, costly to be computed and is sensitive to the network size. Hence, it is not appropriate for comparing networks of different sizes. Albert et al. [1] used diameter to measure robustness of networks, but the measure does not capture network connectivity which should be considered in robustness measures. As mentioned in Sect. 1, they have some drawbacks to measure robustness of weighted networks.

2.2 Graph energy

The energy of a network has always been defined to be some form of deviation of eigenvalues of some network matrix from the mean of eigenvalues. For example, Gutman defined network energy on an adjacency matrix as the absolute deviation of eigenvalues from the mean of eigenvalues which is zero for any adjacency matrix [16]. In [26, 30], *Laplacian energy* has been defined on the *combinatorial Laplacian matrix*. In [7], *normalized Laplacian energy* is defined on the *normalized Laplacian matrix* in a similar manner.

Day and So studied network energy changes with edge or vertex removals [9, 10]. There are some existing works which derive the lower and upper bounds for different energy definitions including Gutman's graph energy [2], Laplacian energy [26, 30, 31], and normalized Laplacian energy [7]. They are not appropriate measures for network robustness as computing them would be time costly.

3 Definition of $\mathcal{R}$-energy

In this section, based on the normalized Laplacian matrix of a weighted network, we address how the eigenvalues of the normalized Laplacian of the weighted network are related to the structure of the network and define the $\mathcal{R}$-energy to measure the robustness of the weighted network.

3.1 Normalized Laplacian

Consider an undirected network $G = (V, E)$ with vertex set V and edge set E (Let $|V| = n$). Let A_G denote the adjacency matrix representing G and be defined as:

$$A_G(i,j) := \begin{cases} 1, & \text{if } (v_i, v_j) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

Definition 1 A weighted network G is triple (V, E, W_G), where V and E are sets of vertices and edges, and each edge $(v_i, v_j) \in E$ associates with weight w_{ij}. As such, W_G is a weight matrix defined as

$$W_G(i,j) = \begin{cases} w_{ij}, & \text{if } (v_i, v_j) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

Formally, we define the neighbor of v_i to be a vertex set as $N(v_i) := \{v_j | (v_i, v_j) \in E\}$, and degree of v_i to be the total weight of vertices in $N(v_i)$, denoted as $d(v_i) := \sum_{v_j \in N(v_i)} w_{ij}$. We then define degree matrix D_G as

$$D_G(i,j) := \begin{cases} d(v_i), & \text{if } i = j \text{ and } v_i \in V; \\ 0, & \text{otherwise.} \end{cases}$$

A network is an unweighted network if all edges have the identical weight.

Based on matrices W_G and D_G, we define the *normalized Laplacian matrix* of a weighted network in Definition 2.

Definition 2 *Normalized Laplacian matrix* N_G of a weighted network G with nonnegative weight matrix W_G is given by

$$N_G := I - D_G^{-1/2} W_G D_G^{-1/2}.$$

We denote $\zeta_1 \leq \zeta_2 \leq \cdots \leq \zeta_n$ as the sequence of eigenvalues of N_G.

The eigenvalues of the normalized Laplacian matrix satisfy three important properties:

Theorem 1 *Let G be a weighted network of n vertices. The eigenvalues of N_G have*

1. $0 = \zeta_1 \leq \zeta_2 \leq \frac{n}{n-1} \leq \zeta_n \leq 2$;
2. $\zeta_2 = \cdots = \zeta_n = \frac{n}{n-1}$ *if and only if G is a clique of equal edge weights;*
3. *G has at least i connected components if and only if $\zeta_j = 0$, for $j = 1, 2, \ldots, i$.*

We prove the theorem in Appendix. Property (1) illustrates that the second smallest and the largest eigenvalues range from 0 to $\frac{n}{n-1}$ and $\frac{n}{n-1}$ to 2, respectively. As a special case, when all except the smallest eigenvalue equal $\frac{n}{n-1}$, the network is a clique as shown in Property (2). Property (3) states that each additional connected component corresponds to a zero eigenvalue. ζ_1 is therefore 0 in any weighted networks.

3.2 Definition of $\mathcal{R}$-energy

According to Theorem 1, for a weighted network G that is sparsely connected and is far from being a clique, its ζ_2 is small, but ζ_n is large. In contrast, a network that is densely connected and similar to a clique will have ζ_2 not much smaller than ζ_n. As such, a robust weighted network should have a small degree of dispersion on eigenvalues.

To evaluate the degree of dispersion of eigenvalues, we define robustness energy as follows:

Definition 3 Let G be a weighted network, the *robustness energy* (short in *$\mathcal{R}$-energy*) of G is defined as:

$$\mathbb{E}_{\mathcal{R}}(G) := \frac{1}{n-1}\sum_{i=2}^{n}(\zeta_i - \overline{\zeta})^2$$

where $\overline{\zeta} = \frac{1}{n-1}\sum_{i=2}^{n}\zeta_i$.

The definition of $\mathcal{R}$-energy does not consider ζ_1 since its value is always zero. A weighted network is more robust if its $\mathcal{R}$-energy is smaller. This is due to the factor that smaller dispersion of $\zeta_2, \zeta_3, \ldots, \zeta_n$ implies that the weighted network is closer to a clique of equal edge weights. Furthermore, $\mathcal{R}$-energy can be applied for measuring the robustness of both connected and disconnected networks. For a disconnected network, we may observe a large $\mathcal{R}$-energy (less robust) since it has multiple zero eigenvalues.

4 Computation of $\mathcal{R}$-energy

To compute the $\mathcal{R}$-energy, the naive approach is to compute all eigenvalues of the normalized Laplacian matrix. As we known, computing all eigenvalues is computationally expensive. Based on the following theorem, we propose a simple and efficient approach to compute $\mathcal{R}$-energy in $O(|V| + |E|)$.

Theorem 2 *Given that a weighted network of n vertices and its eigenvalues of the normalized Laplacian matrix are $\zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_n$, we have*

1. *the mean of eigenvalues $\zeta_2, \zeta_3, \ldots, \zeta_n$, denoted as $\overline{\zeta}$, is $\frac{n}{n-1}$;*
2. *the $\mathcal{R}$-energy can be computed as*

$$\mathbb{E}_{\mathcal{R}}(G) = \frac{1}{n-1}\sum_{(v_i,v_j)\in E}\frac{w_G(i,j)^2}{d(v_i)d(v_j)} - \frac{n}{(n-1)^2}. \tag{4}$$

Proof According to Definition 2, entry $N_G(i,j)$ of N_G is:

$$N_G(i,j) = \begin{cases} 1, & \text{if } i = j \text{ and } d(v_i) \neq 0; \\ -\frac{w_G(i,j)}{\sqrt{d(v_i)d(v_j)}}, & \text{if } A_G(i,j) \neq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Note that each diagonal element of N_G is 1 and $\zeta_1 = 0$, we have,

$$\frac{1}{n-1}\sum_{i=2}^{n}\zeta_i = \frac{1}{n-1}\sum_{i=1}^{n}\zeta_i = \frac{1}{n-1}\cdot tr(N_G) = \frac{n}{n-1}.$$

where $tr(N_G)$ denotes the trace of matrix N_G.

In terms of the value of $\overline{\zeta}$, we now compute the $\mathcal{R}$-energy:

$$\begin{aligned}\mathbb{E}_{\mathcal{R}}(G) &= \frac{1}{n-1}\sum_{i=2}^{n}\left(\zeta_i - \frac{n}{n-1}\right)^2 \\ &= \frac{1}{n-1}\sum_{i=1}^{n}\zeta_i^2 - \frac{n^2}{(n-1)^2},\end{aligned}$$

where we have $\sum_{i=1}^{n} \zeta_i^2 = tr(N(G)^2)$. To further compute, the ith diagonal element of $N(G)^2$ is

$$\sum_{j=1}^{n} N_G(i,j)N_G(j,i) = \sum_{j \neq i}^{n} \frac{w_G(i,j)^2}{d(v_i)d(v_j)} + 1.$$

Thus, we have:

$$\begin{aligned} \mathbb{E}_{\mathcal{R}}(G) &= \frac{1}{n-1} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \frac{w_G(i,j)^2}{d(v_i)d(v_j)} - \frac{n}{(n-1)^2}. \\ &= \frac{1}{n-1} \sum_{(v_i,v_j) \in E} \frac{w_G(i,j)^2}{d(v_i)d(v_j)} - \frac{n}{(n-1)^2}. \end{aligned}$$

□

Theorem 2 indicates that we can avoid to calculate all eigenvalues for computing $\mathcal{R}$-energy. In terms of the theorem, Algorithm 1 depicts the steps to compute the $\mathcal{R}$-energy for a weighted network. The algorithm consists of two main steps. One is to compute the degree of vertices (Lines 1–3). The other one is to aggregate the value of $\frac{w_G(i,j)^2}{deg(v_i)deg(v_j)}$ for each edge at Lines 4–6. Both the time and space complexities of the algorithm are $O(|V| + |E|)$.

Algorithm 1: *calEnergy*(G)

```
   Input: input weighted network: $G = (V, E)$ and $W_G$;
   Output: the $\mathcal{R}$-energy of $G$: $e$;
1  for each vertex $v_i \in V$ do
2      $deg(v_i) = \sum_{v_j \in N(v_i)} w_G(i,j)$;
3  end
4  for each edge $(v_i, v_j) \in E$ do
5      $e \leftarrow e + \frac{w_G(i,j)^2}{deg(v_i)deg(v_j)}$;
6  end
7  $e \leftarrow \frac{e}{n-1} - \frac{n}{(n-1)^2}$;
8  return $e$
```

4.1 2-step commute probability

Given a weighted network, it can be considered as a random walk, where its transition probability matrix $P = (p_{ij})_{1 \leq i,j \leq n}$ can be defined as

$$p_{ij} := \begin{cases} \frac{w_G(i,j)}{d(v_i)}, & \text{if } (v_i, v_j) \in E; \\ 0, & \text{otherwise.} \end{cases}$$

where p_{ij} denotes the probability of reaching v_j from v_i in one step.

Let $p_{ij}^{(t)}$ denote the probability of reaching v_j from v_i in exactly t step. Specially, $p_{ii}^{(2)}$ means the probability of returning v_i from v_i in exactly 2 steps, namely *2-step commute probability*, i.e.,

$$P_{ii}^{(2)} = \sum_{j=1}^{n} p_{ij} \cdot p_{ji}.$$

The probability is very important because it computes the possibility of a random walk returning back to vertex v_i after 2 steps when a walker starts at v_i. For a well-connected vertex v_i, a random walk is unlikely to return back to it if the walker starts at v_i, i.e., small 2-step commute probability.

In fact, $\mathcal{R}$-energy is related to the average 2-step commute probability of all vertices in G. That is,

$$\frac{1}{n}\sum_{i=1}^{n} p_{ii}^{(2)} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq i}^{n} p_{ij}\cdot p_{ji} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\frac{w_G(i,j)^2}{d(v_i)d(v_j)}.$$

We rewrite Eq. (4) into Eq. (6).

$$\mathbb{E}_{\mathcal{R}}(G) = \frac{n}{n-1}\left(\frac{1}{n}\sum_{(v_i,v_j)\in E}\frac{w_G(i,j)^2}{d(v_i)d(v_j)} - \frac{1}{n-1}\right) \tag{6}$$

The factor $\frac{n}{n-1}$ in Eq. (6) can be considered as a reward factor for the weighted network of n vertices. Larger graphs are therefore more robust due to monotonically decreasing $\frac{n}{n-1}$ as n increases. This factor facilitates the comparison of $\mathcal{R}$-energy for networks with different sizes. Note that the 2-step commute probability of each vertex in a clique of equal edge weights with n vertices is $\frac{1}{n-1}$. The right side of Eq. (6) is thus the difference between the average 2-step commute probability and the average 2-step community probability of a clique with the same size. Hence, the $\mathcal{R}$-energy of G combines the reward of network size with the difference between the average 2-step commute probability of G and a clique with the same size.

4.2 $\mathcal{R}$-energy for disconnected network

$\mathcal{R}$-energy can measure the robustness of both connected and disconnected weighted networks. Suppose that network G has N connected components, denoted as $\{C_k\}_{k=1}^{N}$. In Eq. (7), the energy is derived by weighted sum of the average 2-step commute probability of vertices from each connected component.

$$\mathbb{E}_{\mathcal{R}}(G) = \frac{n}{n-1}\left(\sum_{k=1}^{N}\frac{n_k}{n}P_{C_k} - \frac{1}{n-1}\right) \tag{7}$$

where P_{C_k} is the average 2-step commute probability of vertices from connected component C_k in Eq. (8).

$$P_{C_k} = \frac{1}{n_k}\sum_{(v_i,v_j)\in C_k}\frac{w_G(i,j)^2}{d(v_i)d(v_j)}, k = 1,\ldots,N \tag{8}$$

$\mathcal{R}$-energy therefore considers a large disconnected network G to be robust if G contains a robust giant component. This conclusion is reasonable and is consistent with our intuition.

For an unweighted network G, $w_G(i,j) = 1$ if $A_G(i,j) = 1$, otherwise 0. The $\mathcal{R}$-energy can be computed in Eq. (9).

$$\mathbb{E}_{\mathcal{R}}(G) = \frac{1}{n-1}\sum_{(v_i,v_j)\in E}\frac{A_G(i,j)}{d(v_i)d(v_j)} - \frac{n}{(n-1)^2}. \tag{9}$$

The result is consistent with our published work [14].

4.3 Computing $\mathcal{R}$-energy in an incremental manner

Based on following theorem, we compute $\mathcal{R}$-energy in an incremental manner. It means that $\mathcal{R}$-energy can be applied for measuring robustness of dynamic networks.

Theorem 3 *Given a weighted network $G = (V, E, W)$ of n vertices and its $\mathcal{R}$-energy denoted as $\mathbb{E}_{\mathcal{R}}(G)$, we have*

1. *if v_0 is a new coming vertex, which connects vertex v_k with weight w in network G. The $\mathcal{R}$-energy of the new formed network can be computed as*

$$\frac{n-1}{n}\mathbb{E}_{\mathcal{R}}(G) + \frac{2}{n}\Delta(v_k) - \frac{1}{n^2(n-1)}. \tag{10}$$

2. *if a new coming edge e with weight w connects vertices v_{i_0} and v_{j_0} in network G. The $\mathcal{R}$-energy of the new formed network can be computed as*

$$\frac{n-1}{n}\mathbb{E}_{\mathcal{R}}(G) + \frac{2}{n}\big(\nabla(v_{i_0}) + \nabla(v_{j_0})\big) - \frac{1}{n^2(n-1)}. \tag{11}$$

where $\Delta(v_k) = -\sum_{v_j \in N(v_k)} \frac{w_G(j,k)^2 w}{(d(v_k)+w)d(v_k)d(v_j)} + \frac{w}{d(v_k)+w}$ and $\nabla(v_k) = -\sum_{v_j \in N(v_k)} \frac{w_G(k,j)^2 w}{(d(v_k)+w)d(v_j)d(v_k)} + \frac{w^2}{(d(v_{i_0})+w)(d(v_{j_0})+w)}$.

Proof 1. For vertex v_k, its degree changes to $d(v_k) + w$. Furthermore, the difference of 2-step commute probability for vertex v_k is

$$\begin{aligned}
\Delta(v_k) &\doteq \sum_{v_j \in N(v_k)} \frac{w_G(j,k)^2}{(d(v_k)+w)d(v_j)} + \frac{w^2}{(d(v_k)+w)d(v_0)} - \sum_{v_j \in N(v_k)} \frac{w_G(j,k)^2}{d(v_k)d(v_j)} \\
&= -\sum_{v_j \in N(v_k)} \frac{w_G(j,k)^2 w}{(d(v_k)+w)d(v_k)d(v_j)} + \frac{w}{d(v_k)+w}.
\end{aligned}$$

For all neighbors of vertex v_k, the difference of their 2-step commute probabilities is

$$\begin{aligned}
&\sum_{v_j \in N(v_k)} \frac{w_G(k,j)^2}{(d(v_k)+w)d(v_j)} + \frac{w^2}{(d(v_k)+w)d(v_0)} - \sum_{v_j \in N(v_k)} \frac{w_G(k,j)^2}{d(v_k)d(v_j)} \\
&= -\sum_{v_j \in N(v_k)} \frac{w_G(j,k)^2 w}{(d(v_k)+w)d(v_k)d(v_j)} + \frac{w}{d(v_k)+w} = \Delta(v_k).
\end{aligned}$$

According to Eq. (6), $\mathcal{R}$-energy of the new formed network can be computed as

$$\begin{aligned}
\mathbb{E}_{\mathcal{R}}(G+v_0) &= \frac{1}{n}\left(\sum_{(v_i,v_j)\in E} \frac{w_G(i,j)^2}{d(v_i)d(v_j)} + 2\Delta(v_k)\right) - \frac{n+1}{n^2} \\
&= \frac{1}{n}\left((n-1)\mathbb{E}_{\mathcal{R}}(G) + \frac{n}{n-1} + 2\Delta(v_k)\right) - \frac{n+1}{n^2} \\
&= \frac{n-1}{n}\mathbb{E}_{\mathcal{R}}(G) + \frac{2}{n}\Delta(v_k) - \frac{1}{n^2(n-1)}
\end{aligned}$$

2. For new formed edge $e = (v_{i_0}, v_{j_0})$, the difference of 2-step commute probabilities for vertices in $N(v_{i_0})$ is

$$\begin{aligned}\nabla(v_{i_0}) \doteq & \sum_{v_k \in N(v_{i_0})} \frac{w_G(k, i_0)^2}{(d(v_{i_0}) + w)d(v_k)} + \frac{w^2}{(d(v_{i_0}) + w)(d(v_{j_0}) + w)} \\ & - \sum_{v_k \in N(v_{i_0})} \frac{w_G(k, i_0)^2}{d(v_k)d(v_{i_0})} \\ = & - \sum_{v_k \in N(v_{i_0})} \frac{w_G(k, i_0)^2 w}{(d(v_{i_0}) + w)d(v_k)d(v_{i_0})} + \frac{w^2}{(d(v_{i_0}) + w)(d(v_{j_0}) + w)}.\end{aligned}$$

For vertex v_{i_0}, the difference of 2-step commute probability is

$$- \sum_{v_k \in N(v_{i_0})} \frac{w_G(i_0, k)^2 w}{(d(v_{i_0}) + w)d(v_k)d(v_{i_0})} + \frac{w^2}{(d(v_{i_0}) + w)(d(v_{j_0}) + w)} = \nabla(v_{i_0}).$$

Similarly, we can compute the difference of 2-step commute probability related to vertex v_{j_0}. According to Eq. (6), $\mathcal{R}$-energy of the new formed network can be computed as

$$\begin{aligned}\mathbb{E}_{\mathcal{R}}(G + e) &= \frac{1}{n-1}\left(\sum_{(v_i, v_j) \in E} \frac{w_G(i, j)^2}{d(v_i)d(v_j)} + 2\nabla(v_{i_0}) + 2\nabla(v_{j_0})\right) - \frac{n}{(n-1)^2} \\ &= \frac{1}{n}\left((n-1)\mathbb{E}_{\mathcal{R}}(G) + \frac{n}{n-1} + 2\nabla(v_{i_0}) + 2\nabla(v_{j_0})\right) - \frac{n+1}{n^2} \\ &= \frac{n-1}{n}\mathbb{E}_{\mathcal{R}}(G) + \frac{2}{n}\big(\nabla(v_{i_0}) + \nabla(v_{j_0})\big) - \frac{1}{n^2(n-1)}\end{aligned}$$

□

In terms of Theorem 3, $\mathcal{R}$-energy can be efficiently updated when vertices are added or edges are modified. As a result, we can compute R-energy for dynamic networks in an incremental and efficient manner.

4.4 Some important properties of $\mathcal{R}$-energy

In this section, we show some properties of $\mathcal{R}$-energy of a weighted network.

4.4.1 $\mathcal{R}$-energy for complete networks

Theorem 4 *Given a weighted network $G = (V, E, W)$ of size n,*

1. *If G is a clique of equal edge weight, then $\mathbb{E}_{\mathcal{R}}(G) = 0$;*
2. *If G is a biclique of equal edge weight, then $\mathbb{E}_{\mathcal{R}}(G) = \frac{n-2}{(n-1)^2}$;*

Proof As G is a clique of equal edge weight w, degree of each vertex is therefore $(n-1)w$.

$$\mathbb{E}_{\mathcal{R}}(G) = \frac{1}{n-1}\sum_{i=1}^{n}\sum_{j \neq i}^{n} \frac{w^2}{(n-1)^2 w^2} - \frac{n}{(n-1)^2} = 0.$$

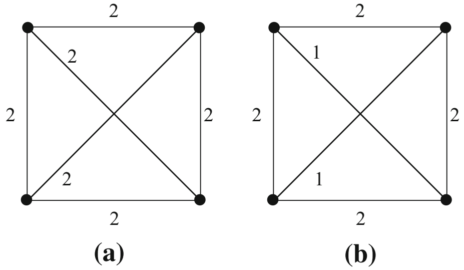


Fig. 1 Two cliques with different weights

If G is a biclique of equal edge weight, let $|V_1| = p$, $|V_2| = q$ and weight be w. Degree of each vertex $v_1 \in V_1$ is qw, and degree of each vertex $v_2 \in V_2$ is pw. Then, $\mathbb{E}_{\mathcal{R}}(G)$ can be computed as:

$$\begin{aligned}&\frac{1}{n-1}\left(\sum_{i=1}^{p}\sum_{j\neq i}^{n}\frac{w^2}{pqw^2}+\sum_{i=p+1}^{p+q}\sum_{j\neq i}^{n}\frac{w^2}{pqw^2}\right)-\frac{n}{(n-1)^2}\\&=\frac{1}{n-1}\left(\sum_{i=1}^{p}\frac{1}{p}+\sum_{i=p+1}^{p+q}\frac{1}{q}\right)-\frac{n}{(n-1)^2}\\&=\frac{2}{n-1}-\frac{n}{(n-1)^2}=\frac{n-2}{(n-1)^2}.\end{aligned}$$

□

This theorem indicates that a clique with equal edge weight is equivalent to an unweighted clique. For example, Fig. 1 shows two cliques of 4 vertices with different weights. In terms of Theorem 4 and Eq. (6), $\mathcal{R}$-energies of two cliques in Fig. 1 are $\mathbb{E}_{\mathcal{R}}(G_a) = 0$ and $\mathbb{E}_{\mathcal{R}}(G_b) = \frac{8}{225}$. Network G_a is therefore more robust than G_b. In addition, only clique with equal weight can achieve zero $\mathcal{R}$-energy. For a biclique, it can be very robust if it is large in size.

4.4.2 Bounds of $\mathcal{R}$-energy

Theorem 5 *Let LB and UB be $\frac{n[w_{min}(n-1)-d_{max}]}{d_{max}(n-1)^2}$ and $\frac{n[w_{max}(n-1)-d_{min}]}{d_{min}(n-1)^2}$, respectively. If G is a connected and weighted network of n vertices, then*

$$\max\{0, LB\} \le \mathbb{E}_{\mathcal{R}}(G) < \min\{1, UB\}, \tag{12}$$

where w_{min} and w_{max} are the minimum and maximum weights of network G, and d_{min} and d_{max} are the minimum and maximum vertex degrees of network G.

Proof At first, we bound the value of $var(G) := \sum_{i=1}^{n}\sum_{j\neq i}^{n}\frac{w_G^2(i,j)}{d(v_i)d(v_j)}$.

$$\begin{aligned}var(G)&=\sum_{i=1}^{n}\frac{1}{d(v_i)}\sum_{v_j\in N(v_i)}\frac{w_G^2(i,j)}{d(v_j)}\\&\le\sum_{i=1}^{n}\frac{w_{max}}{d(v_i)}\sum_{v_j\in N(v_i)}\frac{w_G(i,j)}{d_{min}}=\frac{nw_{max}}{d_{min}}\end{aligned}$$

Similarly, we have

$$var(G) \geq \sum_{i=1}^{n} \frac{w_{min}}{d(v_i)} \sum_{v_j \in N(v_i)} \frac{w_G(i,j)}{d_{max}} = \frac{n w_{min}}{d_{max}}$$

Thus, we have

$$\frac{n w_{min}}{d_{max}} \leq var(G) \leq \frac{n w_{max}}{d_{min}}$$

According to Theorem 2, we have

$$\mathbb{E}_{\mathcal{R}}(G) = \frac{var(G)}{n-1} - \frac{n}{(n-1)^2}$$

It is easy to get the bound of $\mathbb{E}_{\mathcal{R}}(G)$ as

$$LB \leq \mathbb{E}_{\mathcal{R}}(G) \leq UB$$

Note that $\mathbb{E}_{\mathcal{R}}(G) \geq 0$ because it is defined as variance of ζ_i for $i = 2, 3, \ldots, n$. Therefore, we have the left side of Eq. (12). Furthermore, because $0 \leq \frac{w_G(i,j)}{d(v_i)} \leq 1$ and $0 \leq \frac{w_G(i,j)}{d(v_j)} \leq 1$, we have

$$\begin{aligned} var(G) &= \sum_{i=1}^{n} \sum_{v_j \in N(v_i)} \frac{w_G(i,j)}{d(v_i)} \frac{w_G(i,j)}{d(v_j)} \\ &\leq \frac{1}{2} \sum_{i=1}^{n} \sum_{v_j \in N(v_i)} \left(\frac{w_G(i,j)}{d(v_i)} + \frac{w_G(i,j)}{d(v_j)} \right) = n \end{aligned}$$

Then, we have $\mathbb{E}_{\mathcal{R}}(G) < 1$. Thus, we have the right side of Eq. (12). □

This theorem indicates that $\mathcal{R}$-energy ranges from 0 to 1. The left equality holds if weighted network G is a clique with equal weight.

4.4.3 Other topological measures

In this section, we analyze the other possible robustness measures on weighted networks.

The weighted algebraic connectivity, which is defined by the second smallest eigenvalue of Laplacian matrix of a weighted network, is applied to evaluate robustness of weighted airport transportation network [28] and is a measurement of the robustness for weighted networks [20].

The entropy of a weighted network is defined in Eq. (13):

$$entropy = -\sum_{v \in V} \frac{d(v)}{2m} \log \left(\frac{d(v)}{2m} \right), \tag{13}$$

where $d(v)$ denotes the weighted degree of vertex v, and m presents the total weighted degree of all vertices of the network. The entropy of a weighted network evaluates how biased weighted degrees of vertices of the network are. The entropy of a network is maximized, which is $log(n-1)$, if the network is a d-regular network with equal weights whatever the positive integer d is.

The disparity of vertex v_i is defined as below [4]:

$$\eta(v_i) = \sum_{v_j \in N(v_i)} \left(\frac{w_G(i, j)}{d(v_i)} \right)^2 \tag{14}$$

This measure distinguishes how biased weights of out-link edges of a vertices are. For a vertex of k neighbors, when all weights are of the same order, the quantity is closed to $\frac{1}{k}$ ($\ll 1$). In contrast, where only a small number of connections dominate, the quantity is of order $\frac{1}{n}$ ($n \ll k$). Based on disparities of all vertices in a weighted network, we define mean and variance disparities for the network as follows:

$$\textit{m-disparity} = \frac{1}{n} \sum_{v \in V} \eta(v);$$
$$\textit{v-disparity} = \frac{1}{n} \sum_{v \in V} (\eta(v) - \textit{m-disparity})^2.$$

Existing measures may not be suitable to evaluate robustness of weighted networks. We summarize our analysis in Table 2. In detail, a reasonable robustness measure can evaluate how well following networks have:

- Networks with isolated vertices: many vertices in a scale-free network have few neighbors. They are easy to be isolated vertices when the network is attacked. From Table 2, entropy, weighted algebraic connectivity, mean disparity, and variance disparity cannot evaluate robustness of network with isolated vertices because: (1) entropy, mean disparity, and variance disparity have no definition for a vertex of zero degree; (2) weighted algebraic connectivity is always zero.
- Disconnected networks: networks always have many strongly connected components. Weighted algebraic connectivity is zero if the network has multiple strongly connected components. Even though the giant component is a representation of the network, some networks may not have giant component. For the case, weighted algebraic connectivity is invalid which is shown in Table 2.
- d-regular networks: a d-regular network is closed to a clique when d is a large value. On contrast, the network is far away being a clique. Entropy and variance disparity of all d-regular networks are $log(n-1)$ and zero, respectively (note that all edges have the same weight in each d-regular network). Even though regular networks with different d values have different topological structures, entropy and variance disparity cannot distinguish them.
- Weighted networks: edges of a network may associate with weight or attached information, such as times of interactions and # retweets between two users. However, binary version of $\mathcal{R}$-energy ignores weights of edges in a network.
- Large-scale networks: today's networks are usually of very large scale. For example, Cit-Patents [22] has about millions vertices and ten millions edges. Weighted algebraic connectivity cannot be computed efficiently since the second smallest eigenvalue need to be computed. Therefore, efficient measurements of network robustness are required.

5 Robustness on static networks

In this section, we evaluate our proposed $\mathcal{R}$-energy on static networks including synthetically created networks and some real-world networks. We design a set of experiments to compare:

Table 2 Property summary for possible robustness measures for weighted networks

Network	$\mathcal{R}$-energy		Disparity		Entropy	Connectivity		
	Weighted	Binary	Mean	Variance		Algebraic	Node	Edge
Isolated vertex	✓	✓	–	–	–	–	–	–
Disconnected without isolated vertex	✓	✓	✓	✓	✓	–	–	–
Large scale	✓	✓	✓	✓	✓	–	–	–
d-regular	✓	✓	✓	–	–	✓	✓	✓
Weight	✓	–	✓	✓	✓	✓	✓	✓

(1) the effectiveness and scalability of $\mathcal{R}$-energy; (2) common patterns which are found based on $\mathcal{R}$-energy. The experiments were implemented in Java. They were all conducted on a dual-core 64-bit processor with 3.06 GHz CPUs and 128 GB of RAM.

5.1 Networks

Synthetic networks Syn_N is a synthetic graph with N vertices. The generator outputs a synthetic graph as shown in Algorithm 2. The algorithm initializes a graph with N vertices and empty adjacency list. According to the property of the scale-free network, it assigns a degree k to each vertex v in this graph such that $Pr[deg(v) = k] \approx k^{-\alpha}$ from Lines 5 to 8. Note that the value of *totalDeg* should be even. The steps from Lines 9 to 11 guarantee this condition. Next, we assign the neighbors of each vertex after sorting vertices by degree in decreasing order from Lines 13 to 25. Finally, we assign a weight to an undirected edge randomly from Lines 26 to 30.

Algorithm 2: a synthetic undirected and weighted graph

```
   Input: $N, \alpha, weight$;
   Output: $G$: a undirected and weighted graph;
1  $vertexSet \leftarrow$ a vertex set with $N$ vertices; // initialized vertex set;
2  $edgeSet \leftarrow \emptyset$;
3  $weightSet \leftarrow \emptyset$;
4  $totalDeg \leftarrow 0$; // initialized sum of degrees;
5  for each vertex $x \in vertexSet$ //Step 1: Vertex generation do
6      $deg(x) \leftarrow$ sample an integer value from the power law distribution with parameter $\alpha$;
7      $totalDeg \leftarrow totalDeg + deg(x)$;
8  end
9  if numEdge is odd then
10     $deg(x_1) \leftarrow deg(x_1) + 1$;
11     $totalDeg \leftarrow totalDeg + 1$;
12 end
13 sort vertices in $vertexSet$ by descending order of $deg(\cdot)$;
14 $V \leftarrow vertexSet$;
15 for each vertex $x \in vertexSet$ //Step 2: Edge generation do
16     randomly select a vertex set $Y$ from $V$ s.t. $|Y| = deg(x)$ and $x \notin Y$;
17     for each vertex $y \in Y$ do
18         $edgeSet \leftarrow edgeSet \cup \{(x, y)\}$;
19         $deg(y) \leftarrow deg(y) - 1$;
20         if $deg(y) == 0$ then
21             $V \leftarrow V - \{y\}$;
22         end
23     end
24     $V \leftarrow V - \{x\}$;
25 end
26 for each undirected edge $(u, v) \in edgeSet$ //Step 3: Weight assignment do
27     randomly assign a weight $w$ s.t. $w \sim U[0, weight]$;
28     $weightSet \leftarrow ((u, v), w)$;
29     $weightSet \leftarrow ((v, u), w)$;
30 end
31 return $G = (vertexSet, edgeSet, weightSet)$
```

Table 3 Descriptive statistics of experimental networks

Network	Vertices	Edges	Avg. degree
CN	297	2148	7.23
IT	187	11, 907	63.7
HT	7610	15, 751	2.07
AP	16,046	121, 251	7.56
CM	16,264	47, 594	2.93
SP	1,632,804	30,633,564	18.76
WT	1,394,385	5,021,410	3.60
CP	5,969,810	33,037,895	5.53

Real networks We use six static real networks with different sizes from Mark Newman's collection[1], Kristian Skrede Gleditsch's collection[2], and Stanford Large Network Dataset Collection.[3]

- IT: it provides estimates of trade flows between independent states (1948–2000) [15].
- CN: Neural network [27].
- HT: it is a weighted network of coauthorships between scientists posting preprints on the High-Energy Theory E-Print Archive between January 1, 1995, and December 31, 1999 [25].
- AP: it is a weighted network of coauthorships between scientists posting preprints on the Astrophysics E-Print Archive between January 1, 1995, and December 31, 1999 [25].
- CM: it is a weighted network of coauthorships between scientists posting preprints on the Condensed Matter E-Print Archive between January 1, 1995, and March 31, 2005 [25].
- SP: it is an undirected social network on Pokec [22].
- WT: it is an undirected communication network on Wikipedia [27].
- CP: it is a US patent dataset which spans 37 years (January 1, 1963, to December 30, 1999) and includes all the utility patents granted during that period [22].

The descriptive statistics of these networks are demonstrated in Table 3. In this work, we consider these networks weighted and undirected.

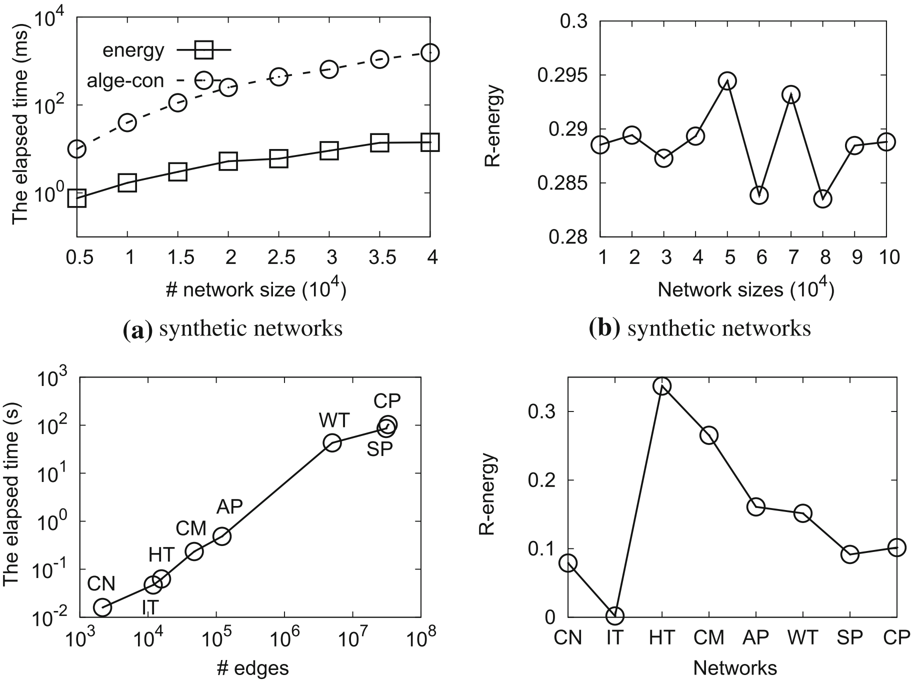
5.2 Efficiency and scalability of computing $\mathcal{R}$-energy

In this task, the synthetic networks are generated with different sizes. We compute the values of $\mathcal{R}$-energy for both synthetic and real networks. We illustrate the elapsed time and the values of $\mathcal{R}$-energy in Fig. 2. Note that computing R-energy and algebraic connectivity is faster than computing node connectivity and edge connectivity since computing later metrics needs to check all cuts of a network, which is very expensive operation. In addition, the later metrics are only proposed for evaluating robustness of unweighted networks. As a result, we only compare the efficiency for computing R-energy and algebraic connectivity in part. In Fig. 2a, we demonstrate the elapsed time for computing R-energy and algebraic connectivity. As illustrated in Fig. 2a, c, we observe that the elapsed time for $\mathcal{R}$-energy linearly scales with

[1] http://www-personal.umich.edu/~mejn/netdata/.

[2] http://ksgleditsch.com/exptradegdp.html.

[3] http://snap.stanford.edu/data/index.html.



(a) synthetic networks

(b) synthetic networks

(c) real networks

(d) real networks

Fig. 2 Performance of computing the $\mathcal{R}$-energy

the number of edges. Furthermore, the elapsed time of computing $\mathcal{R}$-energy is less than 400 ms for synthetic networks with more than 100,000 vertices, and less than 120 s for the largest network Cit-Patents [22] with 5.9M vertices and 33.0M edges. We hold that it is because the complexity of computing $\mathcal{R}$-energy is only $O(|V| + |E|)$. However, the elapsed time of computing algebraic connectivity is more than 1000 times than that of computing R-energy when there are 4×10^4 vertices in a synthetic network. This points to the advantage of using $\mathcal{R}$-energy to measure robustness for large networks.

5.3 Impact of vertex removal to $\mathcal{R}$-energy

Unweighted networks with heavy tail are known to be highly robust against random removal of vertices [8], but are hypersensitive to removal of high-degree vertices [1, 6]. We would like to check whether the same conclusion can be observed from weighted networks.

In this task, we experiment with three vertex removal options, namely (a) remove in decreasing degree order; (b) remove in increasing degree order; and (c) remove in random order. For each option, after removing x fraction of vertices from the networks, we compute $\mathcal{R}$-energy to measure the new network robustness. Figure 3 illustrates the $\mathcal{R}$-energy of resultant network for the three options compared with the $\mathcal{R}$-energy of the original network. From the figure, we obtain three important observations as follows.

- *Networks become less robust sooner when vertices with the highest degrees are removed.* As demonstrated in Fig. 3, compared with the original networks, the values of $\mathcal{R}$-energy increase sooner when vertices with the highest degrees are removed than when vertices with small degrees are removed, or randomly removed. This is due to the fact that vertices with high degrees tend to have smaller 2-step commute probabilities. Removing them leads to an increase in average returning probability. Therefore, the network becomes less robust.

– *Networks remain robust or become slightly more robust when vertices with the smallest degrees are removed*. As illustrated in Fig. 3, we find that the values of $\mathcal{R}$-energy remain constant or decrease slightly when vertices with the smallest degrees are removed from the networks. Because vertices with the smallest degrees have larger 2-step commute probabilities, removing the vertices with the smallest degrees results in little decrease in the average of the returning probabilities.
– *Networks become less robust when vertices are randomly removed. However, the change is slower than that of removing vertices of the highest degrees*. This observation can be attributed to the fact that each vertex has a certain chance to decrease its degree when we remove vertices at random. It indicates that the 2-step commute probability of each vertex increases with certain probability. However, vertices with smallest degrees are more likely to be removed in scale-free networks. Hence, vertices of large 2-step commute probabilities are more likely to be removed leading to a decrease in network energy.

The above three observations are also consistent with the results of the unweighted networks [14] and point out the rationality of proposed $\mathcal{R}$-energy.

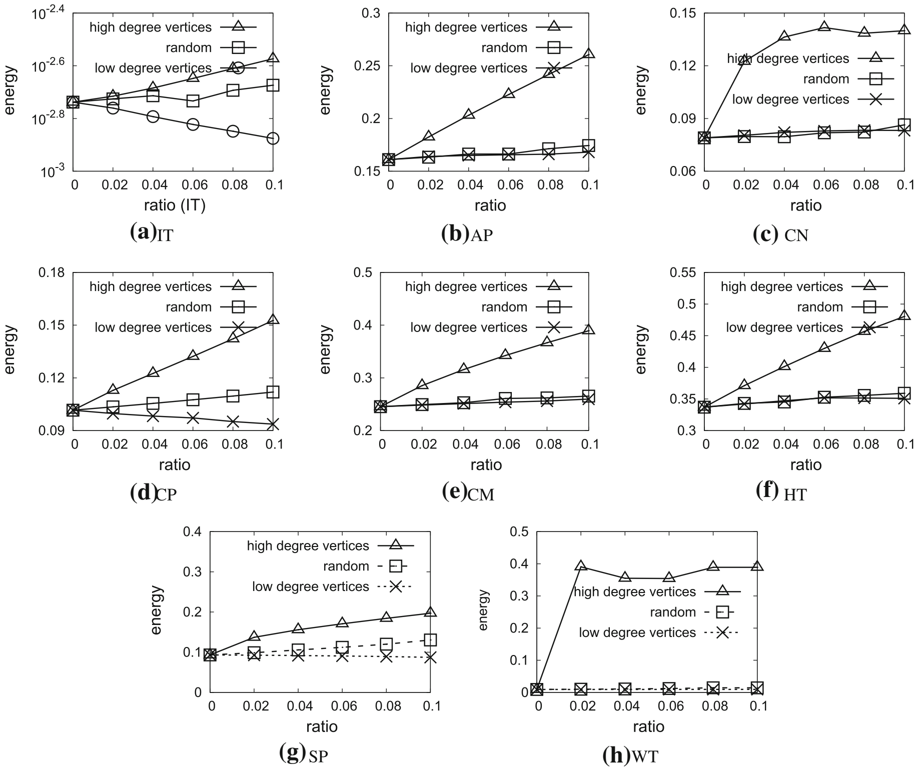
6 Robustness on dynamic networks

Weighted networks evolve with time and so are their robustness. In this section, we apply $\mathcal{R}$-energy on dynamic and time-evolving Twitter weighted network so as to evaluate robustness as a possible measure to detect events and trends. Unlike the previous event and trend detection research which considers time series of messages or news articles generated in social media, our approach utilizes dynamic changes to network structure. These are the changes that cause a network to become suddenly more robust or less robust than usual.

6.1 Data collection

Twitter is a popular microblogging site with users generating and sharing short message contents in real time [21]. In this experiment, we first selected a set of Twitter users U^{us} (U^{sg}) who are the followers and followees of a small set of seed user accounts that belong to US (Singapore) politicians and analysts. These are the users who are more likely to tweet about political topics. We crawled the Twitter data generated by $U^{\cdot}$ from May 1, 2012, to July 29, 2012.

From $U^{\cdot}$, we further selected users who write, reply, or retweet at least a tweet per month over three months. There are 129,056 and 48,339 such users from the USA and Singapore, and we keep them in the user set $U^{\cdot}$ discarding the remaining users and their tweets. Each day, a subset of users in $U^{\cdot}$ may reply or retweet one another. We therefore construct a weighted *reply network* and another weighted *retweet network* for day t and denote them by $G^{\cdot}_{RE}(t)$ and $G^{\cdot}_{RT}(t)$, respectively. An undirected edge (u, v) is included in the reply network for day t if user u replies at least a tweet from user v, or user v replies at least a tweet from user u in



(a) IT **(b)** AP **(c)** CN

(d) CP **(e)** CM **(f)** HT

(g) SP **(h)** WT

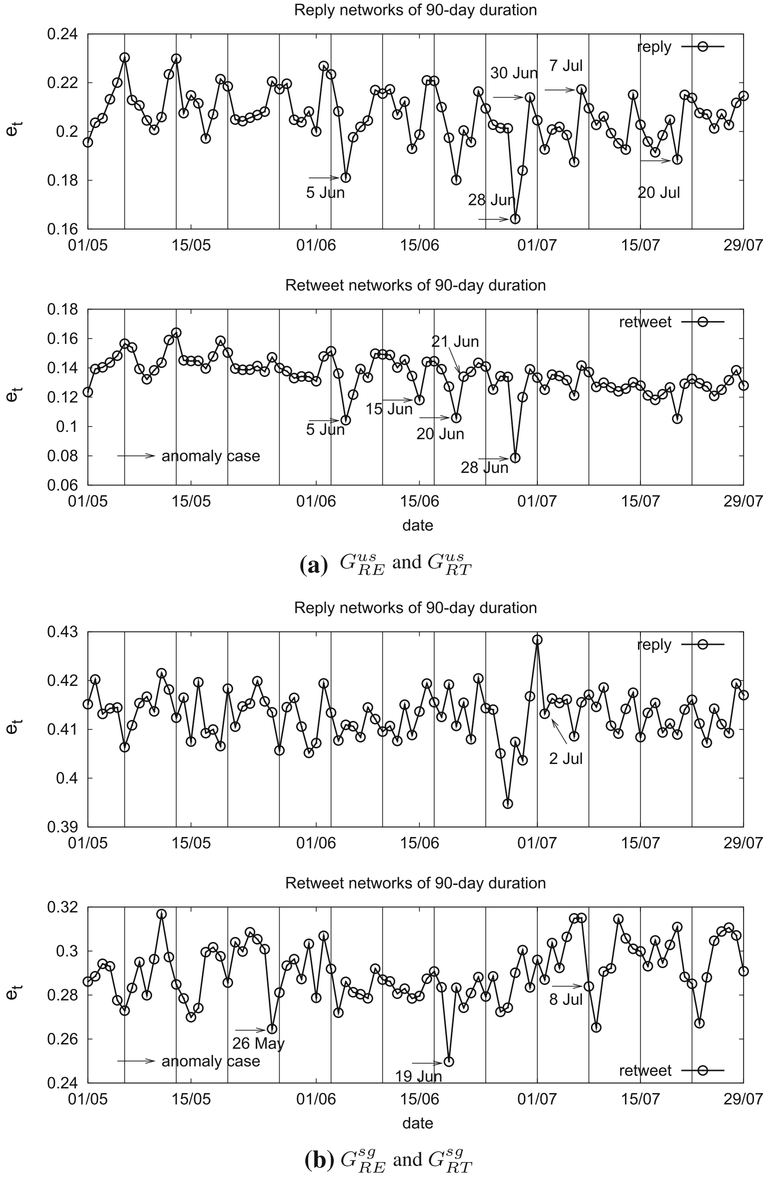
Fig. 3 $\mathcal{R}$-energies of static networks

day t. The weight of undirected edge (u, v) or (v, u) is the number of reply tweets between users u and v. The edges in retweet network on day t are created in a similar manner.

6.2 Event detection

We demonstrate the $\mathcal{R}$-energies of $\dot{G}_{RE}(t)$ and $\dot{G}_{RT}(t)$ in Fig. 4a, b. To facilitate reading, we add vertical lines representing Sundays to the figures. From the figures, we aim to determine events that are characterized by bursts and drops of communication (replies or retweets) by many users. We call these the *internal* and *external events* as the former can be explained by the bursty content but not the latter. For example, a sport event may draw user attention away from tweeting about politics. In addition to event detection, we also want to explain internal events by searching the web.

Suppose $(e_1, e_2 \ldots, e_{90})$ is the sequence of $\mathcal{R}$-energy values, where e_i is the value of $\mathcal{R}$-energy for the i-th day. We calculate the absolute first-order difference of energy sequence, denoted as $(d_1, d_2 \ldots, d_{90})$, where $d_1 = 0$ and $d_{t+1} = |e_{t+1} - e_t|$ for $1 \leq t \leq 89$. Based on the mean and standard deviation of $\{d_t\}$, we can detect an event at time t' statistically if $|d_{t'} - mean(\{d_t\})| > \gamma \cdot stddev(\{d_t\})$ where $mean(\{d_t\})$ and $stddev(\{d_t\})$ denote the mean and standard deviation of $\{d_t\}$, respectively. In other words, an event is found when the absolute first-order difference deviates from mean more than γ times the standard deviation. However, mean is known to be sensitive to anomalies. We therefore employ trimmed mean



(a) G_{RE}^{us} and G_{RT}^{us}

(b) G_{RE}^{sg} and G_{RT}^{sg}

Fig. 4 $\mathcal{G}$-energies of dynamic networks

that is defined as the mean after discarding the smallest and largest $\tau\% \cdot |\{d_t\}|$ values. In this work, we set $\gamma = 3$ and $\tau = 5$ empirically.

To describe an event at day t, we need to extract relevant event description keywords from tweets (which can be replies or retweets) generated on the same day t. We denote the words extracted from reply tweets (or retweets) on day t by $W^{RE}(t)$ (or $W^{RT}(t)$) and the frequency of word $w \in W^{RE}(t)$ (or $W^{RT}(t)$) by $f^{RE}(w, t)$ (or $f^{RT}(w, t)$). We define the

Table 4 Descriptive statistics of reply and retweet networks

Network	Absolute energy gap		Word frequency gap	
	Mean	Standard deviation	Mean	Standard deviation
G^{us}_{RE}	0.0086	0.0063	23.5	145.8
G^{us}_{RT}	0.0077	0.0060	148.3	1230.2
G^{sg}_{RE}	0.0051	0.0030	11.2	54.5
G^{sg}_{RT}	0.0104	0.0065	80.5	321.8

first-order frequency difference of word w for day t as $df^*(w,t) = f^*(w,t) - f^*(w,t-1)$.[4] From $\{df^*(w,t)\}$, we derive the mean and standard deviation as $mean^*(w)$ and $stddev^*(w)$, respectively.

Table 4 illustrates the means and standard deviations of absolute energy difference sequence and word frequency difference sequence of the dynamic reply and retweet networks.

Take the largest difference of energy from both G^{us}_{RE} and G^{us}_{RT} on June 28, 2012, as an example. The top three words from retweets with highest frequency difference are 'tax,' 'Obamacar,' and 'scotu' (*S*upreme *C*ourt *o*f *U*nited *S*tates) after stopword removal and word stemming. By searching the web using these keywords, we verified that the Obamacare healthcare law was upheld by the Supreme Court of United States, and there were concerns about tax increase as its outcome. This event attracted a lot of replies and retweets on June 28. The word frequency difference of 'Obamacar' in retweets subsided quickly on June 29, 2012, as shown by a negative df^{RT}('*Obamacar*,' *June* 29) value.

For each day t, we define the *average frequency difference* of the three words w_1, w_2 and w_3 with highest $df^*(\cdot, t)$ as $M^*(t) = \frac{1}{3}\sum_{i=1}^{3} df^*(w_i, t)$. If $M^*(t)$ deviates far away from the mean $mean^*(w)$ w.r.t. the value $stddev^*(w)$, an event is considered to happen on day t.

Formally, we define the normalized $M^*(t)$ on day t as

$$N^*(t) = \frac{M^*(t) - mean^*(w)}{stddev^*(w)}$$

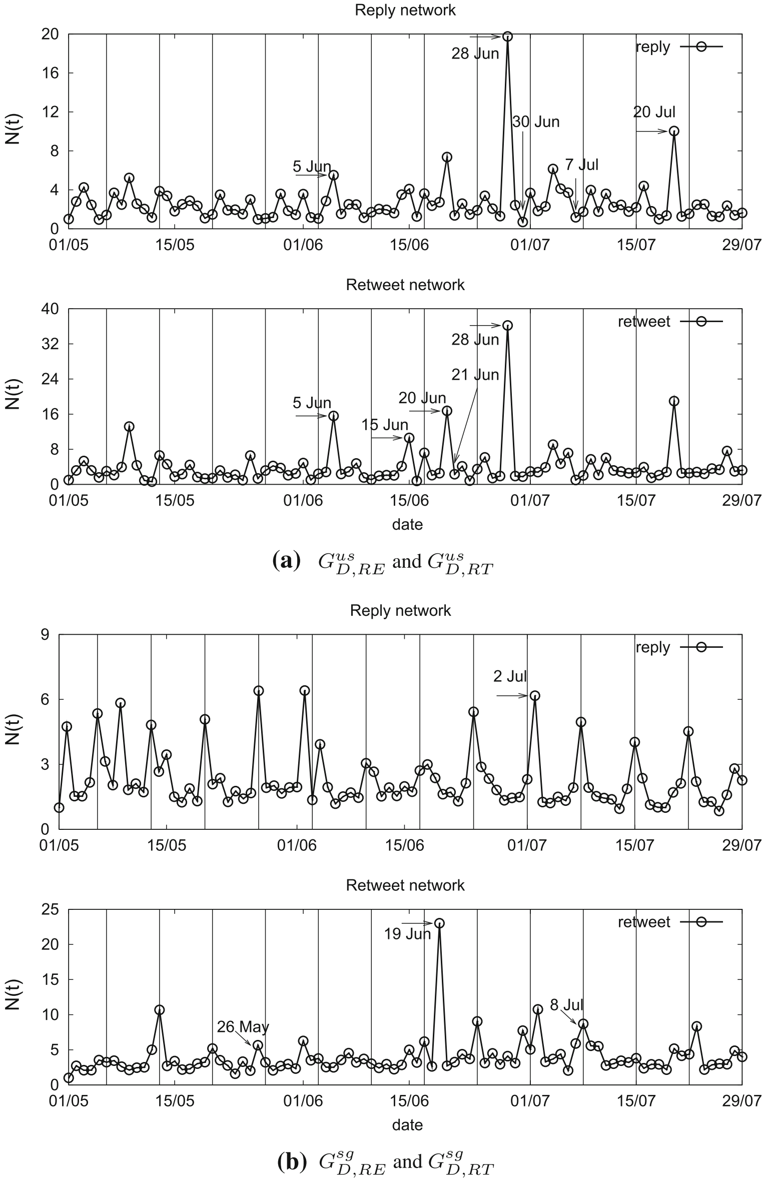
The larger the $N^*(t)$ is, the more likely the top words are able to explain some event on t. Empirically, we use the words with $N^*(t) \geq 8$ to help us to explain internal events. On the other hand, an external event may prevent people from communicating in Twitter. In this case, $N^*(t)$ may be small due to very few users generating tweets. We nevertheless tried to use the frequent words on day t to search the web to confirm if an event is external.

Figure 5a illustrates the $N^*(t)$ values of both G^{us}_{RE} and G^{us}_{RT}. Table 5 lists eight events found from G^{us}_{RE} and G^{us}_{RT} using $\mathcal{R}$-energy. The first column shows the location of event in Fig. 4. The second column shows the date of event and $N^*(t)$ value. The third column shows the top three words derived by top frequency differences in G^{us}_{RE} or G^{us}_{RT} depending on which of the two networks are used to detect the event. The final column shows the description of events manually derived from the Google Search results of the top words.

Similarly, Fig. 5b illustrates the $N^*(t)$ values of both G^{sg}_{RE} and G^{sg}_{RT}. Table 6 lists four events found from G^{sg}_{RE} and G^{sg}_{RT} using $\mathcal{R}$-energy.

Instead of using $\mathcal{R}$-energy, we also experimented with time series of daily reply and retweet counts using a similar event detection method. Unlike the $\mathcal{R}$-energy time series, we

[4] symbol * denotes RE or RT.



(a) $G^{us}_{D,RE}$ and $G^{us}_{D,RT}$

(b) $G^{sg}_{D,RE}$ and $G^{sg}_{D,RT}$

Fig. 5 Normalized difference of word frequencies

could detect only two events on June 28 and June 30 listed in Table 5 and one event on June 19 listed in Table 6. This is because reply and retweet counts fluctuate very much over time. We therefore detect fewer bursty events than that using $\mathcal{R}$-energy. The results also show that $\mathcal{R}$-energy can help detecting events that are different.

Compared the detected events in Table 5 with that of in Table 4 of the original conference version [14], we can observe that four more events are detected from the reply network. This is due to the factor that an unweighted edge in the reply network does not reflect the process

Table 5 Detected events from G_{RE}^{us} and G_{RT}^{us}

Anomaly	Confirmed	Event	Event description
d_{36} (RE)	June 5 (5.5)	Walker (935)	Tom Barr. Wisconsin voters rejected a year-long effort to recall Gov. Scott Walker
		Vote (858)	
		Union (687)	
d_{59} (RE)	June 28 (19.7)	Tax (4777)	Obamacare was the largest tax increase in the history of the world
		Robert (1988)	
		Obamacar (1943)	
d_{61} (RE)	June 30 (1.7)	Natgat (1898)	Honorable Bio visited California to bring the power loss
		Republic (1061)	
		Storm (1049)	
d_{67} (RE)	July 7 (36.2)	Libertyimag (207)	2012 Conquer the Bear Series took place from July 7 to September 9 on Big Bear Lake, CA
		Ron (201)	
		Gibb (178)	
d_{82} (RE)	July 20 (10.0)	Shoot (25,860)	A masked gunman killed 12 people at a midnight showing of the new Batman movie in Aurora, Colorado
		Gun (24,103)	
		Aurora (20,480)	
d_{36} (RT)	June 5 (15.6)	Wisconsin (21089)	The event is also detected by $d_{36}(RE)$
		Walker (20,726)	
		Wirecal (16213)	
d_{47} (RT)	June 15 (10.6)	Obama (17,652)	President Obama was way out of line with his June 15th immigration amnesty
		Immigr (11284)	
		Illeg (10588)	
d_{52} (RT)	June 20 (16.7)	Fastandfuri (23,295)	White House had asserted executive privilege on 'fast and furious' documents
		Holder (19,991)	
		Obama (18,974)	
d_{52} (RT)	June 21 (2.3)	Lebron(3816)	Twitter went down in worst crash in 8 months
		Nba (2694)	
		Twitter (2517)	
d_{60} (RT)	June 28 (36.2)	Tax (52,444)	The event is also detected by $d_{59}(RE)$
		Obamacar (51,390)	
		Scotu (30,247)	

Table 6 Detected events from G^{sg}_{RE} and G^{sg}_{RT}

Anomaly	Confirmed	Event	Event description
d_{63} (RE)	July 2 (6.2)	lol (590)	SIA backed biometrics to improve integrity of medicare, medicaid programs
		sleep (246)	
		sia (208)	
d_{26} (RT)	May 26 (5.7)	Hougangbyelect (2448)	Polling Day for the Hougang by election would be on May 26, 2012
		pap (1645)	
		wp(1607)	
d_{50} (RT)	June 19 (23.0)	Singapor (9073)	Singapore is in Europe was popular on Twitter
		Europ (7543)	
		Bieber (5847)	
d_{69} (RT)	July 8 (8.7)	Anthem (3012)	'China FT' wanted to change Singapore national anthem to Chinese
		Chang (2890)	
		Nation (2735)	

of user interaction. Even though two users have a lively discussion about a hot topic, only an unweighted edge forms between them, while a weighted edge captures the interaction between users. However, we do not observe this phenomenon in the retweet network since users usually do not retweet a tweet many times. This points to the positive effect of defining R-energy on a weighted network.

6.3 Periodic trend pattern detection

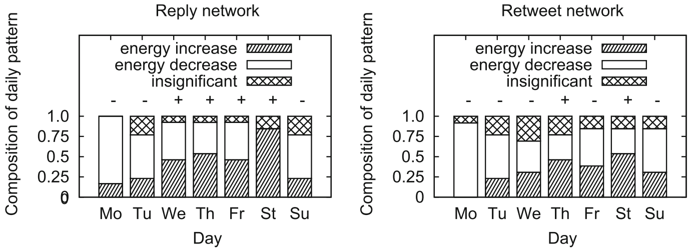
Other than ad hoc events, Mann–Kendall trend test [24] indicates that a periodic pattern significantly exists in $G^{\cdot}_{RE}$ and $G^{\cdot}_{RT}$ of Fig. 4a, b. We also want to detect weekly trend patterns from the figure by examining the regularities in network energy changes. This weekly pattern can be even more distinct when the ad hoc events are removed.

In this section, we therefore focus on detecting weekly pattern. Based on a pre-defined threshold θ ($= 0.1 \times mean(\{d_t\})$), we first derive three kinds of energy changes from the previous day, namely (i) *energy increase* ('$+$'), (ii) *energy decrease* ('$-$'), and (iii) *insignificant change* (*null*). Given a day of a week x, e.g., Tuesday, we count the number of '$+$'s, '$-$'s, and *null*'s and denote them by $p(x)$, $n(x)$, and $null(x)$, respectively. After ignoring the ad hoc events, we increment $p(x)$ if the energy change is more than θ; increment $n(x)$ if the energy change is smaller than $-\theta$; or increment $null(x)$ otherwise. The proportions of '$+$'s and '$-$'s on x across multiple weeks can be defined as:

$$prop(\text{`+'}, x) = \frac{p(x)}{p(x) + n(x) + null(x)}$$

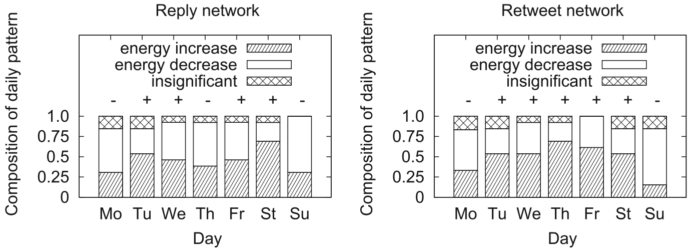
$$prop(\text{`}-\text{'}, x) = \frac{n(x)}{p(x) + n(x) + null(x)}$$

$$prop(null, x) = \frac{null(x)}{p(x) + n(x) + null(x)}$$



(a) G^{us}_{RE} **(b)** G^{us}_{RT}

Fig. 6 Weekly pattern detecting for US users



(a) G^{sg}_{RE} **(b)** G^{sg}_{RT}

Fig. 7 Weekly pattern detecting for Singapore users

Let $max_{prop}(x)$ be maximum value of $prop(\text{'+'}, x)$, $prop(\text{'}-\text{'}, x)$ and $prop(null, x)$. We assign a label l to day x as follows:

$$l = \begin{cases} \text{'+'}, & \text{if } prop(\text{'+'}, x) \text{ equal to } max_{prop}(x) \\ \text{'}-\text{'}, & \text{if } prop(\text{'}-\text{'}, x) \text{ equal to } max_{prop}(x) \\ null, & \text{otherwise} \end{cases} \tag{15}$$

In case of $prop(\text{'+'}, x) = prop(\text{'}-\text{'}, x) = max_{prop}(x)$, we assign a *null* label to the day x.

For example, suppose out of 13 weeks, there are 12 Mondays with '$-$'s, one with '$+$' and zero with *null*. The compositions of positive, negative, and null energy changes on Monday are 7.7%, 92.3%, and 0%, respectively. Monday therefore is assigned to '$-$'. By assembling the proportions of positive, negative, null energy changes for different days of week, we obtain the *weekly trend pattern* of $G^{\cdot}_{RE}$ and $G^{\cdot}_{RT}$.

Figure 6 illustrates the composition of weekly pattern for G^{us}_{RE} and G^{us}_{RT}. According to label assignment rule, we obtain the weekly trend pattern '$- - + + + + -$' for G^{us}_{RE}, and another weekly trend pattern '$- - - + - + -$' for G^{us}_{RT}. Other than Friday, the two weekly trend patterns obtained from G^{us}_{RE} and G^{us}_{RT} are very similar.

Figure 7 illustrates the composition of weekly pattern for G^{sg}_{RE} and G^{sg}_{RT}. According to label assignment rule, we obtain the weekly trend pattern '$- + + - + + -$' for G^{sg}_{RE}, and another weekly trend pattern '$- + + + + + -$' for G^{sg}_{RT}. Other than Friday, the two weekly trend patterns obtained from G^{sg}_{RE} and G^{sg}_{RT} are very similar. In addition, the weekly trend patterns for two countries are very similar.

From the weekly trend pattern, we can casually conclude that users are less likely to tweet on Saturdays but tweet a lot on Sundays as well as Mondays.

7 Conclusion

In many applications, an obvious characteristic is tending to be modeled them with large-scale networks, such as protein–protein interaction networks, neural networks, the Internet, the World Wide Web, social networks, and scientific collaboration networks. To understand the robustness of network with millions or billions vertices is an important and challenged task both in theory and application. In this paper, based on the normalized Laplacian matrix, we define the $\mathcal{R}$-energy for a weighted network to measure its robustness.

In theory, the $\mathcal{R}$-energy is related to the average 2-step commute probability. Vertex has smaller 2-step commute probability if it has higher connectivity. It indicates that the high robustness network has smaller $\mathcal{R}$-energy. Furthermore, the complexity of computing $\mathcal{R}$-energy is $O(|V| + |E|)$ since the algorithm just scans the entire weighted network once after obtaining the weighted degrees of all vertices. Therefore, it can be easily applied to large networks. Our empirical study illustrates that our algorithm takes less than 120 s for a network with millions vertices. In practice, we can find some patterns of robustness of static networks and some anomaly cases of dynamic networks.

In this work, we have only considered the dynamic information network offline, and thus, it may fail for online event detection. Since online event detection may be more helpful in real-world applications. To address this issue, we plan to extend our $\mathcal{R}$-energy to online detect events on real social network platforms, such as Facebook and Twitter. In addition, we can construct the interaction network for a particular event. In terms of the connection of this information network, we can also apply our $\mathcal{R}$-energy to predict its future trend. Thus, we plan to investigate how to accurately predict its burst.

Acknowledgements This work has been supported by the National Natural Science Foundation of China under Grant No. U1911203, Alibaba Group through the Alibaba Innovation Research Program, and the National Natural Science Foundation of China under Grant No. 61877018.

Appendix

Appendix A: Operators on the vertices

Eigenvalues and eigenvectors are used to understand what happens when one repeatedly applies an operator to a vector. If we have an operator that is naturally associated with a weighted network G, then properties of this operator will be revealed by its eigenvalues and eigenvectors. The first operator one typically associates with a weighted network G is its adjacency operator. To understand operator, one must view vectors $g \in R^n$ as function from a vertex to a Real. That is, they should be understood as a vector of R^V. When we apply the adjacency operator to such a function, the result value at a vertex v_i is the sum of the values of the function $g \in R^V$ over all neighbors of vertex v_i.

$$(A_G g)(v_i) = \sum_{v_j:(v_i,v_j)\in E} g(v_j).$$

Similarly, we can derive weighted Laplacian and weighted normalized Laplacian operators as:

$$(L_G g)(v_i) = \sum_{(v_i,v_j)\in E} w_G(i,j)(g(v_i) - g(v_j));$$

$$(N_G g)(v_i) = g(v_i) - \sum_{(v_i,v_j)\in E} g(v_j)\frac{w_G(i,j)}{\sqrt{d(v_i)d(v_j)}}.$$

Appendix B: Proof of Theorem 1

Proof (i) follows from considering the Rayleigh quotient and the trace of N_G. Since sum of every column or row of N_G is 0, 0 is eigenvalue of N_G associated with eigenvector $v_1 = \overrightarrow{1} = (1, 1, \dots, 1)^T$. Let R^V be the set of functions from V to R,

$$R^V = \{g : V \longrightarrow R\}.$$

If g be a function of R^V, we can view g as a column vector. Then,

$$\begin{aligned}\frac{\langle g, N_G g\rangle}{\langle g, g\rangle} &= \frac{\langle g, D_G^{-1/2} L_G D_G^{-1/2} g\rangle}{\langle g, g\rangle} = \frac{\langle f, L_G f\rangle}{\langle f, f\rangle}\\ &= \frac{\sum_{(v,u)\in E} w_G(u,v)(f(u)-f(v))^2}{\sum_v f(v)^2 d(v)}\end{aligned}$$

where $g = D_G^{1/2} f$ and $\langle f, g\rangle = \sum_x f(x)g(x)$. In terms of Rayleigh quotient, $\zeta_1 = \inf_g \frac{\langle g, N_G g\rangle}{\langle g, g\rangle} \geq 0$. And the fact that $((f(u) - f(v))^2) \leq 2(f(v)^2 + f(u)^2)$, therefore,

$$\begin{aligned}\zeta_n &= \sup_g \frac{\langle g, N_G g\rangle}{\langle g, g\rangle}\\ &= \sup_f \frac{\sum_{(v,u)\in E} w_G(u,v)(f(u)-f(v))^2}{\sum_v f(v)^2 d(v)} \leq 2.\end{aligned}$$

Equality holds for $i = n - 1$ when $f(v) = -f(u)$ for every edge $(v, u) \in E$.

In addition, the sum of all eigenvalues of N_G is n. Except ζ_1, the mean of remaining eigenvalues is $\frac{n}{n-1}$. We therefore derive $\zeta_2 \leq \frac{n}{n-1} \leq \zeta_n$.

(ii) $\Longleftarrow$ follows from the equation $det(N_G - \zeta I) = 0$.

$\Longrightarrow$: Let $N_G = \Lambda\Pi\Lambda^{-1}$ and $\Pi = diag(0, \frac{n}{n-1}, \dots, \frac{n}{n-1})$. We have

$$\begin{aligned}\left(N_G - \frac{n}{n-1}I\right)N_G &= \left(\Lambda\Pi\Lambda^{-1} - \frac{n}{n-1}I\right)\Lambda\Pi\Lambda^{-1}\\ &= \Lambda\Pi^2\Lambda^{-1} - \Lambda\frac{n}{n-1}\Pi\Lambda^{-1} = 0.\end{aligned}$$

We also have $N_G(N_G - \frac{n}{n-1}I) = 0$. Thus, every column and row of $N_G - \frac{n}{n-1}I$ is an eigenvector of N_G associated with eigenvalue zero. Notice that the eigenvector of N_G associated with eigenvalue zero is $c(\sqrt{d(v_1)}, \sqrt{d(v_2)}, \dots, \sqrt{d(v_n)})$ for some scalar constant c. Matrix $N_G - \frac{n}{n-1}I$ can be represented as

$$\begin{pmatrix} c_1\sqrt{d(v_1)} & c_1\sqrt{d(v_2)} & \cdots & c_1\sqrt{d(v_n)}\\ c_2\sqrt{d(v_1)} & c_2\sqrt{d(v_2)} & \cdots & c_2\sqrt{d(v_n)}\\ & \cdots & \cdots & \\ c_n\sqrt{d(v_1)} & c_n\sqrt{d(v_2)} & \cdots & c_n\sqrt{d(v_n)} \end{pmatrix}$$

Because $N_G - \frac{n}{n-1}I$ is symmetric and has identical diagonal values, we have $c_i = c_j$ and $\sqrt{d(v_i)} = \sqrt{d(v_j)}$ for any i, j. Therefore, $N_G = \frac{n}{n-1}I + cJ$, where c is a scalar constant and J is a $n \times n$ matrix where each entry is equal to 1. To summarize, network G must be a clique of equal edge weights.

(iii)$\Longrightarrow$: follows from the fact that the union of two disjoint networks has as its spectrum the union of the spectra of the original network.

$\Longleftarrow$: it is correct if $i = 1$. Obviously, the network is disconnected if $i > 1$. The network has i strongly connected component, otherwise $\zeta_i = 0$ or $\zeta_i \neq 0$. □

References

1. Albert R, Jeong H, Barabási AL (2000) The internet's Achilles' heel: error and attack tolerance in complex networks. Nature 406:378–382
2. Balakrishnan R (2004) The energy of a graph. Linear Algebra Appl 387:287–295
3. Ball F, Mollison D, Scalia-Tomba G (1997) Epidemics with two levels of mixing. Ann Appl Probab 7(1):46–89
4. Barthélemy M, Barrat A, Pastor-Satorras R, Vespignani V (2005) Characterization and modelling of weighted networks. Phys A 346:34–43
5. Bobkov SG, Houdré C, Tetali P (2000) $\lambda_{+\infty}$ vertex isoperimetry and concentration. Combinatorica 20(2):153–172
6. Callaway DS, Newman MEJ, Strogatz SH, Watts DJ (2000) Network robustness and fragility: percolation on random graphs. Phys Rev Lett 85:5468–5471
7. Cavers M, Fallat S, Kirkland S (2010) On the normalized Laplacian energy and general Randić index r_{-1} of graphs. Linear Algebra Appl 433(1):172–190
8. Cohen R, Erez K, ben Avraham D, Havlin S (2000) Resilience of the internet to random breakdowns. Phys Rev Lett 85:4626–4228
9. Day J, So W (2007) Singular value inequality and graph energy change. Electron J Linear Algebra 16:291–299
10. Day J, So W (2008) Graph energy change due to edge deletion. Linear Algebra Appl 428:2070–2078
11. Dekker AH, Colbert BD (2004) Network robustness and graph topology. In: ACSC, pp 359–368
12. Eubank S, Guclu H, Kumar V, Marathe M, Srinivasan A, Toroczkai Z, Wang N (2004) Modeling disease outbreaks in realistic urban social networks. Nature 429:180–184
13. Fiedler M (1973) Algebraic connectivity of graphs. Czechoslov Math J 23(98):298–305
14. Gao M, Lim E-P, Lo D (2013) R-energy for evaluating robustness of dynamic networks. In: WebSci
15. Gleditsch KS (2002) Expanded trade and GDP data. J Confl Resolut 46:712–724
16. Gutman I (1978) The energy of a graph. BeT Math Stai'ist Sekt FOTschnngsz Gmz 103:1–22
17. Hasegawa T, Masuda N (2011) Robustness of networks against propagating attacks under vaccination strategies. J Stat Mech Theory Exp 429:P09014
18. Hoory S, Linial N, Wigderson A (2006) Expander graphs and their applications. J Bull Am Math Soc 43(4):439–562
19. Jamakovic A, Mieghem PV (2008) On the robustness of complex networks by using the algebraic connectivity. In: Networking, pp 183–194
20. Jamakovic A, Uhlig S (2008) On the relationships between topological measures in real-world networks. NHM 3(2):345–359
21. Java A, Song X, Finin T, Tseng B (2007) Why we twitter: understanding microblogging usage and communities. In: WEBKDD
22. Leskovec J, Kleinberg J, Faloutsos C (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In: SIGKDD, pp 177–187
23. Malliaros FD, Megalooikonomou V, Faloutsos C (2012) Fast robustness estimation in large social graphs: Communities and anomaly detection. In: SDM, pp 942–953
24. Mann HB (1945) Nonparametric tests against trend. Econometrica 13:245–259
25. Newman MEJ (2001) The structure of scientific collaboration networks. Proc Natl Acad Sci 404–409
26. Robbiano M, Jiménez R (2009) Applications of a theorem by Ky Fan in the theory of graph energy. MATCH Commun Math Comput Chem 62:537–552
27. Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. Nature 393(6684):440–442
28. Wei P, Sun D (2011) Weighted algebraic connectivity: an application to airport transportation network. In: The 18th IFAC world congress, pp 404–409
29. Yang YT (2009) Cheeger constant and Cheeger inequality. Technique report, pp 1–17
30. Zhou B (2010) More on energy and Laplacian energy. MATCH Commun Math Comput Chem 64:75–84

31. Zhou B, Gutman I, Aleksié T (2008) A note on the Laplacian energy of graphs. MATCH Commun Math Comput Chem 60:441–446

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

Jianbing Zheng is currently a PhD candidate on the School of Data Science and Engineering, East China Normal University. His research interests include location-based services, precision marketing, social network mining and analysis, software engineering, data fusion and association analysis, and so on.

Ming Gao is currently an associate professor in the School of Data Science and Engineering at East China Normal University, China. He received his doctorate from the School of Computer Science, Fudan University. After that, he conducted three-year postdoctoral research at Singapore Management University for the project LiveLabs. His research interests include knowledge engineering, user profiling, social network analysis and mining, etc. His work appears in major international journals and conferences including DMKD, TKDE, KAIS, ACL, ICDE, ICDM, SIGIR, etc.

Ee-Peng Lim is a professor of information systems at Singapore Management University. His research interests include social network and web mining, information integration, and digital libraries. He has published more than 300 refereed journal and conference papers in these areas. He is currently the faculty director of the Living Analytics Research Center which focuses on urban and social analytics. He is a member of the Singapore's Social Science Research Council and also serves as the Steering Committee Chair of Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD).

David Lo received the PhD degree in computer science from the National University of Singapore, in 2008. He is a ACM Distinguished Member and an associate professor of Information Systems at Singapore Management University. His research interests include the intersection of software engineering and data science, encompassing socio-technical aspects and analysis of different kinds of software artifacts, with the goal of improving software quality and developer productivity. His work has been published in premier and major conferences and journals in the area of software engineering, AI, and cybersecurity.

Cheqing Jin received the bachelor's and master's degrees from Zhejiang University, China, and the PhD degree in computer science from Fudan University, China, in 1999, 2002, and 2005, respectively. He is a professor with East China Normal University, China. Before joining East China Normal University, China on October 2008, he worked as an assistant professor with the East China University of Science and Technology, China. He is the winner of the Fok Ying Tung Education Foundation Fourteenth Young Teacher Award. He is a member of Database Technology Committee of China Computer Federation and serves as a young associate editor of the Frontiers of Computer Science, an SCI journal. He has co-authored more than 80 papers, some of which received excellent paper awards, such as the Best Paper Award of the Chinese Journal of Computers and Best Paper Award of pervasive computing and embedding from the Shanghai Computer Society. His research interests include streaming data management, location-based services, uncertain data management, and sharing database management systems.

Aoying Zhou is a professor on School of Data Science and Engineering (DaSE) at East China Normal University (ECNU), where he is heading DaSE and vice president of ECNU. He is the winner of the National Science Fund for Distinguished Young Scholars supported by NSFC and the professorship appointment under Changjiang Scholars Program of Ministry of Education. His research interests include Web data management, data intensive computing, in-memory cluster computing, and benchmark for big data. His works appear in major international journal and conferences including Sigmod, TKDE, VLDB, SIGIR, KDD, WWW, ICDE, ICDM, etc.