Canadian Journal of Physics
Revue canadienne de physique

# Eigenvalues of transition weight matrix and eigentime identity of weighted network with two hub nodes

| | |
|---|---|
| Journal: | *Canadian Journal of Physics* |
| Manuscript ID | cjp-2017-0274.R1 |
| Manuscript Type: | Article |
| Date Submitted by the Author: | 06-Aug-2017 |
| Complete List of Authors: | Zou, Jiahui; Jiangsu University<br>Dai, Meifeng; Nonlinear Scientific Research Center, Faculty of Science<br>Wang, Xiaoqian; Jiangsu University<br>Tang, Hualong; Jiangsu University<br>He, Di; Jiangsu University<br>Sun, Yu ; Jiangsu University<br>Su, Weiyi; Jiangsu University |
| Keyword: | weighted network, eigentime identity, weighted counting of spanning trees, eigenvalue, mul- tiplicity |
| Is the invited manuscript for consideration in a Special Issue? : | N/A |
| | |

# Eigenvalues of transition weight matrix and eigentime identity of weighted network with two hub nodes

Jiahui Zou[a] Meifeng Dai [a1] Xiaoqian Wang[a] Hualong Tang[a] Di He[a] Yu Sun[a] Weiyi Su[b]

[a] *Institute of Applied System Analysis, Jiangsu University,*
*Zhenjiang, Jiangsu 212013, P.R. China*

[b] *Department of Mathematics, Nanjiang University, Nanjing, 210093, P. R. China*

## ABSTRACT

The eigenvalues of the normalized Laplacian of a graph provide information on its structural properties and also on some relevant dynamical aspects, in particular those related to weight-dependent walk. In this paper, we first present a study on the transition weight matrix of a weighted network. In order to get the eigentime identity for weight-dependent walk and weighted counting of spanning trees, we need to obtain all the eigenvalues and their multiplicities of the transition weight matrix. Then we obtain the recursive relationship of its eigenvalues at two successive generations of transition weight matrix. By substituting, we can obtain the relationship of normalized Laplacian matrix's eigenvalues at two successive generations. Using the relationship and Vietas formulas, we obtain the scalings of the eigentime identity. Afterwards, we classify normalized Laplacian matrix's eigenvalues and compute the product of all nonzero normalized Laplacian eigenvalues by the product recursive relationship. The product is used to obtain weighted counting of spanning trees. Finally, by weighted counting of spanning trees, we validate the obtained eigenvalues and their multiplicities. The obtained results show that the weight factor has a strong effect on the behavior of weight-dependent walks.

***Keywords:*** *weighted network, eigentime identity, weighted counting of spanning trees, eigenvalue, multiplicity*

PACS Nos: 89.75.-k; 89.75.Hc; 89.75.Fb

## 1 Introduction

In the recent years, the study of networks associated with complex systems has received much attention of researchers from different scientific fields, especially the weighted network [1, 2, 3]. The eigentime identity of weighted network has gained much interest [4, 5, 6].

In the past few years, there has been an increasing interest in the study of the normalized Laplacian as many measures for random walks on un-weighted networks. The eigenvalues and eigenvectors of normalized Laplacian of the associated graph are related to the hitting time, mixing time and Kemeny's constant which can be used as a measure of efficiency of navigation on the network [7, 8, 9]. Julaiti et al. mentioned that the sum of reciprocals of each nonzero eigenvalues of normalized Laplacian matrix for a network determines the eigentime identity for random walks on the network, which is a global characteristic of the network, and reflects the architecture of the whole network [5]. Zhang et al. presented a first study on the transition weight matrix of a family of weight driven networks. They applied the obtained eigenvalues to derive a closed-form expression for the random target access time for biased random walks occurring on the studied weighted networks [10]. Previous works about spectra of the transition matrix were limited to binary networks, and the influence of inhomogeneous weight distribution on the spectral properties of transition matrix still remains unknown. Now we study the weighted network with weight distribution decided by the weighter factor.

There are many differences between this model (i.e., weighted network with two hub nodes) and other weighted networks. Firstly, the weighted networks are divided into two cases by the total number of nodes. One case is that the total number of nodes is exponential [6]. The Laplacian matrix is used to study the

[1] Corresponding author. Tel.: +86 13815158555.
E-mail address: daimf@mail.ujs.edu.cn , daimf0225@163.com (M. Dai).

structure and dynamics of the network. The other case is that the number of nodes is not exponential [10, 11]. The normalized Laplacian matrix is used to study the architectures and dynamics of the network. Since the total number of nodes of this model is not exponential, we use the normalized Laplacian matrix to study the architectures and dynamics of the network. Comparing with the model in Ref. [10], the weight of each edge at two successive generations of this model constructed by the weight factor is invariable, while the weight of each edge at two successive generations is variable in Ref. [10]. Comparing with the two models in Ref. [11], the number of hub nodes is different from the polymer networks, and the solution method is different from treelike networks in Ref. [11]. We use the definition of eigenvalues and eigenvectors to compute the relationship of normalized Laplacian matrix's eigenvalues at two successive generations while Dai et al. (in Ref. [11]) use the characteristic equation of normalized Laplacian matrix to compute the relationship at two successive generations of treelike networks.

In this paper, intuited by the weight driven networks [10] and the weighted networks with weight factor [12, 13, 14], the weighted network with two hub nodes is built. We study analytically the eigentime identity for normalized Laplacian matrix of weighted network with two hub nodes carrying a weight factor. Based on the particular construction of this network, we get all the eigenvalues and their corresponding multiplicities for transition weight matrix of weighted network with two hub nodes. Using the obtained eigenvalues for transition weight matrix, we can obtain the eigenvalues for normalized Laplacian matrix of weighted network with two hub nodes. Then, we could deduce an explicit expression for the eigentime identity and its leading scalings, which is different from those previously obtained for binary heterogeneous networks, implying that the weight factor has an important impact for weight-dependent walk behavior. Furthermore, we use two methods to compute the weighted counting of spanning trees in the studied network. By comparing the solutions got from different methods, we could verify the validity of our computation for the eigenvalues.

## 2 The weighted network with two hub nodes and related quantities

The weighted network with two hub nodes, parameterized by a positive number $r$, is constructed in an iterative manner [15]. We denote the network after $g$ ($g \geq 0$) steps by $F_g$, which is built as follows. For $g = 0$, $F_0$ is an edge with unit weight connecting two nodes. For $g \geq 1$, $F_g$ is obtained from $F_{g-1}$ by performing the following operations. For each edge with weight $w$ in $F_{g-1}$, we add two new nodes and link them to either end of the edge, respectively, and each new edge carries weight $rw$. Here we call $r$ the weight factor. And for the old edges existed in $F_{g-1}$, the weight remains the same. Figure 1 illustrates the network generation process from $g = 0$ to 2.

Let $N_g$, $E_g$, $Q_g$ denote the total number of nodes, the total number of edges, and the total weight of all edges in $F_g$, respectively. By construction, for $g \geq 0$, we have

$$Q_g = (1+4r)Q_{g-1},$$

which under the initial condition $Q_0 = 1$ yields

$$Q_g = (1+4r)^g. \tag{1}$$

Furthermore, by construction, for $g \geq 0$, we have

$$E_g = 5E_{g-1} = 5^g E_0 = 5^g,$$

and

$$N_g = E_g + 1 = 5^g + 1. \tag{2}$$

For an edge connecting two nodes $i$ and $j$ in $F_g$, we use $w_{ij}(g)$ to denote its weight. Let $d_i(g)$, $s_i(g)$ denote the degree and strength of node $i$ in $F_g$, respectively, which is added to the network at generation $g_i$. It is easy to obtain

$$d_i(g) = 3d_i(g-1) = 3^{g-g_i},$$

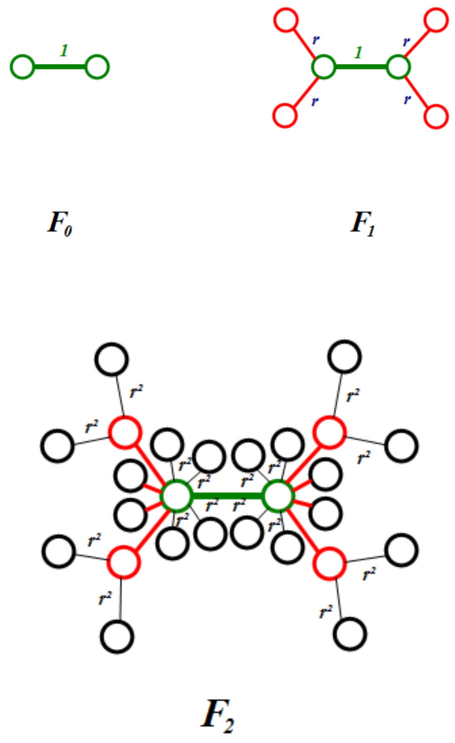

Figure 1: Illustration of the growth for a network $F_0$, $F_1$ and $F_2$ for example. The weight of edge (green on-line) is 1. The weight of edge (red on-line) is $r$. The weight of edge (black on-line) is $r^2$.

and

$$\begin{aligned} s_i(g) &= \sum_{j \in v(i)} w_{ij}(g) = (1+2r)s_i(g-1) \\ &= (1+2r)^{g-g_i} s_i(g_i), \end{aligned} \tag{3}$$

where $v(i)$ is the set of neighbors of $i$ in $F_g$.

## 3 Eigentime identity and weighted counting of spanning trees

Let $W_g$ be the generalized adjacency matrix (weight matrix) of $F_g$. The entries $W_g(i,j)$ of $W_g$ are defined as follows: $W_g(i,j) = w_{ij}(g)$ if nodes $i$ and $j$ are adjacent in $F_g$, or $W_g(i,j) = 0$ otherwise. Afterwards, the transition weight matrix for weight-dependent walk ( i.e., the walker moves to any of its neighbors with probability proportional to the weight of edge linking them [16, 17] ) in $F_g$, denoted by $T_g$, is defined as $T_g = S_g^{-1} W_g$, where $S_g$ is the diagonal strength matrix of $F_g$ with its $ith$ diagonal entry being the strength $s_i(g)$ of node $i$. Thus, The $(i,j)th$ element of $T_g$ is $T_g(i,j) = w_{ij}(g)/s_i(g)$, which represents the local transition probability for a walker going from node $i$ to node $j$.

### 3.1 Eigentime identity

The eigentime identity and the relationship of the eigentime identity and the normalized Laplacian eigenvalues of weighted networks are introduced in [10].

Transition weight matrix $T_g$ describes the weight-dependent walk in $F_g$, and thus various interesting quantities for weight-dependent walk are reflected in eigenvalues of the transition matrix. For example, the sum of reciprocals of 1 minus each eigenvalue (excluding eigenvalue 1 itself) of transition weight matrix $T_g$ determines the random target access time, also called eigentime identity, in $F_g$ [18, 19, 20].

Let $H_{ij}(g)$ denote the mean first-passage time for weight-dependent walk from node $i$ to node $j$ in $F_g$, defined as the expected time for a walker starting from node $i$ to visit node $j$ for the first time. Let $\pi = (\pi_1, \pi_2, \ldots, \pi_{N_g})^T$ represent the steady state distribution on $F_g$ [8, 21], where $\pi_i = s_i(g)/(2Q_g)$ satisfying $\sum_{i=1}^{N} \pi_i = 1$ and $\pi^T T_g = \pi^T$. The eigentime identity, denoted by $H_g$, for weight-dependent walk on $F_g$, is defined as the expected time needed by a walker from a node $i$ to another target node $j$, chosen randomly from all nodes according to the steady state distribution, that is,

$$H_g = \sum_{j=1}^{N_g} \pi_j H_{ij}(g),$$

which does not depend on the starting node and can be recast as

$$H_g = \sum_{i=1}^{N_g} \pi_i \sum_{j=1}^{N_g} \pi_j H_{ij}(g) = \sum_{j=1}^{N_g} \pi_j \sum_{i=1}^{N_g} \pi_i H_{ij}(g).$$

Since $H_g$ can be looked upon as the average trapping time of a special trapping problem [22], it encodes much useful information about trapping in $F_g$.

Since $T_g$ is asymmetric, we can normalize it to obtain the following real and symmetric matrix $P_g$, which is defined as

$$P_g = S_g^{-\frac{1}{2}} W_g S_g^{-\frac{1}{2}} = S_g^{\frac{1}{2}} T_g S_g^{-\frac{1}{2}}.$$

We introduce a matrix $L_g = I_g - P_g$, where $I_g$ denotes the $N_g \times N_g$ identity matrix. Actually, $L_g$ is the normalized Laplacian matrix [23, 24, 25] of $F_g$. Let $\lambda_i(g) (1 \leq i \leq N_g)$ be the $N_g$ eigenvalues of $P_g$. By definition, for any $i$, $\sigma_i(g) = 1 - \lambda_i(g)$ is an eigenvalue of $L_g$. And $\sigma_1(g) = 1 - \lambda_1(g) = 0$. It can be proved [26] that $H_g$ can be represented in terms of the nonzero eigenvalues of $L_g$, given by

$$H_g = \sum_{i=2}^{N_g} \frac{1}{\sigma_i(g)}. \tag{4}$$

### 3.2 Weighted counting of spanning trees

For a weighted network $\mathcal{G}$, the set of its spanning trees is denoted by $\Upsilon(\mathcal{G})$. For a tree $T \in \Upsilon(\mathcal{G})$, its weight $w(T)$ is defined to be the product of weights of all edges $e$ in $T$, that is, $w(T) = \Pi_{e \in T} w_e$, where $w_e$ is the weight of edge $e$. Let $\tau(\mathcal{G})$ denote the weighted counting of spanning trees of $\mathcal{G}$, which is defined by $\tau(\mathcal{G}) = \sum_{T \in \Upsilon(\mathcal{G})} w(T)$.

Since $F_g$ is a tree, which has only one spanning tree, in fact $F_g$ itself. Then, the weighted counting of spanning trees in $F_g$ is $\tau(F_g) = \Pi_{e \in F_g} w_e(g)$, where the product is running over the weight $w_e(g)$ of all edges $e \in F_g$. According to previous results [26, 27], we have

$$\tau(F_g) = \frac{\prod_{i=1}^{N_g} s_i(g) \prod_{i=2}^{N_g} \sigma_i(g)}{\sum_{i=1}^{N_g} s_i(g)}. \tag{5}$$

From Eqs. (4) and (5), we know that both eigentime identity and weighted counting of spanning trees are computed by the eigenvalues of normalized Laplacian matrix. Eigentime identity is quantified as the sum of reciprocals of all nonzero normalized Laplacian eigenvalues while weighted counting of spanning trees is quantified as the product of of all nonzero normalized Laplacian eigenvalues. So, we need to compute the eigenvalues of normalized Laplacian matrix. The eigenvalue $\sigma_i$ of normalized Laplacian matrix and the eigenvalue $\lambda_i$ of the transition weight matrix satisfies $\sigma_i = 1 - \lambda_i$. So, in the next section, we will study the eigenvalues and their multiplicities of the transition weight matrix for the network.

## 4 Eigenvalues and multiplicities of transition weight matrix

To obtain eigentime identity and weighted counting of spanning trees, we will study the eigenvalues and their multiplicities of the transition weight matrix for the network.

### 4.1 Recursive relationship of eigenvalues at two successive generations

In this subsection, the recursive relationship of eigenvalues at two successive generations is studied.

We now consider the eigenvalues and their corresponding eigenvectors of $T_g$. Since $T_g$ is asymmetric, we can normalize it to obtain the following real and symmetric matrix $P_g$, which is defined as

$$P_g = S_g^{-\frac{1}{2}} W_g S_g^{-\frac{1}{2}} = S_g^{\frac{1}{2}} T_g S_g^{-\frac{1}{2}}. \tag{6}$$

By definition of $P_g$, the $(i,j)th$ element of $P_g$ is $P_g(i,j) = \frac{w_{ij}(g)}{\sqrt{s_i(g)}\sqrt{s_j(g)}}$. As $P_g$ and $T_g$ are similar matrixes, they have the connection as follows. First and foremost, they have the same set of eigenvalues. Furthermore, if $\phi$ is an eigenvector of matrix $P_g$ associated with eigenvalue $\lambda$, then $S^{-\frac{1}{2}}\phi$ is an eigenvector of $T_g$ corresponding to eigenvalue $\lambda$. Therefore, we reduce the problem of finding eigenvalues for an asymmetric matrix $T_g$ to the issue of determining eigenvalues for a symmetric matrix $P_g$.

Suppose that $\lambda$ is an eigenvalue of $P_g$, and $\phi = (\phi_1, \phi_2, \ldots, \phi_{N_g})^T$ is its corresponding eigenvector, where $\phi_j$ is the component corresponding to node $j$ in $F_g$. Let $\tilde{\phi}$ be a vector of dimension $N_{g-1}$ that is obtained from $\phi$ by restricting its components to the old nodes, namely, nodes generated before or at iteration $g-1$. As will be shown below, $\tilde{\phi}$ is an eigenvector of $P_{g-1}$, associated with eigenvalue $\tilde{\lambda}$, from which $\lambda$ is generated. By definition, we have

$$\lambda\phi = P_g\phi. \tag{7}$$

Let $o$ be an old node in $F_g$. According to Eq. (7),

$$\lambda\phi_o = \sum_{i\epsilon v(o)} P_g(o,i)\phi_i, \tag{8}$$

where $v(o)$ denotes the set of the $d_o(g)$ neighbors of node $o$. It is obvious that $v(o)$ consists of two parts. One part is the set of the $d_o(g-1)$ old neighbors of node $o$. The other part is $d_o(g) - d_o(g-1)$ new neighbors. Let $\tilde{v}(o)$, $\bar{v}(o)$ denote, respectively, the old neighbors and the new ones. For each new neighboring node $i \in \bar{v}(o)$, the component $\phi_i$ satisfies

$$\lambda\phi_i = \sum_{j\epsilon v(o)} P_g(i,j)\phi_j = P_g(o,i)\phi_o,$$

implying

$$\phi_i = \frac{P_g(o,i)}{\lambda}\phi_o. \tag{9}$$

In the case $\lambda \neq 0$, inserting Eq. (9) into Eq. (8), we obtain

$$\begin{aligned}\lambda\phi_o &= \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i + \sum_{i\epsilon\bar{v}(o)} P_g(o,i)\phi_i \\ &= \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i + \sum_{i\epsilon\bar{v}(o)} \frac{P_g^2(o,i)}{\lambda}\phi_o.\end{aligned} \tag{10}$$

For each new neighboring node $i \in \bar{v}(o)$, one has

$$P_g(o,i) = \sqrt{s_i(g)/s_o(g)}. \tag{11}$$

Considering the three relations $[P_g(o,i)]^2 = s_i(g)/s_o(g)$, $\sum_{i\epsilon\bar{v}(o)} s_i(g) = 2rs_o(g-1)$ and $s_o(g) = (1+2r)s_o(g-1)$, Eq. (10) can be solved as

$$\begin{aligned}\lambda\phi_o &= \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i + \sum_{i\epsilon\bar{v}(o)} \frac{s_i(g)}{\lambda s_o(g)}\phi_o \\ &= \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i + \frac{2rs_o(g-1)}{\lambda s_o(g)}\phi_o \\ &= \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i + \frac{2rs_o(g-1)}{\lambda(1+2r)s_o(g-1)}\phi_o \\ &= \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i + \frac{2r}{(1+2r)\lambda}\phi_o.\end{aligned}$$

Thus, we can obtain

$$[\lambda - \frac{2r}{(1+2r)\lambda}]\phi_o = \sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i. \tag{12}$$

Eq. (12) only involves old nodes, which were already existing at iteration $g-1$. For $\lambda \neq 0$, Eq. (12) is true for an arbitrary node present at generation $g-1$. It is not difficult to see that for each old neighboring node $i \in \tilde{v}(o)$,

$$P_g(o,i) = \frac{1}{1+2r}P_{g-1}(o,i).$$

We also have the following corresponding equation for the old node $o$ at iteration $g-1$:

$$\begin{aligned}\tilde{\lambda}\phi_o &= \sum_{i\epsilon\tilde{v}(o)} P_{g-1}(o,i)\phi_i \\ &= (1+2r)\sum_{i\epsilon\tilde{v}(o)} P_g(o,i)\phi_i. \end{aligned} \tag{13}$$

Comparing Eq. (12) with Eq. (13), we obtain that

$$\tilde{\lambda} = (1+2r)\lambda - \frac{2r}{\lambda}. \tag{14}$$

Solving the quadratic equation in the variable $\lambda$ given by Eq. (14) yields

$$\lambda_{\pm} = \frac{\tilde{\lambda} \pm \sqrt{\tilde{\lambda}^2 + 8(1+2r)r}}{2(1+2r)}, \tag{15}$$

which shows that each eigenvalue $\tilde{\lambda}$ of $P_{g-1}$ gives rise to two eigenvalues of $P_g$, $\lambda_+$ and $\lambda_-$.

### 4.2 Multiplicities of eigenvalues

In this subsection, we will determine the multiplicity of each eigenvalue for matrix $P_g$. Let $D_g^{mul}(\lambda)$ represent the multiplicity of eigenvalue $\lambda$ for matrix $P_g$.

For $F_0$ and $F_1$, the eigenvalues and their multiplicities can be calculated directly. The eigenvalues of $P_0$ are 1 and $-1$. For $P_1$, its eigenvalues are $1, -1, 0, \frac{2r}{1+2r}$ and $-\frac{2r}{1+2r}$, where two pairs of eigenvalues $1, -\frac{2r}{1+2r}$ and $-1, \frac{2r}{1+2r}$ are generated, respectively, by eigenvalues 1 and $-1$ of $P_0$. For $g \geq 2$, the eigenvalues of matrix $P_g$ display the following remarkable nature. To begin with, every eigenvalue appearing at current generation $g_i$ always exists at the next generation $g_i + 1$, and all new eigenvalues of $P_{g_i+1}$ are produced by the eigenvalues that were newly borne at generation $g_i$. Moreover, since both $\lambda_+$ and $\lambda_-$ are monotonously increasing functions, every new eigenvalue inherits the multiplicity of its parent. Hence, for $g \geq 2$, all eigenvalues (excluding zero eigenvalue) of $P_g$ are generated from $1, -1$ and $0$, with all the offspring eigenvalues of $1$ and $-1$ being nondegenerate. Therefore, all that is left is to determine the multiplicity of $0$, as well as the multiplicities of its descendants.

Let $r(M)$ denote the rank of matrix $M$. Then, the multiplicity of the zero eigenvalues for $P_g$ is

$$D_g^{mul}(0) = N_g - r(P_g).$$

We now evaluate $r(P_g)$. For the set of all nodes in $F_g$, let $\alpha$ denote the subset of nodes in $F_{g-1}$, and $\beta$ the subset of nodes newly produced at generation $g$. Then, $P_g$ can be written in a block form

$$P_g = \begin{pmatrix} P_{\alpha,\alpha} & P_{\alpha,\beta} \\ P_{\beta,\alpha} & P_{\beta,\beta} \end{pmatrix}.$$

As $P_{\beta,\beta}$ is the $(N_g - N_{g-1}) \times (N_g - N_{g-1})$ zero matrix, so $P_g$ is simplified as

$$P_g = \begin{pmatrix} P_{\alpha,\alpha} & P_{\alpha,\beta} \\ P_{\beta,\alpha} & 0 \end{pmatrix}.$$

Notice that $r(P_{\alpha,\beta}) = r(P_{\beta,\alpha})$. According to Eq. (11), we can know that any new node $i$ connecting to old node $o$ has the same value $P_g(o,i)$. So, $P_{\beta,\alpha}$ is a full column rank matrix. Then, $r(P_{\beta,\alpha}) = N_{g-1}$ and $r(P_g) = 2N_{g-1}$. So, we can obtain that $D_g^{mul}(0) = N_g - 2N_{g-1}$ and the degeneracy of eigenvalue $0$ of $P_g$ is

$$D_g^{mul}(0) = \begin{cases} 0, & \text{if } g = 0, \\ 3 \cdot 5^{g-1} - 1, & \text{if } g \geq 1. \end{cases}$$

We denote by $N_g^{seed}(0)$ the total number of eigenvalue $0$ and all of its descendants in $P_g$ $(g \geq 1)$,

$$\begin{aligned} N_g^{seed}(0) &= \sum_{i=1}^{g} (3 \cdot 5^{i-1}) 2^{g-i} \\ &= 5^g - 2^{g+1} + 1. \end{aligned}$$

Similarly, for either $-1$ or $1$, the total number of eigenvalues and their descendants in $P_g$ $(g \geq 1)$ is

$$N_g^{seed}(-1) = N_g^{seed}(1) = 2^g.$$

Summing up the number of eigenvalues obtained above gives

$$N_g^{seed}(0) + N_g^{seed}(-1) + N_g^{seed}(1) = 5^g + 1 = N_g,$$

which implies that we have found all the eigenvalues of matrix $P_g$ and thus the transition weight matrix $T_g$.

## 5 The analytic expression of eigentime identity and weighted counting of spanning trees

In this section, we apply the obtained eigenvalues and their multiplicities to determine eigentime identity for weight-dependent walk and the weighted counting of spanning trees in the weighted network $F_g$. Notice that $F_g$ has a treelike structure, the weighted counting of spanning trees is just be the product of weights of all edges in $F_g$. Thus, our aim for evaluating weighted counting of spanning trees is to verify that our computation for eigenvalues and their multiplicities is correct.

## 5.1 Eigentime identity

Suppose that $\Omega_g$ is the set of all the $N_g$ eigenvalues of matrix $L_g$ at generation $g$. $\Omega_0 = \{0, 2\}$, $0 \in \Omega_0$ generates two eigenvalues $0, \frac{1+4r}{1+2r} \in \Omega_1$. $2 \in \Omega_0$ generates two eigenvalues $2, \frac{1}{1+2r} \in \Omega_1$. We have $\Omega_1 = \{0, \Omega_1^{(1)}, \Omega_1^{(2)}, \Omega_1^{(3)}\} = \{0, \underbrace{\{1, 1\}}_{2}, \frac{1+4r}{1+2r}, \{2, \frac{1}{1+2r}\}\}$. Similarly, $\Omega_2 = \{0, \Omega_2^{(1)}, \Omega_2^{(2)}, \Omega_2^{(3)}\} = \{0, \underbrace{\{1, 1, \cdots, 1\}}_{14}, \frac{1+4r}{1+2r}, \Omega_2^{(3)}\}$, where $\Omega_2^{(3)}$ is generated by $1, \frac{1+4r}{1+2r}, 2$, and $\frac{1}{1+2r}$, i.e., $\Omega_2^{(3)}$ is initially generated by $1, \frac{1+4r}{1+2r}$ and $2$.

Generally, $\Omega_g$ can be classified into four subsets represented by $0$, $\Omega_g^{(1)}$, $\Omega_g^{(2)}$, and $\Omega_g^{(3)}$, respectively. That is, $\Omega_g = 0 \cup \Omega_g^{(1)} \cup \Omega_g^{(2)} \cup \Omega_g^{(3)}$, where $\Omega_g^{(1)}$ consists of eigenvalue 1 with multiplicity $3 \cdot 5^{g-1} - 1$, and $\Omega_g^{(2)}$ contains only eigenvalue $\frac{1+4r}{1+2r}$ with a single degeneracy, and $\Omega_g^{(3)}$ includes those eigenvalues generated by 1, 2 and $\frac{1+4r}{1+2r}$. For $\Omega_g^{(1)}$, we have $\sum_{i \in \Omega_g^{(1)}} \frac{1}{\sigma_i(g)} = 3 \cdot 5^{g-1} - 1$. For $\Omega_g^{(2)}$, we have $\sum_{i \in \Omega_g^{(2)}} \frac{1}{\sigma_i(g)} = \frac{1+2r}{1+4r}$. Hence, in order to determine $H_g$, we only need to evaluate $\sum_{i \in \Omega_g^{(3)}} \frac{1}{\sigma_i(g)}$.

From Eq. (14), we can easily obtain the following relation governing the eigenvalues of $L_g$ and $L_{g-1}$:

$$(1 + 2r)\sigma_i^2(g) - [1 + 4r + \sigma_i(g-1)]\sigma_i(g) + \sigma_i(g-1) = 0,$$

which means that each eigenvalue $\sigma_i(g-1)$ in $\Omega_{g-1}$ generates two eigenvalues, $\sigma_{i,1}(g)$ and $\sigma_{i,2}(g)$, belonging to $\Omega_g^{(3)}$. According to Vietas formulas, we have

$$\sigma_{i,1}(g) + \sigma_{i,2}(g) = \frac{1 + 4r + \sigma_i(g-1)}{1 + 2r},$$

and

$$\sigma_{i,1}(g)\sigma_{i,2}(g) = \frac{\sigma_i(g-1)}{1 + 2r}. \tag{16}$$

Then

$$\frac{1}{\sigma_{i,1}(g)} + \frac{1}{\sigma_{i,2}(g)} = \frac{1 + 4r}{\sigma_i(g-1)} + 1,$$

indicating that

$$\sum_{\sigma_i(g) \in \Omega_g^{(3)}} \frac{1}{\sigma_i(g)} = (1 + 4r) \sum_{\sigma_i(g-1) \in \Omega_{g-1}} \frac{1}{\sigma_i(g-1)} + 5^{g-1}.$$

Combining the above-obtained results, the recursive relation between $H_g$ and $H_{g-1}$ can be given

$$\begin{aligned} H_g &= \sum_{\sigma_i(g) \in \Omega_g^{(3)}} \frac{1}{\sigma_i(g)} + \sum_{\sigma_i(g) \in \Omega_g^{(2)}} \frac{1}{\sigma_i(g)} + \sum_{\sigma_i(g) \in \Omega_g^{(1)}} \frac{1}{\sigma_i(g)} \\ &= (1 + 4r)H_{g-1} + 5^{g-1} + 3 \cdot 5^{g-1} - 1 + \frac{1 + 2r}{1 + 4r} \\ &= (1 + 4r)H_{g-1} + 4 \cdot 5^{g-1} - \frac{2r}{1 + 4r}. \end{aligned}$$

Using the initial condition $H_0 = \frac{1}{2}$, we obtain

$$H_g = \begin{cases} 2(2g + 1) \cdot 5^{g-1} + \frac{1}{10}, & \text{if } r = 1, \\ \frac{5^g}{1-r} + \frac{1}{2(1+4r)} + \frac{2r^2+2r+1}{r-1}(1 + 4r)^{g-1}, & \\ & \text{if } r > 0 \text{ and } r \neq 1. \end{cases} \tag{17}$$

For very large networks ( i.e., $N_g \to \infty$), using Eqs. (2) and (17), the leading term of $H_g$ obey

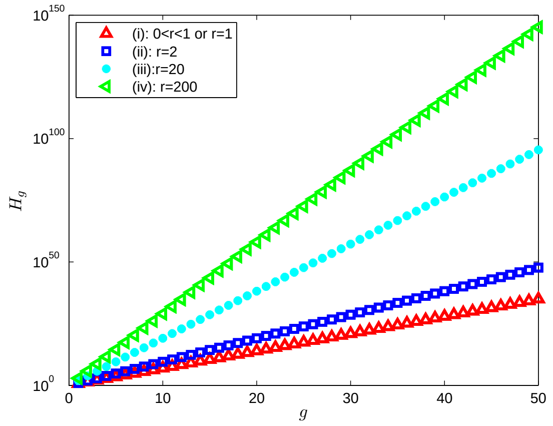

Figure 2: Eigentime identity $H_g$ versus $g$ is on a semilogarithmic scale.

$$H_g \sim \begin{cases} N_g, & \text{if } 0 < r \leq 1, \\ N_g^{\log_5(1+4r)}, & \text{if } r > 1. \end{cases}$$

We note that for eigentime identity studied in this paper, when $0 < r \leq 1$, $H_g$ grows linearly with the network size. When $r > 1$, $H_g$ grows superlinearly with the network size. For the range of $g \leq 50$, eigentime identity $H_g$ versus $g$ on a semilogarithmic scale is shown in Fig. 2.

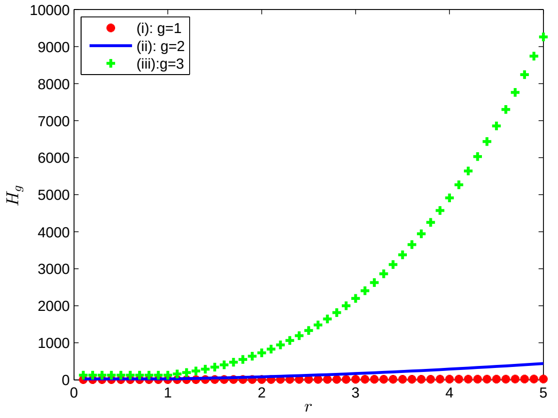

Figure 3: Eigentime identity $H_g$ versus $r$.

We note that for eigentime identity studied in this paper, when $g = 1, 2, 3$, eigentime identity $H_g$ versus $r$ is shown in Fig. 3.

### 5.2 Weighted counting of spanning trees

On the one hand, $\tau(F_g)$ equals the product of weight of all edges in $F_g$. Thus, $\tau(F_g)$ can be directly obtained by evaluating this product. Notice that $Q_g$ is the total weight of all edges in $F_g$, we have

$$Q_g = (1+4r)^g = \sum_{i=0}^{g} C_g^i 4^i r^i.$$

Thus, we can obtain

$$\tau(F_g) = r^{\sum_{i=0}^{g} iC_g^i 4^i}.$$

Let $Z_g = \sum_{i=0}^{g} iC_g^i 4^i$, $Z_g$ can be simplified as follows

$$\begin{aligned} Z_g &= \sum_{i=0}^{g} 4^i (iC_g^i) \\ &= \sum_{i=1}^{g} 4^i (gC_{g-1}^{i-1}) \\ &= 4g \sum_{i=0}^{g-1} 4^i C_{g-1}^i. \end{aligned}$$

Let $S_g = \sum_{i=0}^{g-1} 4^i C_{g-1}^i$, so $\tau(F_g)$ can be represented as

$$\tau(F_g) = r^{4gS_g}. \tag{18}$$

As

$$\begin{aligned} S_g &= \sum_{i=0}^{g-1} 4^i C_{g-1}^i \\ &= 4^0 C_{g-1}^0 + 4^1 C_{g-1}^1 + 4^2 C_{g-1}^2 \\ &\quad + \cdots + 4^{g-1} C_{g-1}^{g-1}, \end{aligned} \tag{19}$$

and

$$\begin{aligned} 4S_g &= 4^1 C_{g-1}^0 + 4^2 C_{g-1}^1 + \cdots \\ &\quad + 4^{g-1} C_{g-1}^{g-2} + 4^g C_{g-1}^{g-1}. \end{aligned} \tag{20}$$

From Eqs. (19) and (20), we have

$$\begin{aligned} 5S_g &= 4^0 C_{g-1}^0 + 4^1 (C_{g-1}^1 + C_{g-1}^0) + 4^2 (C_{g-1}^2 + C_{g-1}^1) \\ &\quad + \cdots + 4^{g-1} (C_{g-1}^{g-1} + C_{g-1}^{g-2}) + 4^g C_{g-1}^{g-1} \\ &= 4^0 C_g^0 + \sum_{i=1}^{g-1} 4^i (C_{g-1}^i + C_{g-1}^{i-1}) + 4^g C_g^g. \end{aligned} \tag{21}$$

Notice that

$$C_g^i = C_{g-1}^{i-1} + C_{g-1}^i.$$

Thus, Eq. (21) can be simplified as

$$\begin{aligned} 5S_g &= 4^0 C_g^0 + \sum_{i=1}^{g-1} 4^i C_g^i + 4^g C_g^g \\ &= \sum_{i=0}^{g} 4^i C_g^i \\ &= S_{g+1}. \end{aligned} \tag{22}$$

By Eqs. (18) and (22), we can obtain the following recursive relation

$$\ln\tau(F_{g+1}) = 5(1+\frac{1}{g})\ln\tau(F_g).$$

Considering $\ln\tau(F_1) = 4\ln r$, we have

$$\tau(F_g) = r^{4g\cdot 5^{g-1}}. \tag{23}$$

Let $R_g$ denote the product of the weight of new edges that generated at step $g$. As the weight of old edges existed in $F_{g-1}$ remains the same in $F_g$, we can obtain

$$\begin{aligned} R_g &= \frac{\tau(F_g)}{\tau(F_{g-1})} \\ &= r^{4(4g+1)\cdot 5^{g-2}}. \end{aligned} \tag{24}$$

On the other hand, in order to solve Eq. (5), we first compute the denominator of Eq. (5)

$$\sum_{i=1}^{N_g} s_i(g) = 2Q_g = 2(1+4r)^g. \tag{25}$$

For the two product terms $\prod_{i=1}^{N_g} s_i(g)$ and $\prod_{i=2}^{N_g}\sigma_i(g)$ in the numerator of Eq. (5), we use $\Delta_g$ and $\Lambda_g$ to represent them, respectively. According to the above-obtained results, the two quantities $\Delta_g$ and $\Lambda_g$ obey the following two recursive relations.

According to Eq. (24), we obtain

$$\begin{aligned} \Delta_g &= \prod_{i=1}^{N_{g-1}} s_i(g)\cdot \prod_{i=N_{g-1}+1}^{N_g} s_i(g) \\ &= \prod_{i=1}^{N_{g-1}} (1+2r)s_i(g-1)\cdot R_g \\ &= (1+2r)^{N_{g-1}}\cdot r^{4(4g+1)\cdot 5^{g-2}}\Delta_{g-1}, \end{aligned}$$

and from Eq. (16), we can obtain

$$\begin{aligned} \Lambda_g &= \prod_{i\in\Omega_g^{(1)}}\sigma_i(g)\cdot\prod_{i\in\Omega_g^{(2)}}\sigma_i(g)\cdot\prod_{i\in\Omega_g^{(3)}}\sigma_i(g) \\ &= \frac{1+4r}{1+2r}\cdot\prod_{i\in\Omega_g^{(3)}}\sigma_i(g) \\ &= \frac{1+4r}{1+2r}\cdot\frac{\Lambda_{g-1}}{(1+2r)^{N_{g-1}-1}} \\ &= \frac{1+4r}{(1+2r)^{N_{g-1}}}\Lambda_{g-1}. \end{aligned}$$

Thus,

$$\Delta_g\Lambda_g = (1+4r)r^{4(4g+1)\cdot 5^{g-2}}\Delta_{g-1}\Lambda_{g-1}. \tag{26}$$

Applying $\Delta_0 = 1$ and $\Lambda_0 = 2$, Eq. (26) is solved to give

$$\Delta_g\Lambda_g = 2(1+4r)^g\cdot r^{4g\cdot 5^{g-1}}. \tag{27}$$

Inserting the results in Eqs. (25) and (27) into Eq. (5) yields

$$\tau(F_g) = r^{4g\cdot 5^{g-1}},$$

which is consistent with Eq. (23), indicating the validity of our computation on the eigenvalues and their multiplicities for the transition weight matrix $T_g$ of $F_g$.

## 6 Conclusions

In conclusion, we have considered the eigentime identity for normalized Laplacian matrix of weighted network with two hub nodes. We have determined all the eigenvalues and their corresponding multiplicities for transition weight matrix of weighted network with two hub nodes. We can obtain the eigenvalues for normalized Laplacian matrix of weighted network with two hub nodes by using the obtained eigenvalues for transition weight matrix. Furthermore, we have used the obtained eigenvalues of normalized Laplacian matrix to derive an explicit expression about the eigentime identity for weight-dependent walk taking place on the network. Finally, we confirmed our results for the eigenvalues and their multiplicities for transition weight matrix via enumerating the weighted spanning trees, based on the connection between the eigenvalues and the weighted spanning trees.

## Acknowledgements

The authors express their gratitude to the referee for valuable comments. Research is supported by the Humanistic and Social Science Foundation from Ministry of Education of China (Grant 14YJAZH012) and National Natural Science Foundation of China (Nos. 11671172, 11501255).

## References

[1] Dai MF, Zhang DP, Li L, Topology properties of a weighted multi-local-world evolving network, *Can. J. Phys.* **93** 3 353-360 (2015).

[2] Wei DJ, Deng XY, Zhang XG, Deng Y, Sankaran Mahadevan, Identifying influential nodes in weighted networks based on evidence theory, *Physica A* **392** 2564ł2575 (2013).

[3] Wei DJ, Liu Q, Zhang HX, Hu Y, Deng Y, Sankaran Mahadevan, Box-covering algorithm for fractal dimension of weighted networks, *Sci. Rep.* **3** 3049 (2013).

[4] Aldous D, Fill J, Reversible Markov chains and random walks on graphs, see http://www.stat.berkeley.edu/ aldous/RWG/book.html (1999).

[5] Julaiti A, Wu B, Zhang ZZ, Eigenvalues of normalized Laplacian matrices of fractal trees and dendrimers: Analytical results and applications, *J. Chem. Phys.* **138** 204116 (2013).

[6] Sun YQ, Dai MF, Shao SX, The entire mean weighted first-passage time on infinite families of weighted tree networks, *Mod. Phys. Lett. B* **31** 7, 1750049 (2017).

[7] Brouwer AE, Haemers WH, Spectra of Graphs, Universitext. Springer New York. 2012.

[8] Zhang Z, Shan T, Chen G, Random walks on weighted networks, *Phys. Rev. E* **87** 012112 (2013).

[9] Xie PC, Zhang ZZ, Francesc Comellas, On the spectrum of the normalized Laplacian of iterated triangulations of graphs, *Appl. Math. Comput.* **273** 1123-1129 (2016).

[10] Zhang ZZ, Guo XY, Yi YH, Spectra of weighted scale-free networks, *Sci. Rep.* **5** 17469 (2015).

[11] Dai MF, Wang XQ, Sun YQ, Sun Y, Su WY, Eigentime identities for random walks on a family of treelike networks and polymer networks, *Physica A* **484** 132ł140 (2017).

[12] Dai MF, Li XY, Xi LF, Random walks on non-homogenous weighted Koch networks, *Chaos* **23** 3 033106-8 (2013).

[13] Sun Y, Dai MF, Sun YQ, Shao SX, Scaling of the average receiving time on a family of weighted hierarchical networks, *Fractals.* **24** 3 1650038 (2016).

[14] Dai MF, Sun YQ, Shao SX, Xi LF, Su WY, Modified box dimension and average weighted receiving time on the weighted fractal networks, *Sci. Rep.* **5** 18210(2015).

[15] Zhang ZZ, Zhou SG, Chen LC, Guan JH, Fang LJ, Zhang YC, Recursive weighted treelike networks, *Eur. Phys. J. B.* **59** 99-107 (2007).

[16] Ye DD, Dai MF, Sun YQ, Shao SX, Xie Q, Average receiving scaling of the weighted polygon Koch networks with the weight-dependent walk, *Physica A* **458** 1-8 (2016).

[17] Ye DD, Dai MF, Sun YQ, Su WY, Average weighted receiving time on the non-homogeneous double-weighted fractal networks, *Physica A* DOI: http://dx.doi.org/10.1016/j.physa.2017.01.013.

[18] Levene M, Loizou G, Kemenys constant and the random surfer, *Am. Math. Mon.* **109** 741-745 (2002).

[19] Ding QY, Sun WG, Chen FY, Network coherence in the web graphs, *Commun Nonlinear Sci Numer Simulat* **27** 228-236 (2015)

[20] Sun WG, Ding QY, Zhang JY, Chen FY, Coherence in a family of tree networks with an application of Laplacian spectrum, *Chaos* **24** 4 043112 (2014).

[21] Lin Y, Zhang Z, Random walks in weighted networks with a perfect trap: An application of Laplacian spectra, *Phys. Rev. E* **87** 062140 (2013).

[22] Tejedor V, Bnichou O, Voituriez R, Global mean first-passage times of random walks on complex networks, *Phys. Rev. E* **80** 065104 (2009).

[23] Chung FR, *Spectral Graph Theory* (American Mathematical Society, Providence, RI, 1997).

[24] Chen H, Zhang F, Resistance distance and the normalized Laplacian spectrum, *Discrete Appl. Math.* **155** 654-661 (2007).

[25] Chung F, Lu L, Vu V, Spectra of random graphs with given expected degrees, *Proc. Natl. Acad. Sci. USA.* **100** 6313-6318 (2003).

[26] Chang X, Xu H, S.-T. Yau, Spanning trees and random walks on weighted graphs, *Pacific J. Math.* **273** 241-255 (2015).

[27] Sun WG, Wang S, Zhang JY, Counting spanning trees in prism and anti-prism graphs, *J. APPL. ANAL. COMPUT.* **6** 1 65-75 (2016)