Markov Fundamental Tensor and Its Applications to Network Analysis

Golshan Golnari, Zhi-Li Zhang, and Daniel Boley

Department of Computer Science and Engineering, University of Minnesota

Abstract

We first present a comprehensive review of various random walk metrics used in the literature and express them in a consistent framework. We then introduce *fundamental tensor* – a generalization of the well-known fundamental matrix – and show that classical random walk metrics can be derived from it in a unified manner. We provide a collection of useful relations for random walk metrics that are useful and insightful for network studies. To demonstrate the usefulness and efficacy of the proposed fundamental tensor in network analysis, we present four important applications: 1) unification of network centrality measures, 2) characterization of (generalized) network articulation points, 3) identification of network most influential nodes, and 4) fast computation of network reachability after failures.

Keywords: Markov chain, random walk, fundamental tensor, network analysis, centrality measures, articulation points, influence maximization, network reachability

1. Introduction

Random walk and Markov chain theory, which are in close relationship, shown to be powerful tools in many fields from physics and chemistry to social sciences, economics, and computer science [1, 2, 3, 4, 5]. For network analysis, too, they have shown promises as effective tools [6, 7, 8, 9], where the hitting time, a well-known Markov metric, is used to measure the distance (or similarity) between different parts of a network and provide more insight to structural properties of the network. We believe though that the applicability of Markov chain theory to network analysis is more extensive and is not restricted to using the hitting time. Markov chain theory enables us to

provide more general solutions which cover the directed networks (digraphs) and is not tailored only to special case of undirected networks.

In this paper, we revisit the fundamental matrix in Markov chain theory [10], extend it to a more general form of tensor representation, which we call *fundamental tensor*, and use that to tackle four interesting network analysis applications. Fundamental tensor $\boldsymbol{F}_{smt}$ is defined [1] over three dimensions of source node s, middle (medial) node m, and target node t, which represents the expected number of times that the Markov chain visits node m when started from s and before hitting t for the first time. We show that the entire fundamental tensor can be computed by a single matrix inversion, which is much more efficient than computing the fundamental matrices for each target node separately ($O(n^3)$ vs. $O(n^4)$).

As the first application, we show that the fundamental tensor provides a unified way to compute the random walk distance (hitting time), random walk betweenness measure [11], random walk closeness measure [12], and random walk topological index (Kirchhoff index)[13] in a conceptual and insightful framework: hitting time distance as the aggregation of the fundamental tensor over the middle node dimension, betweenness as the aggregation over the source and target nodes, closeness as the aggregation over the source and middle node dimensions, and Kirchhoff index resulted as the aggregation over all the three dimensions. These four random walk measures are of well-known network analysis tools which have been vastly used in the literature [14, 15, 16, 17, 18, 19].

In the second application, we extend the definition of articulation points to the directed networks which has been originally defined for undirected networks, known as cut vertices as well. We show that the (normalized) fundamental tensor nicely functions as a look up table to find all the articulation points of a directed network. Founded on the notion of articulation points, we also propose a load balancing measure for the networks. Load balancing is important for network robustness against targeted attacks, where the balance in the loads help the network to show more resilience toward the failures. Through extensive experiments, we evaluate the load balancing in several specific-shaped networks and real-world networks.

[1] Note that the fundamental matrix is mostly denoted by N in Markov chain theory literature, but since N might reflect other meanings in computer science venues, we usually use F (or $\boldsymbol{F}$ to denote the tensor) in our papers.

The applicability and efficiency of the fundamental tensor in social networks is the subject of the third application in this paper. We show that the (normalized) fundamental tensor can be used in the field of social networks to infer the cascade and spread of a phenomena or an influence in a network and derive a formulation to find the most influential nodes for maximizing the influence spread over the network. While the original problem is NP-hard, we propose a greedy algorithm which yields a provably near-optimal solution. We show that this algorithm outperforms the state-of-the-art as well as the centrality/importance measure baselines in maximizing the influence spread in the network.

Since it is inefficient to use the regular reachability methods in large and dense networks with high volume of reachability queries whenever a failure occurs in the network, devising an efficient dynamic reachability method is necessary in such cases. As the fourth application, we present a dynamic reachability method in the form of a pre-computed oracle which is cable of answering to reachability queries efficiently ($O(1)$) both in the case of having failures or no failure in a general directed network. This pre-computed oracle is in fact the fundamental matrix computed for the extended network G^o and target o. The efficiency of the algorithm is resulted from the theorem that we prove on incremental computation of the fundamental tensor when a failure happens in the network. The storage requirement of this oracle is only $O(n^2)$. Note that in the last two applications, the directed network G does not need to be strongly connected, and our algorithms can be applied to any general network.

For the sake of completeness, we also provide a comprehensive review of the other Markov metrics, such as hitting time, absorption probability, and hitting cost, which is a very useful metric for weighted networks and was introduced in a more recent literature [19], but can be rarely found in Markov chain literature. In the review, we include Markov metrics' various definitions and formulations, and express them in a consistent form (matrix form, recursive form, and stochastic form). We also show that the fundamental tensor provides a basis for computing these Markov metrics in a unified manner. In addition, we review, gather, and derive many insightful relations for the Markov metrics.

The remainder of this paper is organized as follows. A preliminary on network terminology is presented in Section 2. In Section 3, we review and present various Markov metrics in a unified format. In Section 4, we gather and derive useful relations among the reviewed Markov metrics. Finally,

four applications are presented in Sections 5, 6, 7, and 8 to demonstrate the usefulness and efficacy of the fundamental tensor in network analysis.

2. Preliminaries

In general, a network can be abstractly modeled as a *weighted* and *directed* graph, denoted by $G = (\mathcal{V}, \mathcal{E}, W)$. Here $\mathcal{V}$ is the set of nodes in the network such as routers or switches in a communication network or users in a social network, and its size is assumed to be n throughout the paper $|\mathcal{V}| = n$; $\mathcal{E}$ is the set of (*directed*) edges representing the (physical or logical) connections between nodes (*e.g.*, a communication link from a node i to a node j) or entity relations (*e.g.*, follower-followee relation between two users). The *affinity* (or adjacency) matrix $A = [a_{ij}]$ is assumed to be nonnegative, *i.e.*, $a_{ij} \geq 0$, where $a_{ij} > 0$ if and only if edge e_{ij} exists, $e_{ij} \in \mathcal{E}$. The *weight* (or cost) matrix $W = [w_{ij}]$ represents the costs assigned to edges in a weighted network. Network G is called strongly connected if all nodes can be reachable from each other via at least one path. In this paper, we focus on strongly connected networks, unless stated otherwise.

A random walk in G is modeled by a discrete time Markov chain, where the nodes of G represent the states of the Markov chain. The target node in the network is modeled by an absorbing state at which the random walk arrives it cannot leave anymore. The Markov chain is fully described by its transition probability matrix: $P = D^{-1}A$, where D is the diagonal matrix of (out-)degrees, *i.e.*, $D = diag[d_i]$ and $d_i = \sum_j a_{ij}$. The d_i is often referred to as the (out-)degree of node i. Throughout the paper, the words "node" and "state", "network" and "Markov chain" are often used interchangeably depending on the context. If the network G is strongly connected, the associated Markov chain is irreducible and the stationary probabilities π are strictly positive according to Perron-Frobenius theorem [20]. For an undirected and connected G, the associated Markov chain is reversible and the stationary probabilities are a scalar multiple of node degrees: $\pi_i = \frac{d_i}{\sum_i d_i}$.

3. Definitions of Markov Metrics

We review various Markov metrics and present them using three unified forms: 1) matrix form (and in terms of the fundamental matrix), 2) recursive form, and 3) stochastic form. The matrix form is often the preferred form in this paper and we show how two other forms can be obtained from the

matrix form. The stochastic form, however, provides a more intuitive definition of random walk metrics. We also introduce *fundamental tensor* as a generalization of the fundamental matrix and show how it can be computed efficiently.

3.1. Fundamental Matrix

The *expected number of visits* counts the expected number of visits at a node, when a random walk starts from a source node and before a stopping criterion. The stopping criterion in random walk (or Markov) metrics is often "visiting a target node for the first time" which is referred to as hitting the target node. *Fundamental matrix* F is formed for a specific target node, where the entries are the expected number of visits at a medial node starting from a source node, for all such pairs. In the following, the fundamental matrix is defined formally using three different forms. [2]

- **Matrix form [22, 10]:** Let P be an $n \times n$ transition probability matrix for a strongly connected network G and node n be the target node. If the nodes are arranged in a way to assign the last index to the target node, transition probability matrix can be written in the form of $P = \begin{bmatrix} P_{11} & \boldsymbol{p}_{12} \\ \boldsymbol{p}'_{21} & p_{nn} \end{bmatrix}$ and the fundamental matrix is defined as follows:

$$F = (I - P_{11})^{-1}, \tag{1}$$

  where entry F_{sm} represents the expected number of visits of medial node m, starting from source node s, and before hitting (or absorption by) target node n [10]. Note that the target node can be any node t which would be specified in the notation by $F^{\{t\}}$ to clarify that it is

[2] Note that there exists another fundamental matrix $Z = (I - P + \mathbf{1}\boldsymbol{\pi}')^{-1}$ in literature as well which is defined for ergodic Markov chain and is shown [21] to be efficient for computing some Markov metrics such as hitting time. However, the fundamental matrix $F = (I - P_{11})^{-1}$, which is defined for absorbing Markov chain, is of special interest of the authors of this paper due to: 1- It is nicely interpretable in terms of random walk and is conceptually interesting as aggregation over the fundamental tensor dimensions would result to different Markov metrics (Section 5) and the articulation points of a network can be directly found from it (Section 6), 2- It can be used for both applications that are represented by ergodic chain (Sections 5 and 6) and absorbing chain (Sections 7 and 8), 3- It is easily generalizable to absorbing Markov chain with multiple absorbing states (Section 3.6) which we use to model network applications with multiple target nodes.

computed for target node t. This is discussed more in Markov metrics generalization to a set of targets (3.6).

Expanding F_{sm} as a geometric series, namely, $F_{sm} = [(I - P_{11})^{-1}]_{sm} = [I]_{sm} + [P_{11}]_{sm} + [P_{11}^2]_{sm} + ...$, it is easy to see the probabilistic interpretation of the *expected number of visits* as a summation over the number of steps required to visit node m.

- **Recursive form:** Each entry of the fundamental matrix, F_{sm}, can be recursively computed in terms of the entries of s's outgoing neighbors. Note that if $s = m$, F_{sm} is increased by 1 to account for $X_0 = m$ (the random walk starts at $s = m$, thus counting as the first visit at m).

$$F_{sm} = 1_{\{s=m\}} + \sum_{j \in \mathcal{N}_{out}(s)} p_{sj} F_{jm} \tag{2}$$

  It is easy to see the direct connection between the recursive form and the matrix form: from $F = I + P_{11}F$, we have $F = (I - P_{11})^{-1}$.

- **Stochastic form [23]:** Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with the transition probability matrix P, where X_k is the state of Markov chain in time step k. The indicator function $1_{\{X_k=m\}}$ is a Bernoulli random variable, equal to 1 if the state of Markov chain is m at time k, *i.e.* $X_k = m$, and 0 otherwise. The number of visits of node m, denoted by ν_m, can be written in terms of the indicator function: $\nu_m = \sum_{k=0}^{\infty} 1_{\{X_k=m\}}$. The stopping criteria is hitting target node t for the first time. In an irreducible chain, this event is guaranteed to occur in a finite time. Hence $k < \infty$. F_{sm} is defined as the expected value of ν_m starting from s.

$$\begin{aligned} F_{sm} &= \mathbb{E}_s(\nu_m) = \mathbb{E}_s \sum_{k=0}^{<\infty} 1_{\{X_k=m\}} = \sum_{k=0}^{<\infty} \mathbb{E}_s(1_{\{X_k=m\}}) \\ &= \sum_{k=0}^{<\infty} \mathbb{P}(X_k = m | X_0 = s, X_{<k} \neq t) = \sum_{k=0}^{<\infty} [P_{11}^k]_{sm}, \end{aligned} \tag{3}$$

  where the expression is simply the expanded version of the matrix form. Note that in order for F_{sm} to be finite (namely, the infinite summation converges), it is sufficient that node t be reachable from all other nodes

in network. In other words, the irreducibility of the entire network is not necessary.

3.2. Fundamental Tensor

We define the *fundamental tensor*, $\boldsymbol{F}$, as a generalization of the fundamental matrix $F^{\{t\}}$, which looks to be formed by stacking up the fundamental matrices constructed for each node t as the target node in a strongly connected network (Eq.(4)), but is in fact computed much more efficiently. In Theorem (1), we show that the whole fundamental tensor can be computed from Moore-Penrose pseudo-inverse of Laplacian matrix with only $O(n^3)$ of complexity and there is no need to compute the fundamental matrices for every target node which require $O(n^4)$ of computation in total.

$$\boldsymbol{F}_{smt} = \begin{cases} F^{\{t\}}_{sm} & \text{if } s, m \neq t \\ 0 & \text{if } s = t \text{ or } m = t \end{cases} \tag{4}$$

Fundamental tensor is presented in three dimensions of source node, medial (middle) node, and target node (Fig. (1)).

3.3. Hitting Time

The (expected) *hitting time* metric, also known as the first transit time, first passage time, and expected absorption time in the literature, counts the expected number of steps (or time) required to hit a target node for the first time when the random walk starts from a source node. Hitting time is frequently used in the literature as a form of (random walk) distance metric for network analysis. We formally present it in three different forms below.

- **Matrix form** [10]: Hitting time can be computed from the fundamental matrix (1) as follows:

$$\boldsymbol{h}^{\{t\}} = F^{\{t\}}\mathbf{1}, \tag{5}$$

where $\mathbf{1}$ is a vector of all ones and $\boldsymbol{h}^{\{t\}}$ is a vector of $H^{\{t\}}_s$ computed for all $s \in \mathcal{V} \setminus \{t\}$. $H^{\{t\}}_s$ represents the expected number of steps required to hit node t starting from s and is obtained from: $H^{\{t\}}_s = \sum_m F^{\{t\}}_{sm}$. The intuition behind this formulation is that enumerating the average number of nodes visited on the way from the source node to the target node yields the number of steps (distance) required to reach to the target node.

- **Recursive form** [21, 23, 19]: The *recursive* form of $H_s^{\{t\}}$ is the most well-known form presented in the literature for deriving the hitting time:

$$H_s^{\{t\}} = 1 + \sum_{m \in \mathcal{N}_{out}(s)} p_{sm} H_m^{\{t\}} \tag{6}$$

It is easy to see the direct connection between the recursive form and the matrix form: from $\boldsymbol{h} = \mathbf{1} + P_{11}\boldsymbol{h}$, we have $\boldsymbol{h} = (I - P_{11})^{-1}\mathbf{1}$.

- **Stochastic form** [23]: Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with the transition probability matrix P. The hitting time of the target node t is denoted by a random variable $\kappa_t : \Omega \to \{0, 1, 2, ...\} \cup \{\infty\}$ given by $\kappa_t = \inf\{\kappa \geq 0 : X_\kappa = t\}$, where by convention the infimum of the empty set $\emptyset$ is ∞. Assuming that the target node t is reachable from all the other nodes in the network, we have $\kappa_t < \infty$. The (expected) hitting time from s to t is then given by

$$\begin{aligned}
H_s^{\{t\}} &= \mathbb{E}_s[\kappa_t] = \sum_{k=1}^{<\infty} k\mathbb{P}(\kappa_t = k | X_0 = s) + \infty\mathbb{P}(\kappa_t = \infty | X_0 = s) \\
&= \sum_{k=1}^{<\infty} k\mathbb{P}(X_k = t | X_0 = s, X_{<k} \neq t) \\
&= \sum_{k=1}^{<\infty} k \sum_{m \neq t} \mathbb{P}(X_{k-1} = m | X_0 = s, X_{<k-1} \neq t) \cdot \mathbb{P}(X_k = t | X_{k-1} = m) \\
&= \sum_{k=1}^{<\infty} k \sum_{m \neq t} [P_{11}^{k-1}]_{sm} [\boldsymbol{p}_{12}]_m,
\end{aligned} \tag{7}$$

where $[P_{11}^0]_{sm} = 1$ for $m = s$ and it is 0 otherwise. The connection between the stochastic form and the matrix form can be found in the appendix.

3.3.1. Commute Time

The commute time between node i and node j is defined as the sum of the hitting time from i to j and the hitting time from j to i:

$$C_{ij} = H_i^{\{j\}} + H_j^{\{i\}} \tag{8}$$

Clearly, commute time is a symmetric quantity, *i.e.*, $C_{ij} = C_{ji}$. In contrast, hitting time is in general not symmetric, even when the network is undirected.

3.4. Hitting Cost

The (expected) *hitting cost*, also known as average first-passage cost in the literature, generalizes the (expected) hitting time by assigning a cost to each transition. Hitting cost from s to t, denoted by $\mathbb{H}_s^{\{t\}}$, is the average *cost* incurred by the random walk starting from node s to hit node t for the first time. The cost of transiting edge e_{ij} is given by w_{ij}. The hitting cost was first introduced by Fouss *et al.* [19] and given in a recursive form. In the following, we first provide a rigorous definition for hitting cost in the stochastic form, and then show how the matrix form and recursive form can be driven from this definition.

- **Stochastic form:** Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with the transition probability matrix P and cost matrix W. The hitting cost of the target node t is a random variable $\eta_t : \Omega \to \mathcal{C}$ which is defined by $\eta_t = \inf\{\eta \geq 0 : \exists k, X_k = t, \sum_{i=1}^{k} w_{X_{i-1}X_i} = \eta\}$. $\mathcal{C}$ is a countable set. If we view w_{ij} as the length of edge (link) e_{ij}, then the hitting cost η_t is the total length of steps that the random walk takes until it hits t for the first time. The expected value of η_t when the random walk starts at node s is given by

$$\mathbb{H}_s^{\{t\}} = \mathbb{E}_s[\eta_t] = \sum_{l \in \mathcal{C}} l\mathbb{P}(\eta_t = l | X_0 = s) \tag{9}$$

  For compactness, we delegate the more detailed derivation of the stochastic form and its connection with the matrix form to the appendix.

- **Matrix form:** Hitting cost can be computed from the following *closed form* formulation:

$$\mathbb{h}^{\{t\}} = F\boldsymbol{r}, \tag{10}$$

  where $\boldsymbol{r}$ is the vector of expected outgoing costs and $\mathbb{h}^{\{t\}}$ is a vector of $\mathbb{H}_s^{\{t\}}$ computed for all $s \in \mathcal{V} \setminus \{t\}$. The expected outgoing cost of node s is obtained from: $r_s = \sum_{m \in \mathcal{N}_{out}(s)} p_{sm} w_{sm}$. Note that the hitting time matrix H in Eq.(5) is a special case of the hitting cost matrix $\mathbb{H}$, obtained when $w_{ij} = 1$ for all e_{ij}.

- **Recursive form [19]:** The recursive computation of $\mathbb{H}_s^{\{t\}}$ is given as follows:

$$\mathbb{H}_s^{\{t\}} = r_s + \sum_{m \in \mathcal{N}_{out}(s)} p_{sm} \mathbb{H}_m^{\{t\}}. \tag{11}$$

It is easy to see the direct connection between the recursive form and the matrix form: from $\mathbb{h} = \boldsymbol{r} + P_{11}\mathbb{h}$, we have $\mathbb{h} = (I - P_{11})^{-1}\boldsymbol{r}$.

3.4.1. Commute Cost

Commute cost $\mathbb{C}_{ij}$ is defined as the expected cost required to hit j for the first time and get back to i. As in the case of commute time, commute cost is a symmetric metric and is given by

$$\mathbb{C}_{ij} = \mathbb{H}_i^{\{j\}} + \mathbb{H}_j^{\{i\}} \tag{12}$$

3.5. Absorption Probability

The absorption probability, also known as hitting probability in the literature, is the probability of hitting or getting absorbed by a target node (or any node in a set of target nodes) in a finite time [23]. For a single target node, this probability is trivially equal to 1 for all nodes in a strongly connected network. We therefore consider more than one target nodes in this paper.

Let indexes $n-1$ and n be assigned to two target nodes in a strongly connected network. We partition the transition probability matrix P as follows:

$$P = \begin{array}{c} \\ n-1 \\ n \end{array} \begin{array}{c} \begin{array}{ccc} n-1 & & n \end{array} \\ \begin{bmatrix} P_{11} & \boldsymbol{p}_{12} & \boldsymbol{p}_{13} \\ \boldsymbol{p}'_{21} & p_{n-1,n-1} & p_{n-1,n} \\ \boldsymbol{p}'_{31} & p_{n,n-1} & p_{n,n} \end{bmatrix} \end{array} \begin{array}{c} \\ n-1 \\ n \end{array} \tag{13}$$

where P_{11} is an $(n-2)\times(n-2)$ matrix, $\boldsymbol{p}_{12}$, $\boldsymbol{p}_{13}$, $\boldsymbol{p}_{21}$, and $\boldsymbol{p}_{31}$ are $(n-2)\times 1$ vectors, and the rest are scalars. The corresponding absorption probability can be expressed in three forms as follows:

- **Matrix form [10]:** The absorption probability matrix denoted by Q is a $(n-2)\times 2$ matrix whose columns represent the absorption probabilities to target $n-1$ and n respectively:

$$Q^{\{n-1,\overline{n}\}} = F\boldsymbol{p}_{12}, \tag{14}$$

$$Q^{\{\overline{n-1},n\}} = F\boldsymbol{p}_{13}, \tag{15}$$

where $F = (I - P_{11})^{-1}$. The notation $Q^{\{n-1,\overline{n}\}}$ emphasizes that target $n-1$ is hit sooner than target n, and $Q^{\{\overline{n-1},n\}}$ indicates hitting target n occurs sooner than target $n-1$. The formulation above states that to obtain the probability of getting absorbed (hit) by a given target when starting a random walk from a source node, we add up the absorption probabilities of starting from the neighbors of the source node, weighted by the number of times we expect to be in those neighboring nodes [10]. For a strongly connected network, these two probabilities are sum up to 1 for each starting node s, *i.e.,* $Q_s^{\{n-1,\overline{n}\}} + Q_s^{\{\overline{n-1},n\}} = 1$.

- **Recursive form [23]:** For each of the target nodes, the absorption probability starting from any source node can be found from the absorption probabilities starting from its neighbors:

$$Q_s^{\{\overline{n-1},n\}} = \sum_{m \in \mathcal{N}_{out}(s)} p_{sm} Q_m^{\{\overline{n-1},n\}}, \tag{16}$$

$$Q_s^{\{n-1,\overline{n}\}} = \sum_{m \in \mathcal{N}_{out}(s)} p_{sm} Q_m^{\{n-1,\overline{n}\}}, \tag{17}$$

  where $s, m \in \mathcal{V} \setminus \{n-1, n\}$. Note that the neighbors of a node can also be the target nodes. Thus, the right-hand side of the above equations is decomposed into two parts: $Q_s^{\{\overline{n-1},n\}} = p_{sn} + \sum_{m \neq n,n-1} p_{sm} Q_m^{\{\overline{n-1},n\}}$, and the same way for $Q_s^{\{n-1,\overline{n}\}}$. Now, it is easy to see how the recursive form is connected to the matrix form: from $Q^{\{\overline{n-1},n\}} = \boldsymbol{p}_{13} + P_{11} Q^{\{\overline{n-1},n\}}$, we have $Q^{\{\overline{n-1},n\}} = (I - P_{11})^{-1} \boldsymbol{p}_{13}$.

- **Stochastic form [23]:** Let $G = (X_k)_{k>0}$ be a discrete-time Markov chain with the transition matrix P. The hitting time of the target state n before $n-1$ is denoted by a random variable $\kappa_n : \Omega \to \{0, 1, 2, ...\} \cup \{\infty\}$ given by $\kappa_n = \inf\{\kappa \geq 0 : X_\kappa = n, X_{k<\kappa} \neq n, n-1\}$. Then the probability of ever hitting n is $\mathbb{P}(\kappa_n < \infty)$ [23]. This can be derived as

follows:

$$\begin{aligned} Q_s^{\{\overline{n-1},n\}} &= \sum_{k=1}^{<\infty} \mathbb{P}(\kappa_n = k | X_0 = s) \\ &= \sum_{k=1}^{<\infty} \mathbb{P}(X_k = n | X_0 = s, X_{<k} \neq n, n-1) \\ &= \sum_{k=1}^{<\infty} \sum_{m \neq n, n-1} \mathbb{P}(X_{k-1} = m | X_0 = s, X_{<k-1} \neq n, n-1) \cdot \\ &\quad \mathbb{P}(X_k = n | X_{k-1} = m) \\ &= \sum_{k=1}^{<\infty} \sum_m [P_{11}^{k-1}]_{sm} [\boldsymbol{p}_{13}]_m \\ &= \sum_m [(I - P_{11})^{-1}]_{sm} [\boldsymbol{p}_{13}]_m. \end{aligned} \tag{18}$$

The stochastic form for $Q_s^{\{n-1,\overline{n}\}}$ is derived in a similar vein.

3.6. Generalization: Markov Metrics for a Set of Targets

Let $\mathcal{A} = \{t_1, ..., t_{|\mathcal{A}|}\}$ be a set of target nodes. Then the transition probability matrix can be written in the following form:

$$P = \begin{bmatrix} P_{\mathcal{T}\mathcal{T}} & P_{\mathcal{T}\mathcal{A}} \\ P_{\mathcal{A}\mathcal{T}} & P_{\mathcal{A}\mathcal{A}} \end{bmatrix}, \tag{19}$$

where $\mathcal{T} = \mathcal{V} \setminus \mathcal{A}$ is the set of non-target nodes. Note that set of target nodes $\mathcal{A}$ can be modeled as the set of absorbing states in a Markov chain, and then $\mathcal{T} = \mathcal{V} \setminus \mathcal{A}$ is the set of transient (non-absorbing) nodes. Since hitting the target nodes is the stopping criterion for all the Markov metrics we have reviewed so far, it does not matter where the random walk can go afterwards and what the outgoing edges of the target nodes are. Therefore, there is no difference between $P = \begin{bmatrix} P_{\mathcal{T}\mathcal{T}} & P_{\mathcal{T}\mathcal{A}} \\ P_{\mathcal{A}\mathcal{T}} & P_{\mathcal{A}\mathcal{A}} \end{bmatrix}$ and $P = \begin{bmatrix} P_{\mathcal{T}\mathcal{T}} & P_{\mathcal{T}\mathcal{A}} \\ \mathbf{0} & I \end{bmatrix}$ for computing the Markov metrics.

For a given set of target nodes $\mathcal{A}$, the fundamental matrix $F^{\mathcal{A}}$ is obtained using the following relation:

$$F^{\mathcal{A}} = I + \sum_{k=1}^{<\infty} P_{\mathcal{T}\mathcal{T}}^k = (I - P_{\mathcal{T}\mathcal{T}})^{-1}, \tag{20}$$

which is a general form of the fundamental matrix defined for a single target (Eq.(1)). Entry $F^{\mathcal{A}}_{sm}$ represents the expected number of visits to m before hitting *any* of the target nodes in $\mathcal{A}$ when starting a random walk from s.

A hitting time for $\mathcal{A}$ is defined as the expected number of steps to hit the set for the first time which can occur by hitting *any* of the target nodes in this set. The vector of hitting times with respect to a target set $\mathcal{T}$ can be computed using

$$\boldsymbol{h}^{\mathcal{A}} = F^{\mathcal{A}}\mathbf{1} \tag{21}$$

If there exists a matrix of costs W defined for the network, the hitting cost for target set $\mathcal{A}$ is given below

$$\boldsymbol{\hbar}^{\mathcal{A}} = F^{\mathcal{A}}\boldsymbol{r}, \tag{22}$$

where $\boldsymbol{r}$ is a vector of expected outgoing cost r_s's: $r_s = \sum_{m\in\mathcal{N}_{out}(s)} p_{sm}w_{sm}$.

The absorption probability of target set $\mathcal{A}$ is a $|\mathcal{T}| \times |\mathcal{A}|$ matrix whose columns represents the absorption probability for each target node if it gets hit sooner than the other target nodes:

$$Q^{\mathcal{A}} = F^{\mathcal{A}}P_{\mathcal{T}\mathcal{A}}, \tag{23}$$

where $Q^{\mathcal{A}}$ is a row-stochastic matrix for a strongly connected network.

We remark that if the network is not strongly connected (thus the corresponding Markov chain is not irreducible), $I - P_{\mathcal{T}\mathcal{T}}$ may not be non-singular for every set of $\mathcal{A}$. Hence $F^{\mathcal{A}}$ may not exist. The necessary and sufficient condition for the existence of $F^{\mathcal{A}}$ is that target set $\mathcal{A}$ includes *at least one node from each recurrent equivalence class* in the network. The recurrent equivalence class is the minimal set of nodes that have no outgoing edge to nodes outside the set. Once a random walk reaches a node in a recurrent equivalence class, it can no longer get out of that set. A recurrent equivalence class can be as small as one single node, which is called an absorbing node.

4. Useful Relations for Markov Metrics

In this section, we first establish several important theorems, and then gather and derive a number of useful relations among the Markov metrics. We start by relating the fundamental tensor with the Laplacian matrices of a general network. For an undirected network or graph G, the graph Laplacian $L^u = D - A$ (where A is the adjacent matrix of G and $D = diag[d_i]$ is the

diagonal matrix of node degrees) and its normalized version $\tilde{L}^u = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ have been widely studied and found many applications (see, *e.g.*, [24] and the references therein). In particular, it is well-known that commute times are closely related to the Penrose-Moore pseudo-inverse of L^u (and a variant of Eq.(24) also holds for $\tilde{L}^u$):

$$C_{ij} = L^{u,+}_{ii} + L^{u,+}_{jj} - L^{u,+}_{ij} - L^{u,+}_{ji}. \tag{24}$$

Li and Zhang [25, 26, 27] were first to introduce the (correct) generalization of graph Laplacians for directed networks/graphs (*digraphs*) using the stationary distribution $\{\pi_i\}$ of the transition matrix $P = D^{-1}A$ for the associated (random walk) Markov chain defined on a directed network G. For a strongly connected network G, its *normalized digraph Laplacian* is defined as $\tilde{L} = \Pi^{\frac{1}{2}}(I - P)\Pi^{-\frac{1}{2}}$, where $\Pi = diag[\pi_i]$ is the diagonal matrix of stationary probabilities. Li and Zhang proved that the hitting time and commute time can be computed from the Moore-Penrose pseudo-inverse $\tilde{L}^+$ of $\tilde{L}$ using the following relations:

$$H_i^{\{j\}} = \frac{\tilde{L}^+_{jj}}{\pi_j} - \frac{\tilde{L}^+_{ij}}{\sqrt{\pi_i \pi_j}} \tag{25}$$

and

$$C_{ij} = H_i^{\{j\}} + H_j^{\{i\}} = \frac{\tilde{L}^+_{ii}}{\pi_i} + \frac{\tilde{L}^+_{jj}}{\pi_j} - \frac{\tilde{L}^+_{ij}}{\sqrt{\pi_i \pi_j}} - \frac{\tilde{L}^+_{ji}}{\sqrt{\pi_i \pi_j}}. \tag{26}$$

We define the (*unnormalized*) *digraph Laplacian* for a general (directed or undirected) network G as $L = \Pi(I - P)$ and the *random walk Laplacian* as $L^p = I - P$. Clearly, $\tilde{L} = \Pi^{-\frac{1}{2}} L \Pi^{-\frac{1}{2}} = \Pi^{\frac{1}{2}} L^p \Pi^{-\frac{1}{2}}$. Note that for a (connected) undirected graph, as $\pi_i = \frac{d_i}{vol(G)}$ where $vol(G) = \sum_j d_j$, we see that the classical graph Laplacian $L^u = D - A = vol(G)L$. Any results which hold for L also hold for $L^u = D - A$ with a scalar multiple. In the following we relate the fundamental tensor to the digraph and random walk Laplacians L and L^p, and use this relation to establish similar expressions for computing hitting and commute times using L, analogous to Eqs.(25) and (26).

Lemma 1 ([28]). *Let* $\begin{bmatrix} L_{11} & \boldsymbol{l}_{12} \\ \boldsymbol{l}'_{21} & l_{nn} \end{bmatrix}$ *be an* $n \times n$ *irreducible matrix such that* $nullity(L) = 1$. *Let* $M = L^+$ *be the Moore-Penrose pseudo-inverse of* L *partitioned similarly and* $(\mathbf{u}', 1)L = 0$, $L(\mathbf{v}; 1) = 0$, *where* $\mathbf{u}$ *and* $\mathbf{v}$ *are* $(n-1)$*-dim column vectors,* $\mathbf{u}'$ *is the transpose of the column vector* $\mathbf{u}$ *(*$(\mathbf{u}', 1)$ *is a*

n-dim row vector and $(\mathbf{v}; 1)$ *is a n-dim column vector,* a la *MATLAB). Then the inverse of the* $(n-1) \times (n-1)$ *matrix* L_{11} *exists and is given by:*

$$L_{11}^{-1} = (I + \mathbf{v}\mathbf{v}')M_{11}(I + \mathbf{u}\mathbf{u}'), \tag{27}$$

where I *denotes the* $(n-1) \times (n-1)$ *identity matrix.*

Note that node n in the above lemma can be substituted by any other node (index).

Theorem 1. *The fundamental tensor can be computed from the Moore-Penrose pseudo-inverse of the* digraph Laplacian matrix $L = \Pi(I - P)$ *as well as the* random walk *Laplacian matrix* $L^p = I - P$ *as follows, which results to* $O(n^3)$ *time complexity:*

$$\boldsymbol{F}_{smt} = (L^+_{sm} - L^+_{tm} + L^+_{tt} - L^+_{st})\pi_m, \tag{28}$$

$$\boldsymbol{F}_{smt} = L^{p+}_{sm} - L^{p+}_{tm} + \frac{\pi_m}{\pi_t}L^{p+}_{tt} - \frac{\pi_m}{\pi_t}L^{p+}_{st}, \tag{29}$$

where π_i *is the stationary probability of node* i *and* Π *is a diagonal matrix whose* i*-th diagonal entry is equal to* π_i*.*

Proof. Note that $F = (I - P_{11})^{-1} = L_{11}^{-1}$ as in Lemma 1. The above equations follow from Lemma 1 with $\boldsymbol{v} = \boldsymbol{u} = \mathbf{1}$. The nullity of matrix $L^p = I - P$ for a strongly connected network is 1. Using Eq.(28) or (29), all n^3 entries of the fundamental tensor $\boldsymbol{F}$ can be computed from L^+ in constant time each. □

Corollary 1.

$$\sum_{s,t} \boldsymbol{F}_{smt} = c\pi_m, \tag{30}$$

where c *is a constant independent of* m*.*

Proof.

$$\begin{aligned} \sum_{s,t} \boldsymbol{F}_{smt} &= \sum_{s,t}(L^+_{sm} - L^+_{tm} - L^+_{st} + L^+_{tt})\pi_m & (31)\\ &= 0 - 0 - 0 + (n\sum_t L^+_{tt})\pi_m & (32)\\ &= c\pi_m, & (33) \end{aligned}$$

where the second equality follows from the fact that the column sum of $L^+ = (\Pi(I - P))^+$ is zero. Later in Section 5, we will show that $c = |\mathcal{E}|K$, where K is the Kirchhoff index of a network. □

Corollary 2. *Hitting time and commute time can also be expressed in terms of entries in the digraph Laplacian matrix $L = \Pi(I - P)$ [27]:*

$$H_i^{\{j\}} = \sum_m (L_{im}^+ - L_{jm}^+)\pi_m + L_{jj}^+ - L_{ij}^+, \tag{34}$$

$$C_{ij} = L_{ii}^+ + L_{jj}^+ - L_{ij}^+ - L_{ji}^+, \tag{35}$$

Proof. Use Eq.(5) and (28). □

Note that we can also write the metrics in terms of the random walk Laplacian matrix L^p by a simple substitution: $L_{im}^+ - L_{ij}^+ = \frac{L^{p+}_{im}}{\pi_m} - \frac{L^{p+}_{ij}}{\pi_j}$.

Corollary 3. *Hitting cost $\mathbb{H}$ and commute cost $\mathbb{C}$ can be expressed in terms of the digraph Laplacian matrix $L = \Pi(I - P)$:*

$$\mathbb{H}_{ij} = \sum_m (L_{im}^+ - L_{jm}^+ + L_{jj}^+ - L_{ij}^+) g_m, \tag{36}$$

$$\mathbb{C}_{ij} = (L_{im}^+ - L_{jm}^+ + L_{jj}^+ - L_{ij}^+) \sum_m g_m, \tag{37}$$

where $g_m = r_m \pi_m$ and $r_m = \sum_{k \in \mathcal{N}_{out}(m)} p_{mk} w_{mk}$.

Proof. Use Eq.(10) and (28). From Eq.(35) and (37), it is also interesting to note that commute cost is a multiple scalar of commute time. □

Lemma 2 ([28])**.** *Let C be an $n \times n$ non-singular matrix and suppose $A = C - \boldsymbol{u}\boldsymbol{v}'$ is singular. Then the Moore-Penrose pseudo-inverse of A is given as:*

$$A^+ = (I - \frac{\boldsymbol{x}\boldsymbol{x}'}{\boldsymbol{x}'\boldsymbol{x}}) C^{-1} (I - \frac{\boldsymbol{y}\boldsymbol{y}'}{\boldsymbol{y}'\boldsymbol{y}}), \tag{38}$$

where $\boldsymbol{x} = C^{-1}\boldsymbol{u}$, $\boldsymbol{y}' = \boldsymbol{v}' C^{-1}$.

Theorem 2. *For an ergodic Markov chain, the Moore-Penrose pseudo-inverse of random-walk Laplacian L^{p+} can be computed from fundamental matrix $\boldsymbol{Z} = (\boldsymbol{I} - \boldsymbol{P} + \boldsymbol{1}\boldsymbol{\pi}')^{-1}$ [21] as follows:*

$$L^{p+} = (I - \frac{Z\mathbf{1}\mathbf{1}'Z'}{\mathbf{1}'Z'Z\mathbf{1}}) Z (I - \frac{Z'\boldsymbol{\pi}\boldsymbol{\pi}'Z}{\boldsymbol{\pi}'ZZ'\boldsymbol{\pi}}), \tag{39}$$

where $\mathbf{1}$ is a vector of all 1's and $\boldsymbol{\pi}$ denotes the vector of stationary probabilities.

Proof. The theorem is a direct result of applying Lemma 2. □

Theorem 2 along with Theorem 1 reveal the relation between the fundamental matrices F and Z. They also show that the fundamental tensor F can be computed by a single matrix inverse, can it be either a Moore-Penrose pseudo-inverse or a regular matrix inverse, as L^{p+} in Eq. (29) can be computed by either operating the pseudo-inverse on L^p or using Eq. (39). Discussion on computing Markov metrics via the group inverse can be found in [29, 30].

Theorem 3 (Incremental Computation of the Fundamental Matrix)**.** *The fundamental matrix for target set $\mathcal{S}_1 \cup \mathcal{S}_2$ can be computed from the fundamental matrix for target set $\mathcal{S}_1$ as follows,*

$$F_{im}^{\mathcal{S}_1 \cup \mathcal{S}_2} = F_{im}^{\mathcal{S}_1} - F_{i\mathcal{S}_2}^{\mathcal{S}_1} [F_{\mathcal{S}_2\mathcal{S}_2}^{\mathcal{S}_1}]^{-1} F_{\mathcal{S}_2 m}^{\mathcal{S}_1}, \tag{40}$$

where $F_{i\mathcal{S}_2}^{\mathcal{S}_1}$ denotes the row corresponding to node i and the columns corresponding to set $\mathcal{S}_2$ of the fundamental matrix $F^{\mathcal{S}_1}$, and the (sub-)matrices $F_{\mathcal{S}_2\mathcal{S}_2}^{\mathcal{S}_1}$ and $F_{\mathcal{S}_2 m}^{\mathcal{S}_1}$ are similarly defined.

Proof. Consider the matrix $M = I - P_{\mathcal{T}\mathcal{T}}$, where the absorbing set is $\mathcal{A} = \mathcal{S}_1$ and the transient set $\mathcal{T} = V \setminus \mathcal{S}_1$. The inverse of M yields the fundamental matrix $F^{\mathcal{S}_1}$, and the inverse of its sub-matrix obtained from removing rows and columns corresponding to set $\mathcal{S}_2$ yields the fundamental matrix $F^{\mathcal{S}_1 \cup \mathcal{S}_2}$. Using the following equations from the Schur complement, we see that the inverse of a sub-matrix can be derived from that of the original matrix.

If A is invertible, we can factor the matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ as follows

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ 0 & D - CA^{-1}B \end{bmatrix} \tag{41}$$

Inverting both sides of the equation yields

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & -A^{-1}BS^{-1} \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \tag{42}$$

$$= \begin{bmatrix} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}CA^{-1} & S^{-1} \end{bmatrix} \tag{43}$$

$$= \begin{bmatrix} X & Y \\ Z & W \end{bmatrix}, \tag{44}$$

where $S = D - CA^{-1}B$. Therefore, A^{-1} can be computed from $A^{-1} = X - YW^{-1}Z$. □

Corollary 4. *The simplified form of Theorem 3 for a single target is given by*

$$F_{im}^{\{j,k\}} = F_{im}^{\{j\}} - \frac{F_{ik}^{\{j\}} F_{k,m}^{\{j\}}}{F_{k,k}^{\{j\}}} \tag{45}$$

Lemma 3.

$$P_{\mathcal{T}\mathcal{T}} F^{\mathcal{A}} = F^{\mathcal{A}} P_{\mathcal{T}\mathcal{T}} = F^{\mathcal{A}}, \tag{46}$$

where $\mathcal{T} \cup \mathcal{A} = \mathcal{V}$

Proof. It follows easily from Eq.(1). □

Corollary 5 (Another Recursive Form for the Fundamental Matrix)**.**

$$F_{im}^{\{j\}} = \begin{cases} \sum_{k \in \mathcal{N}_{in}(m)} F_{ik}^{\{j\}} p_{km} & \text{if } i \neq m \\ 1 + \sum_{k \in \mathcal{N}_{in}(m)} F_{ik}^{\{j\}} p_{km} & \text{if } i = m \end{cases} \tag{47}$$

Proof. It is a special case of Lemma 3. Note that the recursive relation in Eq.(2) is in terms of s's outgoing neighbors, while this one is in terms of incoming neighbors of m. □

Theorem 4 (Absorption Probability & Normalized Fundamental Matrix)**.** *The absorption probability of a target node j in an absorbing set $\mathcal{A} = \{j\} \cup \mathcal{S}$ can be written in terms of the normalized fundamental matrix $F^{\mathcal{S}}$, where the columns are normalized by the diagonal entries:*

$$Q_{ij}^{\mathcal{A}} = \frac{F_{ij}^{\mathcal{S}}}{F_{jj}^{\mathcal{S}}} \tag{48}$$

Proof.

$$\begin{aligned} Q_{ij}^{\mathcal{A}} &= [F^{\mathcal{A}} P_{\mathcal{TA}}]_{ij} \\ &= \sum_{m\in\mathcal{T}} F_{im}^{\mathcal{A}} p_{mj} \\ &= \sum_{m\in\mathcal{T}} (F_{im}^{\mathcal{S}} - \frac{F_{ij}^{\mathcal{S}} F_{jm}^{\mathcal{S}}}{F_{jj}^{\mathcal{S}}}) p_{mj} \\ &= \sum_{m\in\mathcal{T}} F_{im}^{\mathcal{S}} p_{mj} - \frac{F_{ij}^{\mathcal{S}}}{F_{jj}^{\mathcal{S}}} \sum_{m\in\mathcal{T}} F_{jm}^{\mathcal{S}} p_{mj} \\ &= F_{ij}^{\mathcal{S}} - \frac{F_{ij}^{\mathcal{S}}}{F_{jj}^{\mathcal{S}}} (F_{jj}^{\mathcal{S}} - 1) \\ &= \frac{F_{ij}^{\mathcal{S}}}{F_{jj}^{\mathcal{S}}}, \end{aligned} \tag{49}$$

where the third and fifth equalities follow directly from of Theorem 3 and Lemma 3, respectively. □

We are now in a position to gather and derive a number of useful relations among the random walk metrics.

Relation 1 (Complementary relation of absorption probabilities)**.**

$$Q_{ij}^{\mathcal{A}} = 1 - \sum_{k\in\mathcal{A}\setminus\{j\}} Q_{ik}^{\mathcal{A}}, \tag{50}$$

where $i \in \mathcal{T}$ and $j \in \mathcal{A}$.

Proof. Based on the definition of Q and the assumption that all the nodes in $\mathcal{T}$ are transient, the probability that a random walk eventually ends up in set $\mathcal{A}$ is 1. □

Relation 2 (Relations between the fundamental matrix and commute time)**.**

$$(1) \quad F_{ii}^{\{j\}} = \pi_i C_{ij} \tag{51}$$

$$(2) \quad \frac{F_{im}^{\{j\}}}{\pi_m} + \frac{F_{mi}^{\{j\}}}{\pi_i} = C_{ij} + C_{jm} - C_{im} \tag{52}$$

$$(3) \quad \frac{F_{im}^{\{j\}}}{\pi_m} + \frac{F_{ij}^{\{m\}}}{\pi_j} = C_{jm} \tag{53}$$

$$(4) \quad F_{im}^{\{j\}} + F_{jm}^{\{i\}} = \pi_m C_{ij} \tag{54}$$

Proof. Use (28) and (35). □

Relation 3 (The hitting time detour overhead in terms of other metrics)**.**

$$(1) \quad H_i^{\{j\}} + H_j^{\{m\}} - H_i^{\{m\}} = \frac{F_{im}^{\{j\}}}{\pi_m} \tag{55}$$

$$(2) \quad H_i^{\{j\}} + H_j^{\{m\}} - H_i^{\{m\}} = Q_i^{\{m,\bar{j}\}} C_{mj} \tag{56}$$

Proof. For the first equation use (28) and (34), and for the second one use the previous equation along with (4) and (51). □

Relation 4 (The hitting time for two target nodes in terms of hitting time for a single target)**.**

$$H_i^{\{j,k\}} = H_i^{\{k\}} - Q_i^{\{j,\bar{k}\}} H_j^{\{k\}} = H_i^{\{j\}} - Q_i^{\{k,\bar{j}\}} H_k^{\{j\}}, \tag{57}$$

which can also be reformulated as: $H_i^{\{j\}} = H_i^{\{j,k\}} + Q_i^{\{k,\bar{j}\}} H_k^{\{j\}}$.

Proof. Aggregate two sides of Eq.(3) over m and substitute Eq.(4) in it. □

Relation 5 (Inequalities for hitting time)**.**

$$(1) \quad H_i^{\{m\}} + H_m^{\{j\}} \geq H_i^{\{j\}} \qquad \textbf{\textit{(triangular inequality)}} \tag{58}$$

$$(2) \quad H_i^{\{j\}} \geq H_i^{\{j,m\}} \tag{59}$$

$$(3) \quad H_i^{\{m\}} + H_m^{\{j,k\}} \geq H_i^{\{j,k\}} \tag{60}$$

Proof. For the first inequality, use (34) and (64). For the second inequality, use the aggregated form of Eq.(3) over m and the fact that the entries of F are non-negative. The third inequality is a generalization of the first one. □

Relation 6 (Inequalities for the fundamental matrix)**.**

$$(1) \quad F_{im}^{\{j\}} F_{kk}^{\{j\}} \geq F_{ik}^{\{j\}} F_{km}^{\{j\}} \tag{61}$$

$$(2) \quad F_{kk}^{\{j\}} \geq F_{ik}^{\{j\}} \tag{62}$$

Proof. For the first inequality, use Eq.(3) and the fact that F is non-negative. The second one can be derived from Eqs.(51), (55) and (58). Note that these two inequalities hold for any absorbing set $\mathcal{A}$, hence we drop the superscripts. □

Relation 7 (Inequality for absorption probabilities)**.**

$$Q_i^{\{m,\bar{j}\}} \geq Q_i^{\{k,\bar{j}\}} Q_k^{\{m,\bar{j}\}} \tag{63}$$

Proof. Use (4) and (61). □

Relation 8 (Inequality for the digraph Laplacian matrix)**.**

$$L_{im}^{+} + L_{kk}^{+} \geq L_{ik}^{+} + L_{km}^{+} \tag{64}$$

Proof. Use (28) and the fact that F's entries are always non-negative. □

Relation 9 (Relations for undirected networks (reversible Markov chain))**.**

$$(1) \quad \frac{F_{im}^{\{S\}}}{\pi_m} = \frac{F_{mi}^{\{S\}}}{\pi_i} \tag{65}$$

$$(2) \quad Q_i^{\{m,\bar{j}\}} C_m^{\{j\}} = Q_m^{\{i,\bar{j}\}} C_i^{\{j\}} \tag{66}$$

$$(3) \quad H_i^{\{m\}} + H_m^{\{j\}} + H_j^{\{i\}} = H_m^{\{i\}} + H_j^{\{m\}} + H_i^{\{j\}} \tag{67}$$

Proof. The first equation follows from Eq.(28) and the fact that L^{+} is symmetric for undirected networks. The second equation can be derived by using Eqs. (4), (28), (35) and the fact that L^{+} is symmetric. The third equation follows from Eq.(34) and L^{+} being symmetric. □

5. Unifying Random-Walk Distance, Centrality, and Topological Measures

Many network measures have been proposed in the literature for network analysis purposes [31], such as distance metrics for measuring the similarity (or diversity) between nodes or entities of a network, centrality measures to assess a node's involvement or importance in the connectivity or communication between network entities, and topological indices to measure the structural stability of networks. In this section, we review some of these network measures proposed in the literature, and show that these measures can be unified in terms of the fundamental tensor, which provides a coherent framework for computing them and understanding the relations among them.

Statement 1. *Fundamental tensor* $\boldsymbol{F}$ *unifies various network random-walk measures via summation along one or more dimensions shown in Figure 1.*

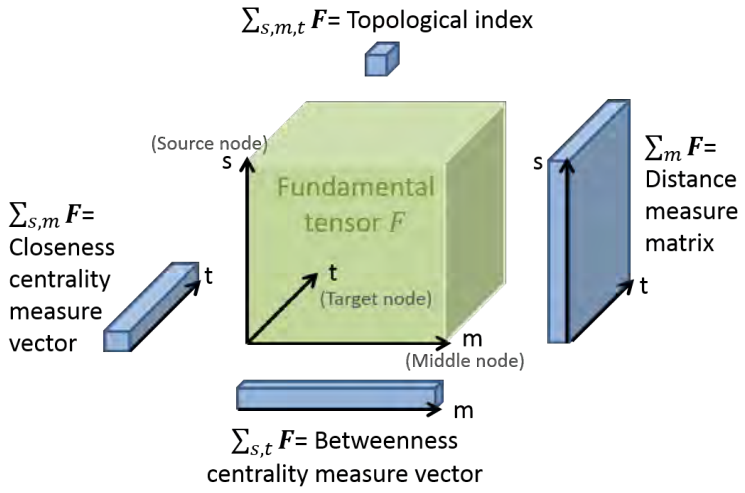


Figure 1: Markov fundamental tensor and unifying framework for computing the random walk distance, betweenness measure, closeness measure, and topological index

5.1. Random-walk distance measure

The hitting time metric has been used extensively in different application domains, such as a distance (or dissimilarity) measure for clustering and classification purposes [9]. Note that this distance metric satisfies two out of three conditions for the general distance metric: It is positive when two ends are different and zero when two ends are identical. As noted earlier, the hitting time metric is in general not symmetric, but it satisfies the triangle inequality. In Section 3, we have shown that hitting time can be computed from the fundamental tensor by summing over m's (the middle node dimension, see Figure 1).

$$Distance_{\mathrm{rw}}(s,t) = H_s^{\{t\}} = \sum_m \boldsymbol{F}_{smt}. \tag{68}$$

With a cost matrix W, the hitting cost distance (10) is obtained by the weighted sum over the medial node dimension of the fundamental tensor: $\mathbb{H}_s^{\{t\}} = \sum_m \boldsymbol{F}_{smt} b_m$, where $b_m = \sum_i w_{mi} p_{mi}$ is the expected outgoing cost of node m.

5.2. Random-walk centrality measures

Network centrality measures can be broadly categorized into two main types [31]: i) distance-based and ii) volume-based. The *closeness* centrality is an example of the distance-based measures, whereas the *betweenness* centrality is an example of volume-based measures.

- Random-walk closeness measure: Closeness centrality ranks nodes in a network based on their total distance from other nodes of the network. This measure reflects how easily/closely the node is accessible/reachable from the other parts of the network, and in a nutshell how "central" the node is located within a network. The classical closeness centrality metric is defined using the shortest path distances. Noh and Rieger [12] introduces the *random walk closeness centrality*, which is defined using the hitting time distance: A node is considered to have a high centrality value if and only if its total hitting time distances from other nodes in the network is small. This closeness centrality measure can be easily expressed in terms of the random walk fundamental tensor:

$$Closeness_{\mathrm{rw}}(t) = \sum_{s} H_s^{\{t\}} = \sum_{s,m} \boldsymbol{F}_{smt}, \tag{69}$$

  or in the reciprocal form to imply lower importance with small closeness value: $Closeness_{\mathrm{rw}}(t) = \frac{|\mathcal{V}|}{\sum_{s,m} \boldsymbol{F}_{smt}}$.

- Random-walk betweenness measure: Betweenness measure quantifies the number of times a node acts as a "bridge" along the paths between different parts of the network. The larger the number of paths crossing that node, the more central the node is. As a special case, the node degree, $deg(m)$, can be viewed as a betweenness centrality measure in an undirected network. Clearly, it captures how many paths of length 1 going through node m (or many 1-hop neighbors it has) [31]. It is also proportional to the total number of (random) walks passing through node m from any source to any target in the network. This follows from the following more general statement. For a general (strongly connected) network, we define the *random walk betweenness* of node m as follows and show that it is proportional to π_m, the stationary

probability of node m:

$$\begin{aligned} Betweenness_{\mathrm{rw}}(m) &= \sum_{s,t} \boldsymbol{F}_{smt} && (70)\\ &= \sum_{s,t} (L^+_{sm} - L^+_{tm} - L^+_{st} + L^+_{tt})\pi_m && (71)\\ &= |\mathcal{V}| \sum_{t} L^+_{tt}\pi_m && (72)\\ &= |\mathcal{E}| K \pi_m, && (73) \end{aligned}$$

where K is the Kirchhoff index (see Section 5.3). The third equality follows by using the fact that the column sum of the digraph Laplacian matrix $L^+ = (\Pi(I-P))^+$ is zero [27, 28]. For a (connected) undirected network, $\pi_m = \frac{d_m}{2|\mathcal{E}|}$, where d_m is the degree of node m.

For undirected networks, Newman [11] proposes a variation of the random walk betweenness measure defined above, which we denote by $Betweenness_{\mathrm{Newman,bidirect}}(m)$ (the use of subscript *bidirect* will be clear below): it is defined as the (net) electrical current flow I through a medial node in an undirected network (which can be viewed as an electrical resistive network with bi-directional links with resistance), when a unit current flow is injected at a source and removes at a target (ground), aggregated over all such source-target pairs. Formally, we have

$$\begin{aligned} Betweenness_{\mathrm{Newman,bidirect}}(m) &= \sum_{s,t} I(s \to m \to t)\\ &= \sum_{s,t} \sum_{k} \frac{1}{2} |\boldsymbol{F}_{smt} p_{mk} - \boldsymbol{F}_{skt} p_{km}|. \end{aligned}$$

We remark that the original definition given by Newman is based on current flows in electrical networks, and is only valid for *undirected* networks. Define $f(\boldsymbol{F}_{smt}) = \sum_k \frac{1}{2} |\boldsymbol{F}_{smt} p_{mk} - \boldsymbol{F}_{skt} p_{km}|$, then $Betweenness_{\mathrm{Newman,directed}}(m) = \sum_{s,t} f(\boldsymbol{F}_{smt})$ yields a general definition of Newman's random walk betweenness measure that also holds for directed networks. In particular, we show that if a network is strictly *unidirectional*, namely, if $e_{ij} \in E$ then $e_{ji} \notin E$, Newman's random walk

betweenness centrality reduces to $Betweenness_{\text{rw}}(m) = |\mathcal{E}|K\pi_m$:

$$\begin{aligned} Betweenness_{\text{Newman,unidirect}}(m) &= \sum_{s,t}\sum_{k} |\boldsymbol{F}_{smt}p_{mk}| \\ &= \sum_{s,t} \boldsymbol{F}_{smt} \sum_{k} p_{mk} \\ &= \sum_{s,t} \boldsymbol{F}_{smt} = |\mathcal{E}|K\pi_m, \end{aligned} \quad (74)$$

where K is the Kirchhoff index (see Section 5.3) and the last equality follows from Corollary 1.

5.3. Kirchhoff Index

The term *topological index* is a single metric that characterizes the topology ("connectivity structure") of a network; it has been widely used in mathematical chemistry to reflect certain structural properties of the underlying molecular graph [32][33]. Perhaps the most known topology index is the Kirchhoff index [13] which has found a variety of applications [34, 35, 36, 7, 37]. Kirchhoff index is also closely connected to Kemeney's constant [38, 39]. The Kirchhoff index is often defined in terms of *effective resistances* [13], $K(G) = \frac{1}{2}\sum_{s,t}\Omega_{st}$, which is closely related to commute times, as $\Omega_{st} = \frac{1}{|\mathcal{E}|}C_{st}$ [40]. Hence we have

$$K(G) = \frac{1}{2|\mathcal{E}|}\sum_{s,t} C_{st} = \frac{|\mathcal{V}|}{|\mathcal{E}|}\sum_{t} L^{+}_{tt} = \frac{1}{|\mathcal{E}|}\sum_{s,m,t} \boldsymbol{F}_{smt}, \quad (75)$$

where the second equality comes from Eq.(35). In other words, the Kirchhoff index can be computed by summing over all three dimensions in Figure 1, normalized by the total number of edges.

The relations between Kirchhoff index, effective resistance, and Laplacian matrix have been well studied in the literature. The authors in [7] provided three interpretations of L^{+}_{ii} as a topological centrality measure, from effective resistances in an electric network, random walk detour costs, and graph-theoretical topological connectivity via connected bi-partitions, and demonstrate that the Kirchhoff index, as a topological index, captures the overall robustness of a network. The relation between the effective resistance and the Moore-Penrose inverse of the Laplacian matrix is more elaborated in [41], and insightful relations between Kirchhoff index and inverses of the Laplacian eigenvectors can be found in [42, 43].

6. Characterization of Network Articulation Points and Network Load Distribution

We extend the definition of *articulation point* to a general (undirected and directed) network as a node whose removal reduces the amount of reachability in the network. For instance, in a network G, if t is previously reachable from s, i.e. there was at least one path from s to t, but t is no longer reachable from s after removing m, node m is an articulation point for network G. Note that s may still be reachable from t after removing m in a directed network, which is not the case for an undirected network. Hence, in an undirected network, the reduction in the number of reachabilities results to the increase in the number of connected components in the network, which is the reason to call articulation point as *cut vertex* in the undirected networks. Removal of an articulation point in a *directed* network, however, does not necessarily increase the number of connected components in the network.

As an application of the fundamental tensor, we introduce the normalized fundamental tensor $\hat{\boldsymbol{F}}$ and show that its entries contain information regarding articulation points in a general (directed or undirected) network. If $\boldsymbol{F}_{smt}$ exists, its *normalized* version is defined as follow,

$$\hat{\boldsymbol{F}}_{smt} = \begin{cases} \frac{\boldsymbol{F}_{smt}}{\boldsymbol{F}_{mmt}} & \text{if } s, m \neq t \\ 0 & \text{if } s = t \text{ or } m = t \end{cases} \tag{76}$$

The normalized fundamental tensor satisfies the following properties: a) $0 \leq \hat{\boldsymbol{F}}(s, m, t) \leq 1$, and b) $\hat{\boldsymbol{F}}_{smt} = Q_s^{\{m,\bar{t}\}}$. Recall that $Q_s^{\{m,\bar{t}\}}$ is the absorption probability that a random walk starting from node s hits (is absorbed by) node m sooner than node t. The second property (b) is a result of Theorem 4 and the first property (a) follows from (b). Clearly, $\hat{\boldsymbol{F}}_{smt} = Q_s^{\{m,\bar{t}\}} = 1$ means that with probability 1, any random walk starts from node s always hit node m before node t. Hence node m is on any path (thus walk) from s to t. Hence it is an articulation point. We therefore have the following statement:

Statement 2. *The normalized fundamental tensor captures the articulation points of a network: if $\hat{\boldsymbol{F}}_{smt} = 1$, then node m is an articulation point; namely, node m is located on all paths from s to t. On the other extreme, $\hat{\boldsymbol{F}}_{smt} = 0$ indicates that m is not located on any path from s to t and thus it plays no role for this reachability.*

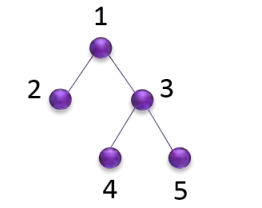
Figure 2 depicts two simple networks, one undirected and one directed, and displays the corresponding normalized fundamental tensors we have computed for these two networks (the tensors are "unfolded" as a series of matrices, each with fixed t). Any column that contains an entry with value 1 indicates the corresponding node m, $1 \leq m \leq 5$, is located on all paths between a pair of source and target, and so is an articulation point for the network [3]. Counting the number of 1's in each column m over the entire tensor yields the number of source-target pairs for which node m is an articulation point. The larger this count is, the more critical node m is for the overall network reachability. For instance, for both networks, node 3 is the most critical node for network reachabilities.

More generally, we can view $\hat{\boldsymbol{F}}_{smt}$ as a measure of how critical a role node m plays in the reachability from node s to node t. As a generalization of articulation points, we define the overall *load* that node m carries for all source-target pairs in a network as follows:

$$Load(m) = \frac{1}{(n-1)^2} \sum_{s,t} \hat{\boldsymbol{F}}_{smt}, \tag{77}$$

It is interesting to compare Eq.(77) with Eq.(70), where the latter (the unnormalized summation $\sum_{s,t} \boldsymbol{F}_{smt}$) is proportional to the stationary probability of node m (and degree of m if the network is undirected). The distribution of $Load(m)$'s provides a characterization of how balanced a network in terms of distributing its load (reachability between pairs of nodes), or how robust it is against targeted attacks. A network with a few high-valued articulation points (*e.g.*, a star network) is more vulnerable to the failure of a few nodes. Using a few synthetic networks with specific topologies as well as real-world networks as examples, Figure 3 plots the distribution of $Load(m)$ for these networks (sorted based on increasing values of $Load(m)$'s). Among the specific-shaped networks, it is interesting to note that comparing to a chain network, the loads on a cycle network are evenly distributed – this shows the huge difference that adding a single edge can make in the structural property of a network. It is not surprising that the complete graph has evenly distributed loads. In constrast, a star graph has the most skewed load distribution, with the center node as an articulation point of the net-

[3] As a convention, the source node is considered as the articulation point of the reachability, but not the target.



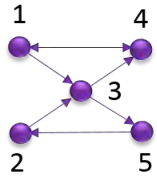
m						
s	1	2	3	4	5	t
1	0	0	0	0	0	1
2	0	1	0	0	0	1
3	0	0	1	0.5	0.5	1
4	0	0	1	1	0.5	1
5	0	0	1	0.5	1	1

m						
s	1	2	3	4	5	t
1	1	0	0.5	0.33	0.33	2
2	0	0	0	0	0	2
3	1	0	1	0.67	0.67	2
4	1	0	1	1	0.67	2
5	1	0	1	0.67	1	2

m						
s	1	2	3	4	5	t
1	1	0.5	0	0	0	3
2	1	1	0	0	0	3
3	0	0	0	0	0	3
4	0	0	0	1	0	3
5	0	0	0	0	1	3

m						
s	1	2	3	4	5	t
1	1	0.67	1	0	0.5	4
2	1	1	1	0	0.5	4
3	0.5	0.33	1	0	0.5	4
4	0	0	0	0	0	4
5	0.5	0.33	1	0	1	4

m						
s	1	2	3	4	5	t
1	1	0.67	1	0.5	0	5
2	1	1	1	0.5	0	5
3	0.5	0.33	1	0.5	0	5
4	0.5	0.33	1	1	0	5
5	0	0	0	0	0	5



m						
s	1	2	3	4	5	t
1	0	0	0	0	0	1
2	0	1	1	1	0.5	1
3	0	0.5	1	1	0.5	1
4	0	0	0	1	0	1
5	0	1	1	1	1	1

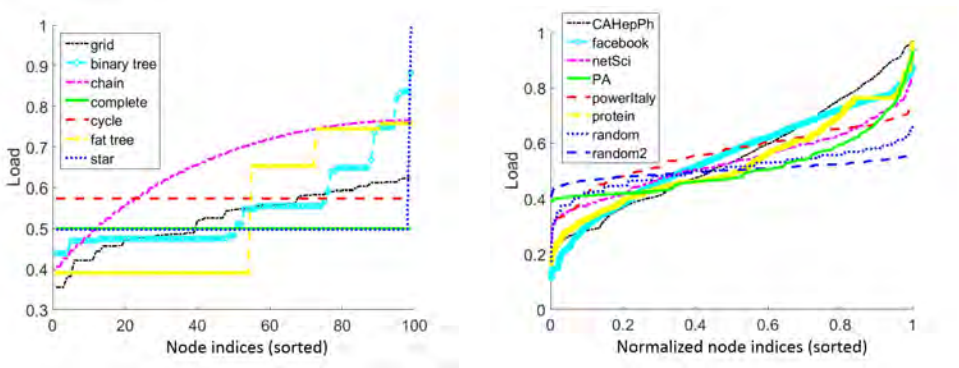
m						
s	1	2	3	4	5	t
1	1	0	1	0.75	1	2
2	0	0	0	0	0	2
3	0.5	0	1	0.5	1	2
4	1	0	1	1	1	2
5	0	0	0	0	1	2

m						
s	1	2	3	4	5	t
1	1	0	0	0.5	0	3
2	0	1	0	0	0	3
3	0	0	0	0	0	3
4	1	0	0	1	0	3
5	0	1	0	0	1	3

m						
s	1	2	3	4	5	t
1	1	0.25	0.5	0	0.25	4
2	0	1	1	0	0.5	4
3	0	0.5	1	0	0.5	4
4	0	0	0	0	0	4
5	0	1	1	0	1	4

m						
s	1	2	3	4	5	t
1	1	0	1	0.75	0	5
2	0.5	1	1	0.5	0	5
3	0.5	0	1	0.5	0	5
4	1	0	1	1	0	5
5	0	0	0	0	0	5

Figure 2: Two networks, one undirected and one directed, and the corresponding normalized fundamental tensor



(a) (b)

Figure 3: Load balancing in a) specific-shaped networks and b) real-world networks

work. Comparing the binary tree, the grid network has a less skewed load distribution. It is also interesting to compare the load distribution of the binary with that of a 3-ary "fat tree" network – such a topology is used widely in data center networks [44]. The real networks used in Figure 3(b) include the Arxiv High Energy Physics - Phenomenology collaboration network (CAHepPh) [45], a sampled network of Facebook [46], the coauthorship network of network scientists (netSci) [47], the Italian power grid [48], and a protein-protein interaction network [49]. For comparison, we also include three networks generated via two well-known random network models, the Preferential Attachment generative model (PA) [50] and Erdos Renyi (ER) random graph model [51] with two different initial links of 8 (random) and 40 (random2). We see that the two ER random networks yield most balanced load distributions, whereas the PA network exhibits behavior similar to a tree network, with a few nodes bearing much higher loads than others. The real networks exhibit load distributions varying between these types of random networks (with the Italian power grid closer to an ER random network, whereas netSci closer to a PA random network).

7. Most Influential Nodes in Social Networks

Online social networks have played a key role as a medium for the spread of information, ideas, or "influence" among their members. The Influence maximization problem in social networks is about finding the most influential persons who can maximize the spread of influence in the network. This

problem has applications in viral marketing, where a company may wish to spread the publicity, and eventually the adoption, of a new product via the most influential persons in popular social networks. A social network is modeled as a (directed or undirected) graph where nodes represent the users, and edges represent relationships and interactions between the users. An *influence cascade* over a network can be modeled by a diffusion process, and the objective of the influence maximization problem is to find the k most influential persons as the initial adopters who will lead to most number of adoptions.

The heat conduction model [52] is a diffusion process which is inspired by how heat transfers through a medium from the part with higher temperature to the part with lower temperature. In this diffusion process, the probability that a user adopts the new product is a linear function of adoption probabilities of her friends who have influence on her as well as her own independent tendency. We modeled the independent tendency of users for the product adoption by adding an *exogenous* node, indexed as o, and linked to all of the nodes in the network. Network G with added node o is called extended G, denoted by G^o. We showed that the influence maximization problem for $k = 1$, where $k = \#initial\ adopters$, under the heat conduction diffusion process has the following solution in terms of the normalized fundamental tensor over G^o [52]:

$$t^* = \arg\max_t \sum_{s\in\mathcal{V}} \hat{\boldsymbol{F}}_{sto}. \tag{78}$$

We also proved that the general influence maximization problem for $k > 1$ is NP-hard [52]. However, we proposed an efficient greedy algorithm, called *C2Greedy* [52], which finds a set of initial adopters who produce a provably near-optimal influence spread in the network. The algorithm iteratively finds the most influential node using Eq.(78), then removes it from the network and solves the equation to find the next best initiator.

Statement 3. *For $k = 1$, $\arg\max_t \sum_{s\in\mathcal{V}} \hat{\boldsymbol{F}}_{sto}$ finds the most influential node of network G^o as the initial adopter for maximizing the influence spread over the network with heat conduction [52] as the diffusion process. For $k > 1$, the greedy algorithm,* C2Greedy *[52], employs this relation to iteratively find the k most influential nodes, which yields a provably near-optimal solution.*

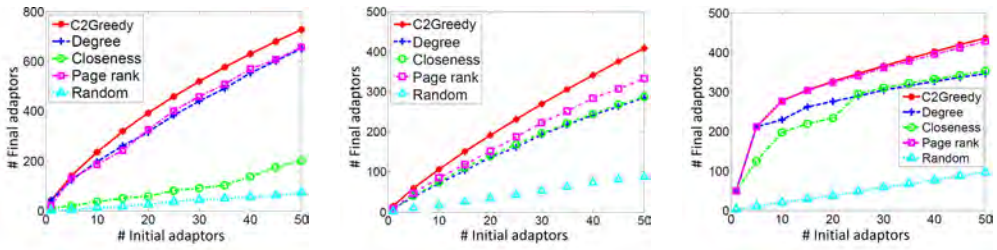
In [52], we showed that C2Greedy outperforms the state-of-the-art influence maximization algorithms in both performance and speed, which we do not repeat here. Instead, we present two new sets of experiments in the rest.

(a) (b)

Figure 4: Influence spread by a) two most influential nodes found from C2Greedy [52], b) two neighbors of nodes in part a

We remark that the metric in Eq.(78) addresses both the *global* characteristics of the network by placing the most influential node in the critical and strategical "center" of the network, and the *local* characteristics by specifying the highly populated and "neighbor-rich" nodes. In Figure 4(a), we visualize the influence spread of the two most influential nodes found from C2Greedy using the ESNet [53] network. The initiators are colored in black, and the green shades indicate the influence spread over the nodes in the network; the darker the green, the higher probability of production adoption for the node. In Figure 4(b), we pick two nodes, which are a neighbor of the two most influential nodes identified by C2Greedy, as the initiators, and visualize the probability of influence spread caused by these two nodes over other nodes in the network. The lower green intensity of Figure 4(b), compared to that of Figure 4(a) shows that not any two initiators – even if they are their immediate neighbors and *globally* located very closely – can cause the same influence spread as the two most influential nodes identified by C2Greedy.

Moreover, we show that the k most influential initiators found by C2Greedy have higher influence spread in the network compared to that of well-known centrality/importance measure algorithms: 1- top k nodes with highest (in-)degree, 2- top k nodes with highest closeness centrality scores, 3- top k nodes with highest Pagerank score [54], and 4- a benchmark which consists of k nodes picked randomly. For this purpose, we use real-world network data from three social networks, wiki vote [55], hepPh citation [56], and Facebook [46]. Figure 5 illustrates how the C2Greedy outperforms the other algorithms



(a) wiki vote (b) hepPh citation (c) facebook

Figure 5: Influence spread comparison between 5 different "most influential nodes" methods for a wide choice of initiator sizes.

for a wide range of k.

8. Fast Computation of Dynamic Network Reachability Queries

Reachability information between nodes in a network is crucial for a wide range of applications from gene interactions in bioinformatics [57] to XML query processing [58]. Given a network, a reachability query $R(s,t)$ has a binary answer with 1 indicating that target node t is reachable from source node s, and 0 representing that it is not. Several efficient algorithms have been devised to answer reachability queries when the network is *static* [59, 60, 61, 62]. However, few efficient solutions have been developed to answer reachability query for *dynamic* networks, *e.g.*, after node or link failures. For example, garbage collection in programming languages requires dynamic (re-)computation of reachability to balance the reclamation of memory, which might be reallocated. The speed of answering reachability queries affect the performance of applications [63].

As a final application of the fundamental tensor, we illustrate how it can be employed to develop an efficient algorithm to answer reachability queries for dynamic networks. Here we do *not* require the network G under consideration (before or after failures) be strongly connected, otherwise the reachability query problem is trivial. For simplicity of exposition, in the following we only consider node failures. Similar in Section 7, we add an exogenous node o to network G and connecting all the nodes to it. We note that this extended network G^o has only one recurrent equivalence class, and $\boldsymbol{F}_{sto}$ exists for any pairs of s and t. Moreover, $\boldsymbol{F}_{sto}$ is non-zero if and only t is reachable from s in G. This is because with non-zero probability

a random walk will visit every node that is reachable from s before hitting o. By pre-computing the fundamental matrix $F^{\{o\}}$ once, we can answer any reachability query $R(s,t)$ in constant time using $F^{\{o\}}$ by performing a table look-up.

Now suppose a set of nodes, $\mathcal{F}$, fail. We claim that we can answer the *dynamic* reachability query $R(s,t,\mathcal{F})$ (after the nodes in $\mathcal{F}$ fail, but *without* prior knowledge of the node failure set $\mathcal{F}$) in $O(|\mathcal{F}|)$. In particular, if $|\mathcal{F}|$ is of a constant order $O(1)$ compared to the size of network $|\mathcal{V}|$, then the queries are answered in constant $O(1)$ time. This is achieved by leveraging Theorem 3 for incremental computation of the fundamental matrix. Let $\mathcal{S} = \mathcal{F} \cup \{o\}$ and define $\boldsymbol{F}_{st\mathcal{S}}$

$$\boldsymbol{F}_{st\mathcal{S}} = \boldsymbol{F}_{sto} - \boldsymbol{F}_{s\mathcal{F}o}\boldsymbol{F}^{-1}_{\mathcal{F}\mathcal{F}o}\boldsymbol{F}_{\mathcal{F}to}, \tag{79}$$

which is the tensor form of $F^{\mathcal{S}}_{st} = F^{\{o\}}_{st} - F^{\{o\}}_{s\mathcal{F}}(F^{\{o\}}_{\mathcal{F}\mathcal{F}})^{-1}F^{\{o\}}_{\mathcal{F}t}$. Note that the sub-matrix $(F^{\{o\}}_{\mathcal{F}\mathcal{F}})^{-1}$ is non-singular. This comes from the fact that $F^{\{o\}}$ is an inverse M-matrix (an inverse M-matrix is a matrix whose inverse is an M-matrix), hence each of its principal sub-matrix is also an inverse M-matrix. We have the following statement:

Statement 4. *In the extended network G^o, $\boldsymbol{F}_{sto}$ is non-zero if and only if t is reachable from s in the original network G. Furthermore, if the nodes in the set $\mathcal{F}$ fail, $\boldsymbol{F}_{st\mathcal{S}}$ is non-zero (where $\mathcal{S} = \mathcal{F} \cup \{o\}$) if and only if t is still reachable from s in network G after the failures.*

Using the above statement and Theorem 3, we can answer (*static* and *dynamic*) reachability queries both before and after failures in constant times (for a constant size node failure set $\mathcal{F}$) by pre-computing $\boldsymbol{F}_{::o}$ ($=F^{\{o\}}$) and storing the results in a table. The method for answering reachability queries is summarized in Algorithm (1). The function $1_{\{b\}}$ is an indicator function which is equal to 1 if $b = True$ and 0 if $b = False$.

9. Conclusion

We revisited the fundamental matrix in Markov chain theory and extended it to the fundamental tensor which we showed that can be built much more efficiently than computing the fundamental matrices separately ($O(n^3)$ vs. $O(n^4)$) for the applications that the whole tensor is required. We also showed that fundamental matrix/tensor provides a unifying framework to

Algorithm 1 ANSWERING A REACHABILITY QUERY

query: $R(s, t, \sim \mathcal{F})$
input: transition matrix P of the extended network G^o
precomputed oracle: $\boldsymbol{F}_{::o} = (I - P_{\setminus o})^{-1}$
output: answer to reachability queries.
if $\mathcal{F} = \emptyset$ **then**
    $R(s, t) = 1_{\{\boldsymbol{F}_{sto} > 0\}}$
else
    $R(s, t, \sim \mathcal{F}) = 1_{\{\boldsymbol{F}_{sto} - \boldsymbol{F}_{s\mathcal{F}o}\boldsymbol{F}^{-1}_{\mathcal{F}\mathcal{F}o}\boldsymbol{F}_{\mathcal{F}to} > 0\}}$
end if

derive other Markov metrics and find useful relations in a coherent way. We then tackled four interesting network analysis applications in which fundamental tensor is exploited to provide effective and efficient solutions: 1) we showed that fundamental tensor unifies various network random-walk measures, such as distance, centrality measure, and topological index, via summation along one or more dimensions of the tensor; 2) we extended the definition of articulation points to the directed networks and used the (normalized) fundamental tensor to compute all the articulation points of a network at once. We also devised a metric to measure the load balancing over nodes of a network. Through extensive experiments, we evaluated the load balancing in several specifically-shaped networks and real-world networks; 3) we showed that (normalized) fundamental tensor can be exploited to infer the cascade and spread of a phenomena or an influence in social networks. We also derived a formulation to find the most influential nodes for maximizing the influence spread over the network using the (normalized) fundamental tensor, and demonstrated the efficacy of our method compared to other well-known ranking methods through multiple real-world network experiments; and 4) we presented a dynamic reachability method in the form of a pre-computed oracle which is cable of answering to reachability queries efficiently both in the case of having failures or no failure in a general directed network.

Acknowledgement

The research was supported in part by US DoD DTRA grants HDTRA1-09-1-0050 and HDTRA1-14-1-0040, and ARO MURI Award W911NF-12-1-0385. We would also like to thank our colleagues, Amir Asiaei and Arindam Banerjee, who helped us with the results reported in Section 7.

[1] V. Kopp, V. Kaganer, J. Schwarzkopf, F. Waidick, T. Remmele, A. Kwasniewski, M. Schmidbauer, X-ray diffraction from nonperiodic layered structures with correlations: analytical calculation and experiment on mixed aurivillius films, Acta Crystallographica Section A: Foundations of Crystallography 68 (1) (2012) 148–155.

[2] P. S. Kutchukian, D. Lou, E. I. Shakhnovich, Fog: Fragment optimized growth algorithm for the de novo generation of molecules occupying druglike chemical space, Journal of chemical information and modeling 49 (7) (2009) 1630–1642.

[3] K. G. Wilson, Quantum chromodynamics on a lattice, in: New Developments in Quantum Field Theory and Statistical Mechanics Cargèse 1976, Springer, 1977, pp. 143–172.

[4] D. Acemoglu, G. Egorov, K. Sonin, Political model of social evolution, Proceedings of the National Academy of Sciences 108 (Supplement 4) (2011) 21292–21296.

[5] L. Calvet, A. Fisher, Forecasting multifractal volatility, Journal of econometrics 105 (1) (2001) 27–58.

[6] J. He, X. Yao, From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms, IEEE Transactions on Evolutionary Computation 6 (5) (2002) 495–511.

[7] G. Ranjan, Z.-L. Zhang, Geometry of complex networks and topological centrality, Physica A: Statistical Mechanics and its Applications 392 (17) (2013) 3833–3845.

[8] G. Golnari, Y. Li, Z.-L. Zhang, Pivotality of nodes in reachability problems using avoidance and transit hitting time metrics, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 1073–1078.

[9] M. Chen, J. Liu, X. Tang, Clustering via random walk hitting time on directed graphs., in: AAAI, Vol. 8, 2008, pp. 616–621.

[10] P. Snell, P. Doyle, Random walks and electric networks, Free Software Foundation.

[11] M. E. Newman, A measure of betweenness centrality based on random walks, Social networks 27 (1) (2005) 39–54.

[12] J. D. Noh, H. Rieger, Random walks on complex networks, Physical review letters 92 (11) (2004) 118701.

[13] D. J. Klein, M. Randić, Resistance distance, Journal of mathematical chemistry 12 (1) (1993) 81–95.

[14] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: Structure and dynamics, Physics reports 424 (4-5) (2006) 175–308.

[15] F. Blöchl, F. J. Theis, F. Vega-Redondo, E. O. Fisher, Vertex centralities in input-output networks reveal the structure of modern economies, Physical Review E 83 (4) (2011) 046127.

[16] S. P. Borgatti, Centrality and network flow, Social networks 27 (1) (2005) 55–71.

[17] S. Fortunato, Community detection in graphs, Physics reports 486 (3-5) (2010) 75–174.

[18] L. Grady, Random walks for image segmentation, IEEE transactions on pattern analysis and machine intelligence 28 (11) (2006) 1768–1783.

[19] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens, Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation, IEEE Transactions on knowledge and data engineering 19 (3) (2007) 355–369.

[20] F. R. Gantmacher, Theory of Matrices. 2V., Chelsea publishing company, 1960.

[21] C. M. Grinstead, J. L. Snell, Introduction to probability, American Mathematical Soc., 2012.

[22] J. G. Kemeny, J. L. Snell, et al., Finite markov chains, Vol. 356, van Nostrand Princeton, NJ, 1960.

[23] J. R. Norris, Markov chains, no. 2, Cambridge university press, 1998.

[24] F. R. K. Chung, Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92), Cbms Regional Conference Series in Mathematics, 2006.

[25] Y. Li, Z.-L. Zhang, Random walks on digraphs: A theoretical framework for estimating transmission costs in wireless routing, in: INFOCOM '10: Proceedings of the 29th IEEE Conference on Computer Communications (To appear), IEEE, San Diego, USA, 2010.

[26] Y. Li, Z.-L. Zhang, Random walks on digraphs, the generalized digraph laplacian and the degree of asymmetry, in: LNCS WAW 2010, LNCS, Stanford, CA, 2010.

[27] Y. Li, Z.-L. Zhang, Digraph laplacian and degree of asymmetry, Internet Mathematics 8 (4) (2012) 381–401.

[28] D. Boley, G. Ranjan, Z.-L. Zhang, Commute times for a directed graph using an asymmetric laplacian, Linear Algebra and its Applications 435 (2) (2011) 224–242.

[29] C. D. Meyer, Jr, The role of the group generalized inverse in the theory of finite markov chains, Siam Review 17 (3) (1975) 443–464.

[30] S. J. Kirkland, M. Neumann, Group inverses of M-matrices and their applications, CRC Press, 2012.

[31] S. P. Borgatti, M. G. Everett, A graph-theoretic perspective on centrality, Social networks 28 (4) (2006) 466–484.

[32] D. H. Rouvray, Predicting chemistry from topology., Scientific American 255 (3) (1986) 40.

[33] D. H. Rouvray, The role of the topological distance matrix in chemistry., Tech. rep., DTIC Document (1985).

[34] B. Zhou, N. Trinajstić, A note on kirchhoff index, Chemical Physics Letters 455 (1) (2008) 120–123.

[35] J. L. Palacios, J. M. Renom, Bounds for the kirchhoff index of regular graphs via the spectra of their random walks, International Journal of Quantum Chemistry 110 (9) (2010) 1637–1641.

[36] E. Bendito, A. Carmona, A. Encinas, J. Gesto, M. Mitjana, Kirchhoff indexes of a network, Linear algebra and its applications 432 (9) (2010) 2278–2292.

[37] J. L. Palacios, Resistance distance in graphs and random walks, International Journal of Quantum Chemistry 81 (1) (2001) 29–33.

[38] X. Wang, J. L. Dubbeldam, P. Van Mieghem, Kemeny's constant and the effective graph resistance, Linear Algebra and its Applications 535 (2017) 231–244.

[39] S. Kirkland, Random walk centrality and a partition of kemenyâĂŹs constant, Czechoslovak Mathematical Journal 66 (3) (2016) 757–775.

[40] P. Tetali, Random walks and the effective resistance of networks, Journal of Theoretical Probability 4 (1) (1991) 101–109.

[41] I. Gutman, W. Xiao, Generalized inverse of the laplacian matrix and some applications, Bulletin: Classe des sciences mathematiques et natturalles 129 (29) (2004) 15–23.

[42] H.-Y. Zhu, D. J. Klein, I. Lukovits, Extensions of the wiener number, Journal of Chemical Information and Computer Sciences 36 (3) (1996) 420–428.

[43] I. Gutman, B. Mohar, The quasi-wiener and the kirchhoff indices coincide, Journal of Chemical Information and Computer Sciences 36 (5) (1996) 982–985.

[44] C. E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE transactions on Computers 100 (10) (1985) 892–901.

[45] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Transactions on Knowledge Discovery from Data (TKDD) 1 (1) (2007) 2.

[46] J. Leskovec, J. J. Mcauley, Learning to discover social circles in ego networks, in: Advances in neural information processing systems, 2012, pp. 539–547.

[47] M. E. Newman, Scientific collaboration networks. i. network construction and fundamental results, Physical review E 64 (1) (2001) 016131.

[48] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. Porcellinis, R. Setola, Modelling interdependent infrastructures using interacting dynamical models, International Journal of Critical Infrastructures 4 (1) (2008) 63–79.

[49] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, A.-L. Barabási, Hierarchical organization of modularity in metabolic networks, science 297 (5586) (2002) 1551–1555.

[50] A.-L. Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, Physica A: Statistical mechanics and its applications 311 (3) (2002) 590–614.

[51] P. Erdös, A. Rényi, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci 5 (17-61) (1960) 43.

[52] G. Golnari, A. Asiaee, A. Banerjee, Z.-L. Zhang, Revisiting non-progressive influence models: Scalable influence maximization in social networks., in: UAI, 2015, pp. 316–325.

[53] ESNet, Us energy science network, `http://www.es.net/`.

[54] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: Bringing order to the web., Tech. rep., Stanford InfoLab (1999).

[55] J. Leskovec, D. Huttenlocher, J. Kleinberg, Signed networks in social media, in: Proceedings of the SIGCHI conference on human factors in computing systems, ACM, 2010, pp. 1361–1370.

[56] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, 2005, pp. 177–187.

[57] J. Van Helden, A. Naim, R. Mancuso, M. Eldridge, L. Wernisch, D. Gilbert, S. J. Wodak, Representing and analysing molecular and cellular function in the computer, Biological chemistry 381 (9-10) (2000) 921–935.

[58] D. Chamberlin, Xquery: An xml query language, IBM systems journal 41 (4) (2002) 597–615.

[59] Y. Chen, Y. Chen, An efficient algorithm for answering graph reachability queries, in: Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, IEEE, 2008, pp. 893–902.

[60] F. Merz, P. Sanders, Preach: A fast lightweight reachability index using pruning and contraction hierarchies, in: European Symposium on Algorithms, Springer, 2014, pp. 701–712.

[61] R. Jin, G. Wang, Simple, fast, and scalable reachability oracle, Proceedings of the VLDB Endowment 6 (14) (2013) 1978–1989.

[62] H. Wang, H. He, J. Yang, P. S. Yu, J. X. Yu, Dual labeling: Answering graph reachability queries in constant time, in: Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on, IEEE, 2006, pp. 75–75.

[63] M. M. Michael, Safe memory reclamation for dynamic lock-free objects using atomic reads and writes, in: Proceedings of the twenty-first annual symposium on Principles of distributed computing, ACM, 2002, pp. 21–30.

10. Appendix

In this appendix, we provide the detailed derivations regarding the relations between the stochastic form and matrix form of hitting time and hitting costs, respectively.

• **Relation between the stochastic form and matrix form of hitting time**

Let t be the only absorbing node and rest of nodes belong to $\mathcal{T}$, then:

$$\begin{aligned} H_s^{\{t\}} &= \sum_{k=1} k \sum_{m\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm}[P_{\mathcal{T}\mathcal{A}}]_{mt} = \sum_{k=1} k \sum_{m\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm}(1 - \sum_{j\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}]_{mj}) \\ &= \sum_{k=1} k(\sum_{m\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm} - \sum_{j\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k}]_{sj}) = \sum_{m\in\mathcal{T}} \sum_{k=1} k([P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm} - [P_{\mathcal{T}\mathcal{T}}^{k}]_{sm}) \\ &= \sum_{m\in\mathcal{T}} [P_{\mathcal{T}\mathcal{T}}^{k-1}]_{sm} = \sum_{m} F_{sm}^{\{t\}}, \end{aligned} \tag{80}$$

which is the matrix form of hitting time Eq.(5).

• **Relation between the stochastic form and matrix form of hitting cost**

Let $\mathcal{Z}_{sm}$ be the set of all possible walks from s to m and ζ_j be the j-th walk from this set. We use $\mathcal{Z}_{sm}(l)$ to denote the subset of walks whose total length is l, and $\mathcal{Z}_{sm}(k,l)$ to specify the walks which have total length of l and total step size of k. Recall that a walk (in contrast to a path) can have repetitive nodes, and the length of a walk is the sum of the edge weights in the walk and its step size is the number of edges. Recall that $\mathbb{P}(\eta_t = l|X_0 = s)$ denotes the probability of hitting t in total length of l when starting from s, which can be obtained from the probability of walks: $\mathbb{P}(\eta_t = l|X_0 = s) = \sum_{\zeta_j\in Z_{st}(l)} \Pr_{\zeta_j}$. Probability of walk ζ_j denoted by $\Pr_{\zeta_j}$ is computed by the production over the probabilities of passing edges: $\Pr_{\zeta_j} = p_{sv_1}p_{v_1v_2}...p_{v_{k-1}m}$, where p_{vu} is the edge probability from v to u. The summation over the walk probabilities is computed using the following relation:

$$\sum_{\zeta_j\in Z_{sm}(k)} \Pr_{\zeta_j} = \begin{cases} [P_{\mathcal{T}\mathcal{T}}^{k}]_{sm} & \text{if } m \in \mathcal{T} \\ [P_{\mathcal{T}\mathcal{T}}^{k-1}P_{\mathcal{T}\mathcal{A}}]_{sm} & \text{if } m \in \mathcal{A} \end{cases} \tag{81}$$

With this introduction, the derivation of the stochastic form of hitting cost Eq.(9) can proceed as follows:

$$
\begin{aligned}
I\!H_s^{\{t\}} &= \sum_{l\in\mathcal{C}} l \sum_{k=1}^{<\infty} \sum_{\zeta_j\in\mathcal{Z}_{st}(k,l)} \Pr_{\zeta_j} && (82)\\
&= \sum_{l\in\mathcal{C}} \sum_{k=1}^{<\infty} \sum_{\zeta_j\in\mathcal{Z}_{st}(k,l)} l_{\zeta_j}\Pr_{\zeta_j} \\
&= \sum_{\zeta_j\in\mathcal{Z}_{st}} l_{\zeta_j}\Pr_{\zeta_j} && (83)\\
&= \sum_{\zeta_j\in\mathcal{Z}_{st}} \Pr_{\zeta_j} \sum_{k=1}^{k_{\zeta_j}} w_{v_{k-1}v_k} && (84)\\
&= \sum_{\zeta_j\in\mathcal{Z}_{st}} \sum_{k=1}^{k_{\zeta_j}} [\prod_{i=1}^{k} p_{v_{i-1}v_i}.(p_{v_kv_{k+1}}w_{v_kv_{k+1}}).\prod_{i=k+2}^{k_{\zeta_j}} p_{v_{i-1}v_i}] && (85)\\
&= \sum_{e_{xy}\in E} p_{xy}w_{xy}(\sum_{\zeta_j\in\mathcal{Z}_{sx}} \Pr_{\zeta_j}).(\sum_{\zeta_i\in\mathcal{Z}_{yt}} \Pr_{\zeta_i}) && (86)\\
&= \sum_{e_{xy}\in E} p_{xy}w_{xy}(\sum_{k}\sum_{\zeta_j\in\mathcal{Z}_{sx}(k)} \Pr_{\zeta_j}).(\sum_{k}\sum_{\zeta_i\in\mathcal{Z}_{yt}(k)} \Pr_{\zeta_i}) && (87)\\
&= \sum_{e_{xy}\in E} p_{xy}w_{xy}(\sum_{k}[P_{\mathcal{T}\mathcal{T}}^{k}]_{sx}).(\sum_{k}[P_{\mathcal{T}\mathcal{T}}^{k-1}P_{\mathcal{T}\mathcal{A}}]_{yt}) && (88)\\
&= \sum_{e_{xy}\in E} p_{xy}w_{xy}F_{sx}^{\{t\}}Q_y^{\{t\}} && (89)\\
&= \sum_{e_{xy}\in E} p_{xy}w_{xy}F_{sx}^{\{t\}} && (90)\\
&= \sum_{x} F_{sx}^{\{t\}} \sum_{y\in\mathcal{N}_{out}(x)} p_{xy}w_{xy} && (91)\\
&= \sum_{x} F_{sx}^{\{t\}} r_x, && (92)
\end{aligned}
$$

where l_{ζ_j} and k_{ζ_j} denote the length and step size of a walk ζ_j, respectively, and $r_x = \sum_{y\in\mathcal{N}_{out}(x)} p_{xy}w_{xy}$ is the average outgoing cost of node x. In the above derivation, Eq.(88) comes from Eq.(81), and Eq.(90) follows from the fact that $Q_y^{\{t\}} = 1$ when having t as the only absorbing node in the network and reachable from all the other nodes.