UNIVERSITY OF LEEDS

Article:
Lin, M, Li, W, Song, LJ orcid.org/0000-0002-0969-4091 et al. (3 more authors) (2021) SAKE: Estimating Katz Centrality Based on Sampling for Large-Scale Social Networks. ACM Transactions on Knowledge Discovery from Data, 15 (4). 66. ISSN 1556-4681

https://doi.org/10.1145/3441646

SAKE: Estimating Katz Centrality Based on Sampling for Large-Scale Social Networks

MINGKAI LIN, State Key Laboratory for Novel Software Technology, Nanjing University, China
WENZHONG LI*, State Key Laboratory for Novel Software Technology and Sino-German Institutes of Social Computing, Nanjing University, China
LYNDA J. SONG, University of Leeds, United Kindom
CAM-TU NGUYEN, State Key Laboratory for Novel Software Technology, Nanjing University, China
XIAOLIANG WANG, State Key Laboratory for Novel Software Technology, Nanjing University, China
SANGLU LU, State Key Laboratory for Novel Software Technology and Sino-German Institutes of Social Computing, Nanjing University, China

Katz centrality is a fundamental concept to measure the influence of a vertex in a social network. However, existing approaches to calculating Katz centrality in a large-scale network is unpractical and computationally expensive. In this paper, we propose a novel method to estimate Katz centrality based on graph sampling techniques, which object to achieve comparable estimation accuracy of the state-of-the-arts with much lower computational complexity. Specifically, we develop a Horvitz-Thompson estimate for Katz centrality by using a multi-round sampling approach and deriving an unbiased mean value estimator. We further propose **SAKE**, a **S**ampling based **A**lgorithm for fast **K**atz centrality **E**stimation. We prove that the estimator calculated by **SAKE** is probabilistically guaranteed to be within an additive error from the exact value. Extensive evaluation experiments based on four real world networks show that the proposed algorithm can estimate Katz centralities for partial vertices with low sampling rate, low computation time, and it works well in identifying high influence vertices in social networks.

CCS Concepts: • **Information systems** → **Data mining**; **Social networks**.

Additional Key Words and Phrases: Social Network, Katz Centrality, Graph Sampling

ACM Reference Format:
Mingkai Lin, Wenzhong Li, Lynda J. Song, Cam-tu Nguyen, Xiaoliang Wang, and Sanglu Lu. 2020. SAKE: Estimating Katz Centrality Based on Sampling for Large-Scale Social Networks. *ACM Trans. Knowl. Discov. Data.* 1, 1, Article 1 (January 2020), 21 pages. https://doi.org/10.1145/3441646

*The corresponding author is Wenzhong Li, Email: lwz@nju.edu.cn.

Authors' addresses: Mingkai Lin, mingkai@smail.nju.edu.cn, State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China; Wenzhong Li, lwz@nju.edu.cn, State Key Laboratory for Novel Software Technology and Sino-German Institutes of Social Computing, Nanjing University, Nanjing, China; Lynda J. Song, L.Song@leeds.ac.uk, University of Leeds, Leeds, United Kindom; Cam-tu Nguyen, ncamtu@nju.edu.cn, State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China; Xiaoliang Wang, waxili@nju.edu.cn, State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China; Sanglu Lu, sanglu@nju.edu.cn, State Key Laboratory for Novel Software Technology and Sino-German Institutes of Social Computing, Nanjing University, Nanjing, China.

1556-4681/2020/1-ART1 $15.00
https://doi.org/10.1145/3441646

1 INTRODUCTION

With the rapid development of social network platforms such as Facebook and Twitter, there is a growing interest in network analysis and its applications in social networks. Network analysis is often used to study some important network characteristics such as the degree distribution, the strength of social ties, the average path length between vertices, and the community structure. One of the most fundamental concepts in network analysis is vertex *centrality*, which is normally measured by a real-value function to represent the significance and importance of a vertex in the network [9]. Typical centrality measurements include degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality (i.e., PageRank [32]). In this paper, we focus on *Katz centrality* [21], a special extension of the eigenvector centrality [7]. Unlike other centrality measurements, Katz centrality measures the "influence" of a vertex by taking into account the weighted sum of the walks between the node and the other nodes in the social network. As a result, it can be used in a wide range of AI applications [22, 26], such as finding the most influential users in a social network, forming the word-of-mouth effect, promoting the adoption of innovation, widening the spread of public opinion, and viral marketing [35].

Katz centrality is introduced by Leo Katz in 1953. It computes the relative influence of a vertex by measuring the number of walks from the vertex to the other vertices in the network multiplied by an attenuation factor. Mathematically, Katz centrality of a vertex i can be formulated as:

$$C_{katz}(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{n} \alpha^k (A^k)_{ij}, \tag{1}$$

where A is the adjacency matrix of the network and $\alpha \in (0, 1)$ is the attenuation factor. The definition in Eq. (1) is referred to as the *original Katz centrality* thereafter.

Computation of Katz centrality is non-trivial. According to the definition, the computation involves matrix multiplication. For a graph with n nodes, the adjacency matrix A is a $n \times n$ matrix, and the computational complexity of the exact Katz centrality is $O(n^3)$. A more efficient approach is to solve a linear system when the system is fairly small by using Cholesky decomposition, which costs $O(n^2)$ [31]. Some approximation algorithms [14, 30] are developed based on iterative methods that respectively cost $O(m + n)$ and $O(km)$, where k is the number of limited walks, m is the number of edges in the graph and $m \in O(n^2)$ in the worst case. For large scale network where the number of edges m can also be very large as millions or billions and thus these approximation methods may be still computationally expensive. What's more, these methods need to be conducted by going through the whole network to compute the approximate Katz values.

Based on such consideration, we seek to develop a novel approximate method using sampling technique, with the objective of achieving much lower computational complexity than the state-of-the-arts while retaining comparable estimation accuracy. Specifically, we sample small subsets of vertices and edges from the social network, and use the sampled data to estimate the truncated Katz centrality with limited number of walks. As shown in Eq. (1), the value α^k goes to 0 as the number of walks k goes to infinite, therefore the truncated Katz centrality will approach the original Katz centrality with the increasing number of walks. Since the estimation is based on small subsets of the original network, the proposed method is computationally efficient and practical for large-scale networks.

We firstly develop a theoretical estimator for Katz centrality based on a multi-round graph sampling approach. Using the Horvitz-Thompson theory [19], we prove that the proposed estimator is an unbiased estimator of the Katz centrality (*Theorem 4.1*). We further propose **SAKE**, a **S**ampling based **A**lgorithm for **K**atz centrality **E**stimation. The main idea of **SAKE** is to use a random node sampling design, and compute the Katz centrality estimator by counting the number of walks in the

sampled subgraph. We prove that the estimated Katz centrality computed by **SAKE** is probabilistically guaranteed to be upper-bounded by an additive error from the exact value (*Theorem 4.3* and *4.4*). Moreover, we show that the computational complexity of **SAKE** is $O(l(d-1)\frac{r^2}{n^2}m+ls)$, where r is the size of the sampled subset, s is the size of the successors set for the objective vertex and l, d are small constants, which is more efficient than the state-of-the-arts. We conduct extensive evaluation experiments based on four datasets collected from real world social networks, which show that **SAKE** achieves low mean relative error, and it is efficient in identifying high influence vertices in social networks.

The contributions of the paper are summarized as follows.

- We are the first to propose the idea and design of multi-round sampling technique. To the best of our knowledge, using multi-round samples to approximate Katz centrality in the original graph has not been found in the literature before, and our work address the fundamental problem of inferring global vertex centrality via graph sampling theory, which has important implications in large-scale social networks.
- We propose an efficient algorithm called **SAKE** for Katz centrality estimation based on sampling technique. The proposed algorithm has several advantages: (1) The proposed estimator is proved to be unbiased in theory (*Theorem 4.1*); (2) The estimation error is guaranteed by a provable bound with high probability (*Theorem 4.3 and 4.4*); and (3) The computational complexity is $O(l(d-1)\frac{r^2}{n^2}m+ls)$, where r is the size of the sampled subset, s is the size of the successors set for the objective vertex and l, d are small constants, which is more efficient than the state-of-the-arts.
- The efficiency and feasibility of the proposed algorithm is verified by extensive experiments based on four real world social networks. It is shown that the proposed algorithm runs up to 10x faster than the baseline algorithms with comparable estimation accuracy.

2 RELATED WORK

2.1 Computation of Katz Centrality

Katz centrality is introduced by Leo Katz in 1953 to measure the relative influence of a vertex in a network [21]. A few axioms for Eigenvector and Katz centralities have been proposed in [6, 41]. Katz centrality has been shown to be useful in ranking users in social networks [31] and searching disease genes from gene expression and protein interaction networks [42].

For a graph with size n, Katz centrality can be calculated exactly by solving a linear system. In this case the expression $((I-\alpha A^T)^{-1}-I)\overrightarrow{1}$ can be used to obtain Katz centralities for all vertices. Obtaining an exact solution via such direct methods is prohibitively computationally expensive, since we are required to take the inverse of a matrix where the complexity is $O(n^3)$. If the system is fairly small (meaning the $n \times n$ matrix $I-\alpha A^T$ is not very large), we can calculate Katz values exactly in $O(n^2)$ using Cholesky decomposition. But in real world we usually observe that graphs of interest like Facebook are very large and sparse. Under such circumstances n^2 is much larger than m where m is the number of edges in the graph. Thus Foster et al. [14] presents an $O(m+n)$ vertex-centric heuristic algorithm for Katz centrality by aggregating information from neighbors through iteration until a predefined number of rounds is reached. The algorithm performs well and is widely used in many toolkits for complex network. Furthermore, an algorithm in [30] also uses iterative method to approximate a special personalized variant of Katz centrality with cost of $O(km)$ by computing weighted counts of the number of walks from one vertex to all the vertices in the graph where k is the number of limited walks. Another approximate algorithm [38] is proposed for Katz ranking computation in a special case where $\alpha < 1/\deg_{\max}$. However, the existing

approximation algorithms are mostly operated on the adjacency matrix A of the whole network, which are inevitably computationally expensive for large networks with billions of vertices.

2.2 Social Network Sampling Techniques

Sampling techniques in social networks aim to obtain a smaller graph which can represent the original network so that some useful results can be concluded from the subgraph. There has been a rich literature in statistics, data mining and physics on estimating graph properties using a small subsamples [3, 27, 34]. The works of [2, 3, 24, 27, 33] provide excellent surveys to graph sampling methods. Many sampling methods are based on random crawls [24, 29], such as forest-fire sampling, snowball sampling, and expansion sampling. As has been detailed in previous studies, these methods tend to bias certain parts of the network [15, 23, 27, 34]. And especially for the global centralities like betweenness and closeness centrality, the previous works [8, 10] demonstrate that these centralities are sensitive to the sampling data. Thus obtaining an unbiased estimator of the Katz centrality with a low sampling rate is non-trivial. Not only that, the paper of [39] claims that different network sampling techniques are highly sensitive with regard to capturing the expected centrality of nodes.

As a consequence, there are several attempts to correct the estimation error of sampling techniques. For example respondent-driven sampling (RDS) is a widely used network sampling method to reduce biases introduced by snowball sampling and other chain-referral methods [17, 43]. In [11], the authors illustrate a statistical sampling theory based on Horvitz-Thompson estimation, and suggest that various graph summaries are expressible in terms of the totals by examining per objective unit. The work of [34] analyze the estimation accuracy of degree distributions from graph sampling. Also several works adopt sampling techniques to infer different network characteristics. Wang et al.[40] describe a randomized approximation algorithm **RAND** for closeness centrality in weighted graphs . The fast approximation of Betweenness centrality using VC-dimension theory is proposed in [20, 36]. Fast calculation of clustering coefficient and many other properties by sampling are proposed in [13, 16]. Provable approximation approach for degree distribution using sublinear graph samples is further discussed in [12].

To the best of our knowledge, estimation of Katz centrality via sampling has not been addressed in the literature. Our work makes the first attempt to develop a sampling based method to achieve unbiased Katz centrality estimation.

3 NOTATIONS AND DEFINITIONS

In this section we introduce the notations and definitions that will be used for analysis throughout the paper. The notations used in the rest of the paper are summarized in Table 1.

Let $G = (V, E)$ be a graph, where V is the set of n vertices and E is the set of m edges and $\mathcal{R}(i)$ is the successors of vertex i. Denote the $n \times n$ adjacency matrix A of G with entries $A_{ij} = 1$ if there exits an edge from vertex i to j, and 0 otherwise.

We define *truncated Katz centrality* as:

$$C_{katz}(i) = \sum_{k=1}^{d} \sum_{j=1}^{n} \alpha^k (A^k)_{ij} \tag{2}$$

where d is the upper limit of the number of walks between a pair of vertices in the network. When $d \to \infty$, the above definition approaches the original Katz centrality.

Katz centrality quantifies the ability of a vertex to initiate walks around the network. The number of walks of length k from vertex i to j is $(A^k)_{ij}$. Katz centrality of vertex i counts the number

Table 1. Table of notations

Notation	Description
m, n, A	Nodes and edges number of graph $G = (V, E)$ with adjacency matrix A.
A_{ij}	The element in the i row, j column of adjacency matrix A.
$Katz(i)$	Katz centrality for vertex i.
$\mathcal{R}(i)$	The successors of vertex i.
α	A fixed attenuation factor to penalize long walks.
$D_i^k(G)$	Walks set in G from i with the length of walks k.
$\pi_v^{S_k}$	The probability that a vertex v is included in S_k.
S_k	The set of vertices in k-th sampling procedure (S_0 is the set of objective vertex).
$I, \overrightarrow{1_k}$	I is an $n \times n$ identity matrix and $\overrightarrow{1_k}$ is a $\lvert S_k \rvert \times 1$ vector of all 1s.
$B^{(k)}$	Adjacency matrix constructed from S_{k-1} and S_k.
$B^{[k]}, b^{[k]}$	$B^{[k]}$ is the product of the first k adjacency matrices and $b^{[k]} = B^{[k]}\overrightarrow{1_k}$.
l, d	The calculating loops and maximum walks length
r	Sampled vertices number in each process.

of closed walks beginning at vertex i, while penalizing long walks by multiplying a fixed attenuation factor $\alpha \in (0, 1)$ for distant walk. Unlike the conventional definition of Katz centrality that counts infinity walks, we consider Katz centrality with walks of maximum length d , which is more practical and computable in real large-scale networks.

We denote $D_i^k(G)$ $(k = 1, 2, \cdots, d)$ as the set of all walks in graph G beginning from i with the length of walks k. Note that the number of walks of length k from vertex i to j equals $(A^k)_{ij}$. Formally we can represent the Katz centrality of vertex i with limited walks d as:

$$Katz(i) = \sum_{k=1}^{d} \alpha^k \left| D_i^k(G) \right|, \tag{3}$$

where $\left|D_i^k(G)\right|$ is the cardinality of $D_i^k(G)$ that can be calculated by:

$$\left| D_i^k(G) \right| = \sum_{j \in V} (A^k)_{ij} = \sum_{v_1 \in \mathcal{R}(i)} \sum_{v_2 \in V} \cdots \sum_{v_k \in V} A_{iv_1} A_{v_1 v_2} \cdots A_{v_{k-1} v_k} \tag{4}$$

The term $A_{iv_1} A_{v_1 v_2} \cdots A_{v_{k-1} v_k}$ in the right part of the above equation is a 0-1 indicator representing whether there exists a walk from vertex i to v_k with length k. According to the definition of adjacency matrix, if the walk $i \to v_1 \to v_2 \to \cdots \to v_k$ exists, $A_{iv_1} A_{v_1 v_2} \cdots A_{v_{k-1} v_k}$ equals 1, and otherwise 0.

In this paper, we focus on computing a $(\epsilon, \delta) - approximation$ of $Katz(i)$ with the following definition.

Definition 3.1 ((ϵ,δ)-approximation). Given ϵ,$\delta \in (0, 1)$, the estimator $\overline{Katz(i)}$ is a (ϵ,δ)- approximation of $Katz(i)$ if it satisfies:

$$Pr(|\overline{Katz(i)} - Katz(i)| \leq \epsilon) \geq 1 - \delta. \tag{5}$$

4 KATZ CENTRALITY ESTIMATION

In this section, we propose the sampling approach and estimation algorithm for Katz centrality estimation.

4.1 Sampling and Estimation Method

According to the definitions in Eq. (3) and (4), to calculate $\left|D_i^k(G)\right|$, we need to compute the d-th power of the $n \times n$ adjacency matrix A, which complexity is $O(dn^3)$. In this paper, we explore the method of Katz centrality estimation based on sampling technique. The basic idea is to sample a small subset of vertices and edges from G, based on which we can form an unbiased estimation of Katz centrality for an objective vertex. The tool we use to estimate Katz centrality is Horvitz-Thompson estimator, which is introduced in the following.

The Horvitz-Thompson estimator [19] is a method for estimating the total and mean of a population from sampling. Suppose we have a population $U = \{1, 2, \cdots, N_u\}$ of N_u units, and with each unit $j \in U$ there is an associated value y_j. Let $\tau = \sum_j y_j$ be the total value of y's in the population. Let $U_s \subset U$ be an independent sample of n_s distinct units from U. We observe y_j for each $j \in U_s$, and suppose under the given sampling design each unit $j \in U$ has probability π_j being included in U_s. The Horvitz-Thompson estimator gives an unbiased estimation of the total τ by $\hat{\tau} = \sum_{j \in U_s} \frac{y_j}{\pi_j}$.

Inspired by the Horvitz-Thompson estimator, we derive an unbiased estimate for $\left|D_i^k(G)\right|$ as follows. The key of deriving $\left|D_i^k(G)\right|$ is to estimate the number of walks from i with length k. To achieve that, we develop a k-round sampling design. in this k-round sampling design, for each round we sample a set of vertices S_j $(j = 1, \cdots, k)$ from V without replacement.

Let $\pi_v^{S_j}$ be the probability that a vertex v is included in S_j during the sampling process. In order to ease the notations in this work, we use $p_{iv_1v_2\ldots v_k} = A_{iv_1}A_{v_1v_2}\cdots A_{v_{k-1}v_k}$ and $\pi_{v_1v_2\ldots v_k} = \pi_{v_1}^{S_1}\pi_{v_2}^{S_2}\cdots\pi_{v_k}^{S_k}$. Thus, the value of $\left|D_i^k(G)\right|$ can be estimated by

$$\widehat{\left|D_i^k(G)\right|} = \sum_{v_1 \in S_1}\sum_{v_2 \in S_2}\cdots\sum_{v_k \in S_k}\frac{p_{iv_1v_2\ldots v_k}}{\pi_{v_1v_2\ldots v_k}}. \tag{6}$$

The numerator $p_{iv_1v_2\ldots v_k} = A_{iv_1}A_{v_1v_2}\cdots A_{v_{k-1}v_k}$ in the right part of Eq. (6) indicates the existence of a walk from vertex i to v_k with length k in the sampling nodes, and the denominator $\pi_{v_1v_2\ldots v_k} = \pi_{v_1}^{S_1}\pi_{v_2}^{S_2}\cdots\pi_{v_k}^{S_k}$ is used for correcting the bias according to Horvitz-Thompson estimation. For the reason that the successors of objective vertex i are easy to obtain and there is no need to estimate it, thus when $k = 1$ we can just select all the successors for vertex i as S_1. Under such condition, we have $\pi_{v_1}^{S_1} = 1$ and $\widehat{\left|D_i^1(G)\right|} = \left|D_i^1(G)\right|$ without doubt. It means that at any time $\widehat{\left|D_i^1(G)\right|}$ is always equal to a constant $|\mathcal{R}(i)|$ which is the number of successors for vertex i. Based on the k-round sampling design, for *truncated Katz centrality*, an estimate $\widehat{Katz(i)}$ of $Katz(i)$ can be derived by

$$\begin{aligned}\widehat{Katz(i)} &= \alpha|\mathcal{R}(i)| + \sum_{v_1 \in S_1}\sum_{v_2 \in S_2}\frac{\alpha^2 p_{iv_1v_2}}{\pi_{v_1v_2}} + \cdots + \\ &\quad \sum_{v_1 \in S_1}\sum_{v_2 \in S_2}\cdots\sum_{v_d \in S_d}\frac{\alpha^d p_{iv_1v_2\ldots v_d}}{\pi_{v_1v_2\ldots v_d}} \\ &= \sum_{k=1}^{d}\alpha^k\widehat{\left|D_i^k(G)\right|}.\end{aligned} \tag{7}$$

We compute such estimate for l times and will obtain a series of values for $\widehat{Katz(i)}$. We denote these values as $\widehat{Katz(i)}[1], \widehat{Katz(i)}[2], ..., \widehat{Katz(i)}[l]$. The mean of these values is:

$$\overline{Katz(i)} = \frac{\sum_{t=1}^{l} \widehat{Katz(i)}[t]}{l} \tag{8}$$

The final estimator we propose for $Katz(i)$ is the mean value $\overline{Katz(i)}$. And the following theorem shows that $\overline{Katz(i)}$ is an unbiased estimation of $Katz(i)$.

Theorem 4.1. *To estimate* truncated Katz centrality, *if we adopt the d-rounds independent sampling process* $S_1, S_2, \cdots, S_d$ *to construct the estimate* $\widehat{Katz(i)}$ *as Eq. (7), and compute the final estimator* $\overline{Katz(i)}$ *as Eq. (8) then we have* $E(\overline{Katz(i)}) = Katz(i)$.

Proof. Firstly, we prove that $\widehat{\left|D_i^k(G)\right|}$ is an unbiased estimate of $\left|D_i^k(G)\right|$. We introduce an 0-1 indicator to represent whether a vertex is included in the sampled set S_j:

$$I(v \in S_j) = \begin{cases} 1, & if\ vertex\ v\ is\ included\ in\ S_j \\ 0, & otherwise \end{cases} \tag{9}$$

Since the d-round sampling processes are independent, we have

$$\begin{aligned}
E(\widehat{\left|D_i^k(G)\right|}) &= E[\sum_{v_1 \in S_1} \sum_{v_2 \in S_2} \cdots \sum_{v_k \in S_k} \frac{p_{iv_1v_2v_{k-1}v_k}}{\pi_{v_1v_2\ldots v_{k-1}v_k}}] \\
&= E[\sum_{v_1 \in \mathcal{R}(i)} \sum_{v_2 \in V} \cdots \sum_{v_k \in V} \frac{p_{iv_1v_2v_{k-1}v_k}}{\pi_{v_1v_2\ldots v_{k-1}v_k}} \cdot \prod_{j=2}^{k} I(v_j \in S_j)] \\
&= \sum_{v_1 \in \mathcal{R}(i)} \sum_{v_2 \in V} \cdots \sum_{v_k \in V} \frac{p_{iv_1v_2v_{k-1}v_k}}{\pi_{v_1v_2\ldots v_{k-1}v_k}} \cdot E[\prod_{j=2}^{k} I(v_j \in S_j)] \\
&= \sum_{v_1 \in \mathcal{R}(i)} \sum_{v_2 \in V} \cdots \sum_{v_k \in V} \frac{p_{iv_1v_2v_{k-1}v_k}}{\pi_{v_1v_2\ldots v_{k-1}v_k}} \cdot \prod_{j=2}^{k} E[I(v_j \in S_j)] \\
&= \sum_{v_1 \in \mathcal{R}(i)} \sum_{v_2 \in V} \cdots \sum_{v_k \in V} p_{iv_1v_2\ldots v_{k-1}v_k} \\
&= \left|D_i^k(G)\right|.
\end{aligned}$$

In this formula, the second equation is because of $\sum_{v_i \in S_j} X_{v_i} = \sum_{v_i \in V} I(v_i \in S_j) X_{v_i}$ for a variable X; the third equation is because of the linearity of the expectation; the fourth equation is because of $I(v_i \in S_j)$ for d sampling rounds are independent; and the fifth equation is because of $E(I(v_i \in S_j)) = \pi_{v_i}^{S_j}$.

Applying Eq. (7), Eq. (8) and Eq. (3), we have

$$E(\overline{Katz(i)}) = E(\frac{\sum_{t=1}^{l}\widehat{Katz(i)}[t]}{l}) = \frac{\sum_{t=1}^{l}E(\widehat{Katz(i)})[t]}{l}$$
$$= E(\widehat{Katz(i)}) = E(\sum_{k=1}^{d}\alpha^k\widehat{\left|D_i^k(G)\right|}) = \sum_{k=1}^{d}\alpha^k E(\widehat{\left|D_i^k(G)\right|})$$
$$= \sum_{k=1}^{d}\alpha^k\left|D_i^k(G)\right| = Katz(i).$$

□

4.2 Sampling Based Katz Computation

We in this paper propose an approximation algorithm named **SAKE**, a **S**ampling based **A**lgorithm for **K**atz centrality **E**stimation. It computes an approximation for the Katz centrality for a set of vertices in batch based on the estimator derived in Theorem 4.1.

Assume that $S_0 = \{i\}$ is the objective vertex that we want to estimate its Katz centrality. In each calculation of the estimate $\widehat{Katz(i)}$, to explore the network connectivity condition up to d hops, we conduct d rounds of sampling and computation as follows. In the first sampling procedure we select all the successors of the objective vertex i as $S_1 = \{v_1^{(1)}, v_2^{(1)}, \cdots, v_{|\mathcal{R}(i)|}^{(1)}\}$. While for the $k\text{-}th(k = 2, \cdots, d)$ sampling procedure, we generate vertices set $S_k = \{v_1^{(k)}, v_2^{(k)}, \cdots, v_{|S_k|}^{(k)}\}$ by sampling from V with probabilities $P_k = \{\pi_1^{(k)}, \pi_2^{(k)}, \cdots, \pi_{|V|}^{(k)}\}$, where each element in P_k is the probability that the corresponding vertex is included in S_k according to the sampling method. Note that in the first sampling round, each corresponding probability in P_1 is always equal to 1. In each round, we construct an adjacency matrix $B^{(k)}$ based on S_{k-1} and S_k $(k = 1, \cdots, d)$, where the element $B_{xy}^{(k)}$ $(x \in S_{k-1}, y \in S_k)$ is 1 if there is an edge from x to y in the graph G, and 0 otherwise. We further replace each number 1 in $B_{xy}^{(k)}$ $(x \in S_{k-1}, y \in S_k)$ with number $1/\pi_y^{(k)}$ for the reason of correcting bias. In this way, we can get d matrices $B^{(k)}$ $(k = 1, 2, \cdots, d)$.

Fig. 1 illustrates a sampling process and the adjacency matrices. We refer to the graph structure formed by a sampling process as a *sampled graph*. We define $B^{[k]} = \prod_{j=1}^{k} B^{(j)}$ as the product of the first k adjacency matrices. As illustrated in Fig. 1, $B^{[k]}$ is a $|S_0| \times |S_k|$ matrix, where each row in the matrix represents the number of walks from the objective vertex in S_0 to the vertices in S_k. We define $b^{[k]}$ as the row sum of $B^{[k]}$. The value $b^{[k]} = \sum_{x=1}^{|S_k|} B_x^{[k]}$ equals the estimator $\widehat{\left|D_i^k(G)\right|}$ in Eq. (6) where $\widehat{\left|D_i^k(G)\right|}$ represents the total walks number from i to all vertices in S_k with length k going through the selected nodes in $S_1, S_2, ..., S_{k-1}$. According to Theorem 4.1, each estimates for Katz centrality of the vertex i can be calculated by $\alpha|\mathcal{R}(i)| + \sum_{k=2}^{d}\alpha^k b^{[k]}$.

In the above analysis, we do not place any restriction on the sampling methods and sampling size. Theoretically, all the sampling methods and sampling sizes can fit the proposed Katz centrality estimator. However, in order to improve the computational efficiency, we should make two simplifications in the sampling process: (1) We adopt the *simple random sampling* method and fix the sampling probability for each node in the kth $(k = 2, 3, ..., d)$ sampling round. The reason of adopting node sampling is that the inclusion probability is always $|S_k|/n$ for all vertices, which is easier to derive comparing to other sampling methods like random walk and random edge sampling. (2) We fix the number of sampled nodes in each round to r (in the first round the number is $|\mathcal{R}(i)|$). In this way the inclusion probability of each vertex can be always 1 for the first round, and

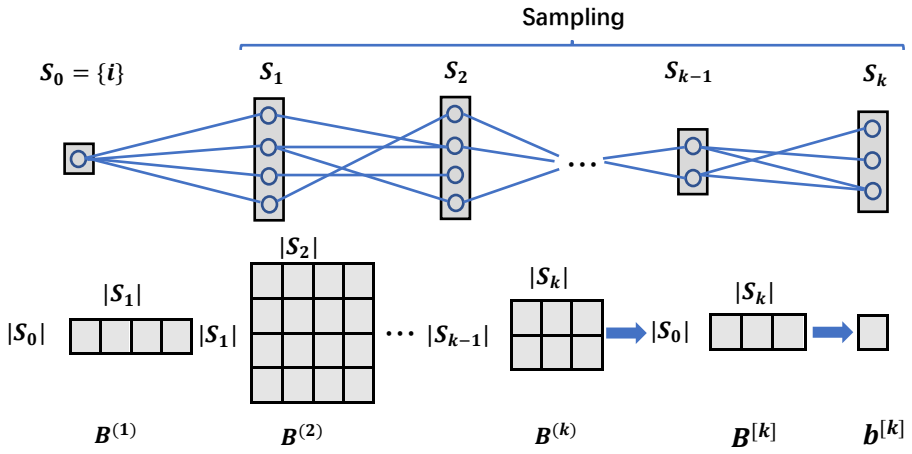


Fig. 1. Illustration of adjacency matrices in k-round sampling.

$\pi = \frac{r}{|V|}$ for the later sampling rounds. Therefore, the computation of the Katz estimator $\widehat{Katz(i)}$ can be simplified greatly.

From the analysis we can find that the basic algorithm runs for l loops, and in each loop, it computes the multiplication of two matrices for d times. Thus, the procedure of two matrices multiplication is repeated for ld times in total. While the procedure of multiplication is processed generally between two matrices with sizes $|S_0| \cdot r$ and $r \cdot r$ which costs $O(|S_0|r^2)$. Therefore when $|\mathcal{R}(i)| \leq r$, the total computation is $O(ld\,|S_0|\,r^2)$. Since l and d are small constants and $|S_0|$ is the number of objective vertices that is fixed, the computational complexity is $O(r^2)$. But sometimes the complexity $O(r^2)$ is also larger than $O(m)$ when the number of edges is small. Thus denoting $\overrightarrow{I_k}$ as $|S_k| \times 1$ vector of all 1s and according to the process in Fig. 1, we can reformulate $\widehat{Katz(i)}$ as:

$$\begin{aligned}
\widehat{Katz(i)} &= \alpha|\mathcal{R}(i)| + \sum_{k=2}^{d} \alpha^k b^{[k]} \\
&= \alpha b^{[1]} + \alpha^2 b^{[2]} + \cdots + \alpha^d b^{[d]} \\
&= \alpha B^{[1]}\overrightarrow{I_1} + \alpha^2 B^{[2]}\overrightarrow{I_2} + \cdots + \alpha^d B^{[d]}\overrightarrow{I_d} \\
&= \alpha B^{(1)}\overrightarrow{I_1} + \alpha^2 B^{(1)}B^{(2)}\overrightarrow{I_2} + \cdots + \alpha^d B^{(1)}B^{(2)}\cdots B^{(d)}\overrightarrow{I_d} \\
&= \alpha B^{(1)}(\alpha B^{(2)}(\alpha B^{(3)}(\cdots(\alpha B^{(d)}\overrightarrow{I_d}) + \cdots + \overrightarrow{I_3}) + \overrightarrow{I_2}) + \overrightarrow{I_1})
\end{aligned} \tag{10}$$

From the equation, we can find that for the estimation of Katz centrality, we just need to each time calculate the product of a matrix and a vector which generally costs $O(r^2)$. What's more when using the Compressed Sparse Row (CSR) matrix data to represent $B^{(\cdot)}$, it only needs considering the non-zero elements in original matrices, namely the edges m in the graph. If we randomly select one vertex respectively from two neighboring sampling process, the probability that there exists one edge between them is $\frac{m}{n^2}$ and the expectation of edges number is also $\frac{m}{n^2}$. Thus if we randomly select r vertices respectively from two neighboring process, the expectation of the edges number is $\frac{m}{n^2} \cdot r^2 = \frac{r^2}{n^2}m$. Therefore, the complex for the first $d-1$ product processes can be further reduced to $O(l(d-1)\frac{r^2}{n^2}m)$. However the last product process costs $O(ls)$ since S_1 with

size s is the successors set of objective vertex. As the consequence, we have the final complexity of $O(l(d-1)\frac{r^2}{n^2}m+ls)$ for the algorithm. The method to reduce complexity can be viewed as a variant of the method from [14], where Foster et al. estimate Katz centrality iteratively by computing partial sums with aggregating Katz values from the successors of vertices. The pseudo-code of **SAKE** is illustrated in Algorithm 1. Given the fact that $r \ll n$ is the size of sampled vertex set and the size of successors set for the objective vertex is generally small, the proposed **SAKE** algorithm reduces the computational complexity of Katz centrality dramatically. Even though the algorithm above only examines starting at a single vertex, our algorithm can be easily adapted to the case starting at multiple vertices. Instead of repeating the process of the algorithm **SAKE** for times independently to evaluate Katz Centralities of multiple objective vertices, we can construct the matrics $B^{(3)}, B^{(4)}, \cdots, B^{(k)}$ only once and reuse them in each process.

Algorithm 1 SAKE(G, r, d, α, i)

Input:
G : The original network
r : Number of vertices to be sampled in each process
d : The maximum number of walks
α : The attenuation factor
i : The objective vertex to estimate Katz centrality
Output:
$\overline{Katz(i)}$: The final estimator of Katz centrality

Let l be the number of loops needed to obtain the desired error bound
for $t = 1, 2 \cdots, l$ **do**
    $Katz[t] =$ **ESTIMATE_KATZ**(G, r, d, α, i)
Let $\overline{Katz(i)} = \frac{\sum_{t=1}^{l} Katz[t]}{l}$
return $\overline{Katz(i)}$

ESTIMATE_KATZ(G, r, d, α, i)

Let $S_0 = \{i\}$
Select all the nodes from $\mathcal{R}(i)$ to form S_1.
for $k = 2, 3 \cdots, d$ **do**
    Randomly sample r nodes from V with probability π to form S_k
$\widehat{Katz} = \overrightarrow{0}$ with shape $|S_d| \times 1$
for $k = d, d-1 \cdots, 1$ **do**
    Construct matrices $B^{(k)}$ based on S_{k-1} and S_k
    Compute $\widehat{Katz} = \alpha B^{(k)}(\widehat{Katz} + \overrightarrow{I_k})$
return $\widehat{Katz}$

We adopt a repeat estimation approach to improve the estimation accuracy. According to Theorem 4.1, the expectation of the mean of the Katz centrality estimates is equal to the true value, therefore we repeat the estimation process for several times and take the mean as the estimation value. When the estimation process is repeated for l times, we will show in theory that by carefully choosing the value of l, the estimation error can be bounded within an extent with high probability.

Next we show that the estimation error of **SAKE** can be bounded to some extent by carefully choosing the number of loops l.

THEOREM 4.2 (HOEFFDING BOUND [18]). *If $x_1, x_2, \cdots, x_k$ are independent random variables, where $a_t \le x_t \le b_t$ $(1 \le t \le k)$, and $\mu = E[\sum_t x_t/k]$ is the expected mean, then for $\xi > 0$*

$$\Pr\{\left|\frac{\sum_{t=1}^{k} x_t}{k} - \mu\right| \ge \xi\} \le 2e^{-2k^2\xi^2/\sum_{t=1}^{k}(b_t-a_t)^2}.$$

The Hoeffding bound provides a probabilistic bound for the mean of independent random variables, which can be applied to compute the error bound of the proposed **SAKE** algorithm as in Theorem 4.3.

THEOREM 4.3. *Given a graph G with n vertices. Assume the **SAKE** algorithm runs l loops to estimate Katz centrality by $\overline{Katz}(i) = \frac{\sum_{t=1}^{l}\widehat{Katz(i)}_t}{l}$, where $\widehat{Katz(i)}_t$ is the Katz estimator in the t-th loop. Let $\Delta = \max_{1\le t\le l}\widehat{Katz(i)}[t]$, and r be the sample size in each sampling process in the sampled graph. Let $\xi = \eta\Delta$ for $\forall\eta > 0$. If the number of loops $l \in \Omega(\frac{\log r}{\eta^2})$, then the estimation results in error $|\overline{Katz}(i) - Katz(i)| \le \xi$ with high probability.*

PROOF. Since Δ is the maximum value in l-time calculations of the estimate $\widehat{Katz(i)}$, the observed Katz centralities satisfy $0 \le \widehat{Katz(i)}_t \le \Delta$ $(t = 1, 2, \cdots, l)$ for all loops. Taking the l estimators as random variables, we can apply the Hoeffding bound with $x_t = \widehat{Katz(i)}_t$, $\mu = Katz(i)$, $a_t = 0$, and $b_t = \Delta$.

Theorem 4.1 has proved that $E(\widehat{Katz(i)}) = Katz(i)$. Thus the probability of the difference between the estimated Katz centrality $\overline{Katz}(i)$ and the actual Katz centrality $Katz(i)$ larger than ξ is:

$$\Pr\{|\overline{Katz}(i) - Katz(i)| \ge \xi\} \le 2e^{-2l^2\xi^2/\sum_{t=1}^{l}(b_t-a_t)^2} = 2e^{-2l\xi^2/\Delta^2}.$$

For $\xi = \eta\Delta$, if the number of loops $l = \frac{\log r}{2\eta^2}$, then

$$\Pr\{|\overline{Katz}(i) - Katz(i)| \ge \xi\} \le \frac{2}{r}.$$

Therefore, if $l \in \Omega(\frac{\log r}{2\eta^2})$, the estimation error is less than ξ with probability $1 - \frac{2}{r}$. For a graph with large number of sampling vertices or with higher sampling ratio, the probability approaches 1 when r increases. This proves the theorem. □

By letting $\epsilon = \xi$ and $\delta = \frac{2}{r}$, it is easy to verify the following theorem.

THEOREM 4.4. *The estimator $\overline{Katz(i)}$ computed by the **SAKE** algorithm is a (ϵ,δ)-approximation of $Katz(i)$.*

4.3 Some other properties

There is something about the distribution of the estimates of Katz centrality that can be identified. That is the distribution of the estimate $\widehat{Katz(i)}$ approaches normal distribution when $|S_k|(k = 1, 2, \cdots, d)$ is large enough. Thus we in the following prove the theorem.

THEOREM 4.5. *The distribution of the estimate $\widehat{Katz(i)}$ approaches normal distribution when using simple random sampling in each sampling procedure and $|S_k|(k = 1, 2, \cdots, d)$ is large enough.*

PROOF. Similar to the proof of Theorem 4.1, we turn to the basic unit $\widehat{D_i^k(G)}$ for $1 \le k \le d$. As Eq. (4), we regard the indicator $A_{iv_1}A_{v_1v_2}\cdots A_{v_{k-1}v_k}$ as an independent variable $p_{iv_1v_2\ldots v_k}$ with value 0 and 1 where 1 means really existing the walk $i \to v_1 \to v_2 \to \cdots \to v_k$ in G and 0

otherwise. When using *simple random sampling*, in each sampling procedure the probability of inclusion $\pi_{v_1 v_2 \dots v_k} = \pi_{v_1}^{S_1} \pi_{v_2}^{S_2} \cdots \pi_{v_k}^{S_k}$ for each walk is equal to $\frac{\prod_{j=1}^{k} |S_j|}{|\mathcal{R}(i)| n^{k-1}}$. Thus if we totally select $\prod_{j=1}^{k} |S_j|$ walks , the mean for these indicators $p_{i v_1 v_2 \dots v_k}$ s is

$$\overline{p_{i v_1 v_2 \dots v_k}} = \frac{\sum_{v_1 \in S_1} \sum_{v_2 \in S_2} \cdots \sum_{v_k \in S_k} p_{i v_1 v_2 \dots v_k}}{\prod_{j=1}^{k} |S_j|} \tag{11}$$

According to classical central limit theorem, $\overline{p_{i v_1 v_2 \dots v_k}}$ approaches normal distribution when the number of selected walks $\prod_{j=1}^{k} |S_j|$ is large enough.

Therefore, when $2 \le k \le d$ the estimate $\widehat{\left|D_i^k(G)\right|}$ in Equation (6) can be derived to:

$$\begin{aligned}
\widehat{\left|D_i^k(G)\right|} &= \sum_{v_1 \in S_1} \sum_{v_2 \in S_2} \cdots \sum_{v_k \in S_k} \frac{p_{i v_1 v_2 \dots v_k}}{\pi_{v_1 v_2 \dots v_k}} \\
&= \frac{\sum_{v_1 \in S_1} \sum_{v_2 \in S_2} \cdots \sum_{v_k \in S_k} p_{i v_1 v_2 \dots v_k}}{(|S_1|/|\mathcal{R}(i)|) \cdot |\prod_{j=2}^{k} (|S_j|/n)} \\
&= |\mathcal{R}(i)| n^{k-1} \cdot \frac{\sum_{v_1 \in S_1} \sum_{v_2 \in S_2} \cdots \sum_{v_k \in S_k} p_{i v_1 v_2 \dots v_k}}{\prod_{j=1}^{k} |S_j|} \\
&= |\mathcal{R}(i)| n^{k-1} \cdot \overline{p_{i v_1 v_2 \dots v_k}}
\end{aligned} \tag{12}$$

In this way, the estimate $\widehat{Katz(i)}$ can be also derived to:

$$\widehat{Katz(i)} = \sum_{k=1}^{d} \alpha^k \widehat{\left|D_i^k(G)\right|} = \alpha |\mathcal{R}(i)| + \frac{|\mathcal{R}(i)|}{n} \sum_{k=2}^{d} (\alpha n)^k \overline{p_{i v_1 v_2 \dots v_k}} \tag{13}$$

For the reason that $\alpha|\mathcal{R}(i)|$, $\frac{|\mathcal{R}(i)|}{n}$ and $(\alpha n)^k$ are constants and each round of sampling is independent, $\widehat{Katz(i)}$ also approaches normal distribution.

□

Building upon Theorem 4.5, we can derive some other properties for loops l by using confidence interval analysis theory. Under some condition of confidence, the width of confidence interval for the expectation of $\widehat{Katz(i)}$ is something to do with the number of samples, namely loops l, the standard deviation σ and the mean value μ for these samples according to the calculation of confidence interval. Therefore when the mean value μ is fixed, if the standard deviation σ gets larger or the number of samples l gets smaller, the length of the interval will get wider. On the other hand, if the we use more samples namely l, the error will become smaller and if we adopt larger number of vertices to be sampled in each process r, we would get smaller standard deviation for the normal distribution and the error will also become smaller. These properties are verified in the following experiments in the next section.

5 PERFORMANCE EVALUATION

In this section we do some empirical experiments to reveal the characteristics of our Algorithms. At first we explore the errors of a set of randomly selected vertices. And then in the second part we illustrate the precision of estimated value for Katz centrality with the parameters l and r changing. Finally, we show the ability of the proposed algorithm to preserve the vertex rankings.

5.1 Datasets

Table 2. Statistics of Datasets

Name	Type	Nodes	Edges
Livemocha [1]	Undirected	104,103	2,193,083
Pokec [37]	Directed	1,632,803	30,622,564
Livejournal [25]	Undirected	5,204,176	49,174,464
Wikipedia [4]	Directed	18,268,992	172,183,984

The experiments are conducted on four real world networks: (1) *Livemocha* [1]: social network of an online language learning community where nodes represent users and edges represent friendships. (2) *Pokec* [37]: Pokec is the most popular online social network in Slovakia where nodes represent users and edges represent relationship. (3) *Livejournal* [25]: This is the social network of LiveJournal users and their connections where nodes represent users and edges represent connections. (4) *Wikipedia* [4]: The network is the hyperlink network of Wikipedia, as extracted in DBpedia. Nodes are pages and edges correspond to hyperlinks.

Before applying these graphs we remove all self-loops in the sampled graphs. What's more we drop all isolated nodes in the sampled graphs for the reason that their Katz centralities are always 0. The statistics of these networks after data cleaning are summarized in Table 2. These datasets including directed and undirected graphs are chosen to represent the typical networks from different domains in real world. Such diversity allows a more comprehensive study of the proposed algorithm in the face of varying network topologies with different types of social relationships.

5.2 Experimental Setup

We firstly present the setup of default system parameters. The number of walks is set to $d = 6$ by default. The attenuation factor $\alpha = \frac{0.85}{\|A\|_2}$ according to the literature [5, 30] where $\|A\|_2$ denotes the 2-norm of matrix A. The number of loops is set to $l = 20$, which is an empirical value from the experiments. We use r_a to denote the sampling ratio for each round, i.e., $r_a = r/n$, where r is the number of vertices to be sampled in each process and n is the size of the original graph. The default sampling ratio is set to 0.01, that is, only 1% of the total vertices are used in each process for Katz centrality estimation. In each configuration, the experiments are repeated for 20 times to obtain the mean value and error bars. We apply Algorithm 1 to estimate the Katz centralities of a set of random vertices in the networks.

For the reason that generally there is obvious big difference among Katz centrality of vertices, it's not so effective to take absolute metrics into consideration. The main metric we use for evaluation is *MRE* (Mean Relative Error). In general, the relative error is defined as the ratio of absolute error to the truth, namely $RE = \frac{|measured\ value - true\ value|}{true\ value}$. Here the true value is the original Katz centrality computed using the whole graph G. The *MRE* is the mean of *RE* for a number of measurements. The experiments are conducted on a personal computer in Python 3.6.5 with Intel Xeon E5-2620 v2 2.10GHz CPU and 64GB memory, running 64-bit CentOS Linux 7.2.

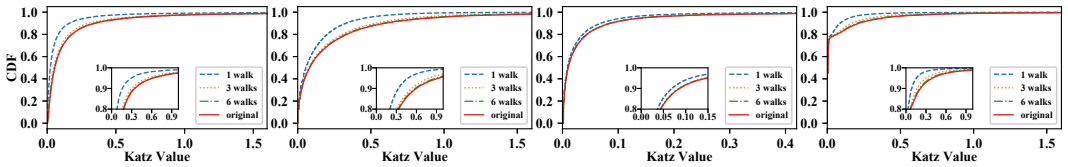
The baseline algorithms we choose to compare with came from the literature [14] by Foster et al. and [30] by Nathan et al. and therefore we refer them as **Foster** and **Nathan** respectively hereinafter:

Foster is widely used in many complex network toolkits such as *NetworkX*[1] and it estimates Katz centrality iteratively through computing partial sums of values from neighbors vertex by vertex and the formulation format is $x_i = \alpha \sum_{j=1}^{n} A_{ij} x_j + 1$. In the algorithm the number of iteration means the maximum walks from objective vertices.

Nathan is an approximate method proposed to calculate personalized Katz centrality using weighted counts of the number of walks from one objective vertex for all the vertices in the graph. It iteratively counts walks from the objective vertex with walks from 1 to d, and then respectively computes the sum of total walks number multiplied by related power of attenuation factor α, which is equal to $Katz(i) = \sum_{k=1}^{d} \alpha^k \left| D_i^k(G) \right|$. This algorithm **Nathan** can only calculate Katz centrality for one vertex at a time.

According the pseudo codes of **Foster** and **Nathan** which have been attached in appendix, they both have two parameters d and α. For **Foster**, d is the number of iterations meaning the maximum walks from source vertices. Thus the parameter is set to the same as the parameter d in **SAKE**. α is the attenuation factor, and its value is set to the same as **SAKE**. Similarly, for algorithm **Nathan**, the maximum walk length d and the attenuation factor α are set to the same value as **SAKE** for fair comparison.

5.3 Numerical Results

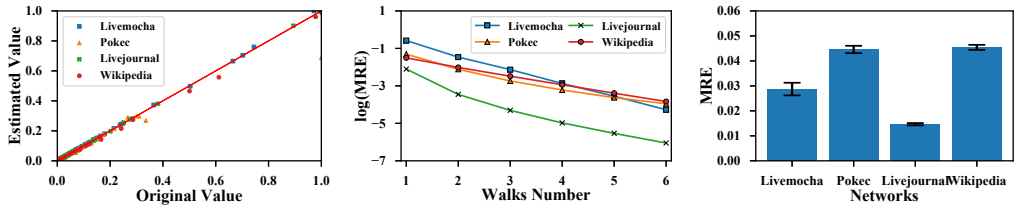


(a) Livemocha (b) Pokec (c) Livejournal (d) Wikipedia

Fig. 2. Cumulative Distribution Function (CDF) of Katz centrality for different number of walks.

5.3.1 Convergence of Katz centrality. We first show how the *truncated Katz centrality* converges to the original Katz centrality when the number of walks d increases. Fig. 2 shows the Cumulative Distribution Function (CDF) of Katz centrality for the number of walks d varying from 1 to 6. We display the situations for four datasets. For convenience, we enlarge a specific part of the figures in order to see the trend for increasing walks clearly. When the number of walks is 1, it's the well known power law distribution for degree centrality. From the four figures, we can see that the CDF of Katz centrality with 1 walk (similar to degree centrality) has visible difference from the Katz with larger walks. More vertex are gathering in low value (0-1) than those with larger walks. But with the number of walks increases, the aggregation interval for Katz values are gradually moving to a larger interval. With the increase of walks, the CDFs are gradually approach to the original Katz, and the differences are getting smaller and smaller. When the walk is 6, we can see that the CDF almost coincides that of the original Katz. The result of these experiments proves our argument that the value of the truncated Katz centrality in Eq. (2) approaches the value of the original Katz centrality in Eq. (1) as d increases, which also ensures our motivation to calculate *truncated Katz centrality*.

[1]https://networkx.github.io



(a) Estimation Deviation (b) log(*MRE*) vs. Number of Walks (c) *MRE*s of four datasets

Fig. 3. Estimation performance for random vertices.

5.3.2 Estimation performance. We show the estimation performance in Fig. 3. Fig. 3(a) compares the deviation between the estimated value and the ground truth in a normalized scale $(0, 1)$. It shows that all points are in or near the diagonal line, which means that the estimation is relatively accurate. Fig. 3(b) shows the $\log(MRE)$s varying with the number of walks d. From the figure, when the walks increases, the $\log(MRE)$s decreases linearly, which means that *MRE* decreases exponentially and as a consequence, the increasing d can improve the estimation accuracy significantly. Fig. 3(c) compares the *MRE*s of estimation in different datasets. It shows that the *MRE* values are below 0.05 for all datasets. The datasets Livejournal has lower *MRE*s around 0.01. According to the error bars in Fig. 3(c), the deviations of errors are small for all datasets.

5.3.3 Comparison with baselines. In this section we report the performance of our algorithm in comparison with two baseline algorithms **Foster** and **Nathan** based on four datasets. We compute the Katz centrality values for top 100 vertices using the three algorithms under varying parameter settings. For the proposed **SAKE** algorithm, we vary the sampling ratio r_a in (0.005, 0.01, 0.015), the maximum walks d in (4, 5, 6), and the loops number l in (10, 15, 20). We compare the running time (unit: second) and accuracy (*MRE*) in this experiment. The results are shown in Table 3.

According to Table 3 , we have the following results.

(1) When comparing the running time, the proposed **SAKE** algorithm is much lower than the baseline algorithms **Foster** and **Nathan** in different parameter settings. The running time of **SAKE** scales linearly with the increasing of d, r_a, l. In most cases, **SAKE** runs 2× to 10× faster than that of **Foster**. The running time of **Nathan** is much longer than both **Foster** and **SAKE**, which is mainly because the algorithm can only calculate Katz centrality for one vertex at a time. Compared with **Nathan**, **SAKE** accelerate the running time up to 100×, which is a great improvement and more efficient in large-scale social networks calculation.

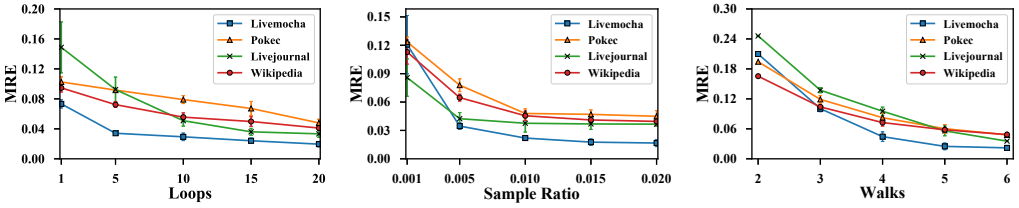
(2) Regarding the comparison of estimation accuracy, the MREs of the three algorithms are close in different parameter settings. The MREs of **Foster** and **Nathan** are the same since they both use the whole graph for estimation. The MREs of **SAKE** decrease when the values of parameters d, r_a, l increase. When parameters l and r_a are small, the MREs of **SAKE** are slightly larger than those of **Foster** and **Nathan**, but when $l = 20$ and $r_a = 0.01$, the accuracy of **SAKE** is comparable with the baselines where the differences are within 0.01. In some cases ($l = 20, r_a = 0.01, d = 4$; $l = 20, r_a = 0.015, d = 4$ and $l = 20, r_a = 0.015, d = 5$ in Livemocha), the MREs of **SAKE** are lower than those of the baselines.

5.3.4 Influence of the parameters. In this section, we will analyze the influence of the parameters d, l, r on the performance of algorithm **SAKE**. The Fig 4 show the performance under varying parameters, note that the default value of them are $l = 20, r_a = 0.01, d = 6$.

Table 3. Comparisons of baseline algorithms in running time and MRE

Livemocha					**Pokec**				
Walks	Algorithms		Running time (s)	MRE	Walks	Algorithms		Running time (s)	MRE
d = 4	Foster		18.704	0.050	d = 4	Foster		201.905	0.076
	Nathan		415.410	0.050		Nathan		1356.767	0.076
	SAKE	l = 20,r_a=0.005	3.697	0.052		SAKE	l = 20,r_a=0.005	21.659	0.088
		l = 20, r_a=0.01	6.405	0.047			l = 20, r_a=0.01	35.830	0.082
		l = 20, r_a=0.015	8.806	0.044			l = 20, r_a=0.015	49.515	0.079
		l = 10, r_a=0.01	3.177	0.069			l = 10, r_a=0.01	12.737	0.131
		l=15, r_a=0.01	4.796	0.055			l=15, r_a=0.01	19.021	0.121
		l=20, r_a=0.01	6.405	0.047			l=20, r_a=0.01	35.830	0.082
d=5	Foster		23.443	0.025	d=5	Foster		244.044	0.052
	Nathan		639.903	0.025		Nathan		4196.175	0.052
	SAKE	l = 20,r_a=0.005	4.157	0.033		SAKE	l = 20,r_a=0.005	26.308	0.076
		l = 20, r_a=0.01	7.341	0.029			l = 20, r_a=0.01	44.418	0.060
		l = 20, r_a=0.015	9.967	0.022			l = 20, r_a=0.015	61.726	0.059
		l = 10, r_a=0.01	3.598	0.039			l = 10, r_a=0.01	23.011	0.080
		l=15, r_a=0.01	5.409	0.028			l=15, r_a=0.01	34.527	0.071
		l=20, r_a=0.01	7.341	0.029			l=20, r_a=0.01	44.418	0.060
d=6	Foster		28.112	0.012	d=6	Foster		290.555	0.040
	Nathan		868.772	0.012		Nathan		7541.538	0.040
	SAKE	l = 20,r_a=0.005	4.621	0.035		SAKE	l = 20,r_a=0.005	31.693	0.078
		l = 20, r_a=0.01	8.121	0.022			l = 20, r_a=0.01	54.236	0.048
		l = 20, r_a=0.015	11.058	0.018			l = 20, r_a=0.015	75.792	0.047
		l = 10, r_a=0.01	4.020	0.031			l = 10, r_a=0.01	28.341	0.079
		l=15, r_a=0.01	6.037	0.023			l=15, r_a=0.01	42.447	0.067
		l=20, r_a=0.01	8.121	0.022			l=20, r_a=0.01	54.236	0.048
Livejournal					**Wikipedia**				
d = 4	Foster		677.250	0.074	d = 4	Foster		1248.857	0.068
	Nathan		6330.910	0.074		Nathan		4061.188	0.068
	SAKE	l = 20,r_a=0.005	79.316	0.103		SAKE	l = 20,r_a=0.005	167.043	0.079
		l = 20, r_a=0.01	129.297	0.086			l = 20, r_a=0.01	252.906	0.078
		l = 20, r_a=0.015	171.935	0.079			l = 20, r_a=0.015	356.900	0.073
		l = 10, r_a=0.01	63.500	0.099			l = 10, r_a=0.01	128.661	0.083
		l=15, r_a=0.01	94.910	0.088			l=15, r_a=0.01	186.060	0.078
		l=20, r_a=0.01	129.297	0.086			l=20, r_a=0.01	252.906	0.078
d=5	Foster		808.693	0.043	d=5	Foster		1485.507	0.045
	Nathan		16316.011	0.043		Nathan		17204.390	0.045
	SAKE	l = 20,r_a=0.005	95.346	0.067		SAKE	l = 20,r_a=0.005	220.062	0.070
		l = 20, r_a=0.01	157.417	0.058			l = 20, r_a=0.01	331.663	0.053
		l = 20, r_a=0.015	210.745	0.048			l = 20, r_a=0.015	461.651	0.052
		l = 10, r_a=0.01	81.493	0.074			l = 10, r_a=0.01	176.424	0.064
		l=15, r_a=0.01	121.867	0.070			l=15, r_a=0.01	263.794	0.058
		l=20, r_a=0.01	157.417	0.058			l=20, r_a=0.01	331.663	0.053
d=6	Foster		939.663	0.026	d=6	Foster		1739.453	0.030
	Nathan		27497.019	0.026		Nathan		36260.968	0.030
	SAKE	l = 20,r_a=0.005	114.723	0.042		SAKE	l = 20,r_a=0.005	268.315	0.065
		l = 20, r_a=0.01	192.266	0.038			l = 20, r_a=0.01	406.204	0.046
		l = 20, r_a=0.015	259.780	0.037			l = 20, r_a=0.015	547.171	0.041
		l = 10, r_a=0.01	100.244	0.063			l = 10, r_a=0.01	219.086	0.056
		l=15, r_a=0.01	150.163	0.040			l=15, r_a=0.01	319.054	0.050
		l=20, r_a=0.01	192.266	0.038			l=20, r_a=0.01	406.204	0.046

Fig. 4(a) shows the performance of **SAKE** with l varying from 1 to 20. With the increasing of loops, the *MRE*s of all datasets declines gradually. It is shown that when $l \geq 5$, the downtrend becomes slow, which implies the algorithm converges. Increasing the value of l can improve the



(a) Parameter l (b) Parameter r_a (c) Parameter d

Fig. 4. Comparison of *MRE* under varying parameters.

estimation accuracy, but the computation cost increases as well. In the experiments, we test l for different values to show the results, and we set $l = 20$ by default unless explicitly specified.

Fig. 4(b) shows the accuracy under different sampling ratios varying from 0.001 to 0.02. When the sampling ratio increases, the value of *MRE* declines gradually at the beginning, and then decreases slowly for sampling ratio larger than 0.005. This implies that sampling a small portion of vertices from the original network yields good approximation result. For larger sampling ratio, some datasets achieve very low *MRE* smaller than 0.03. However, larger sampling ratio will lead to longer running time, therefore we choose sampling ratio r_a as 0.01 for trade-off of efficiency and accuracy.

Fig. 4(c) shows the influence of the maximum number of walks d varying from 2 to 6. Similar with the analysis in Fig. 3(b), when the walks increases, the *MRE*s decreases dramatically. When d increase to 6, the MREs decrease below 0.05. Thus $d = 6$ is a sufficient choice for the algorithm.

5.3.5 Ability of preserving vertex rankings. We further explore the ability of the proposed algorithm to preserve the vertex rankings, i.e., whether the ranking orders of vertices' Katz centralities are preserved in sampling based estimation. Table 4 compares *Jaccard*, *Precision*, *MAP*, and *nDCG*, which are well-known performance metrics in recommendation system [28]. It is shown that the vertices' Katz centrality rankings for top 100 are well preserved with the proposed algorithm. The precision is higher than 97% for all dataset, which means more than 97% high Katz centrality vertices are correctly identified in the top 100 results.

Table 4. The ability of preserving vertex rankings (top 100).

Dataset	Jaccard	Precision	MAP	nDCG
Livemocha	0.970	0.985	0.997	0.997
Pokec	0.980	0.995	0.998	0.998
Livejournal	0.990	1.000	0.998	0.999
Wikipedia	0.951	0.975	0.992	0.992

6 CONCLUSION AND DISCUSSION

Katz centrality is an important concept to measure the influence of a vertex in a social network. In this paper, we combined sampling technique with approximation analysis to develop an algorithm **SAKE**, which can estimate Katz centrality based on a small set of samples from the network. The estimation was proved to be unbiased, and the estimation error could be bounded with high probability. Extensive experiments based on four real social networks showed that **SAKE** achieved

low estimation error and low complexity, and it performed well in identifying the most influential vertices in social networks.

This paper takes the first step towards the design of efficient algorithm for Katz centrality estimation with accuracy guarantee. What's more, the following possible improvements to the current approach may be considered in the future.

- First, the method to estimate Katz centrality is through constructing an unbiased estimator. The estimation relies on d-round random nodes sampling processes which need to examine all links between two vertices sets. One reasonable thought is that we may take the combination of node and edge sampling into consideration in order to reduce the complexity of examining links and find a more concise calculation method. However, designing a hybrid sampling method with theoretical guarantee is difficult. Therefore exploring more efficient sampling method and their theoretical bounds will be our future work.
- Second, most of the time, we want to find a set of high Katz centrality nodes without caring their actual centrality values. Thus a novel research problem is to find the top-k high rank nodes from the social graph regarding their Katz centralities. Our future study will focus on the rank estimation for Katz centrality in large-scale social networks.
- Third, generally speaking Katz centrality is similar to Google's PageRank and to the eigenvector centrality. From the view of vertex-centric heuristic method in [14], Katz centrality can be formulate as $x_i = \alpha \sum_{j=1}^{n} A_{ij} x_j + 1$. While Eigenvector centrality has the similar calculating form as $x_i = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_j$ (λ is eigenvalue) which is also a measure of the influence of a node in a network. It assigns relative scores to all nodes in the network based on the concept that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes. PageRank is more famous than Katz and Eigenvector centralities because of its success in ranking web pages. It has the original formulation of $x_i = \sum_{j=1}^{n} A_{ji} \frac{x_j}{L(j)}$ where $L(j) = \sum_{i=1}^{n} A_{ji}$. We can easily find that these three formulations have similarities in form. Due to the similarities of different centrality measures, the methodology for Katz estimation can also be extended to these centralities in large-scale networks. In the future studies, we aim to develop a more general centrality estimation method based on sampling of adjacency matrix.

ACKNOWLEDGMENT

This work was partially supported by the National Key R&D Program of China (Grant No. 2018YFB 1004704), the National Natural Science Foundation of China (Grant Nos. 61972196, 61672278, 61832008, 61832005, 71772176), the Key R&D Program of Jiangsu Province, China (Grant No. BE2018116), the Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Sino-German Institutes of Social Computing.

REFERENCES

[1] 2017. Livemocha network dataset – KONECT. http://konect.uni-koblenz.de/networks/livemocha
[2] Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. 2010. Reconsidering the foundations of network sampling. In *Proceedings of the 2nd Workshop on Information in Networks (WIN 2010)*.
[3] Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. 2014. Network sampling: From static to streaming graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD 2014)* 8, 2 (2014), 7.
[4] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.
[5] Michele Benzi and Christine Klymko. 2013. Total communicability as a centrality measure. *Journal of Complex Networks* 1, 2 (2013), 124–149.
[6] Paolo Boldi and Sebastiano Vigna. 2014. Axioms for centrality. *Internet Mathematics* 10, 3-4 (2014), 222–262.

[7] Phillip Bonacich. 1972. Factoring and weighting approaches to status scores and clique identification. *Journal of mathematical sociology* 2, 1 (1972), 113–120.

[8] Stephen P Borgatti, Kathleen M Carley, and David Krackhardt. 2006. On the robustness of centrality measures under conditions of imperfect data. *Social networks* 28, 2 (2006), 124–136.

[9] Wei Chen and Shang-Hua Teng. 2017. Interplay between social influence and network centrality: A comparative study on shapley centrality and single-node-influence centrality. In *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. International World Wide Web Conferences Steering Committee, 967–976.

[10] Elizabeth Costenbader and Thomas W Valente. 2003. The stability of centrality measures when networks are sampled. *Social networks* 25, 4 (2003), 283–307.

[11] Eric D. Kolaczyk. 2009. *Statistical Analysis of Network Data: Methods and Models*.

[12] Talya Eden, Shweta Jain, Ali Pinar, Dana Ron, and C Seshadhri. 2018. Provable and Practical Approximations for the Degree Distribution using Sublinear Graph Samples. In *Proceedings of the 27th International Conference on World Wide Web (WWW 2018)*. 449–458.

[13] Roohollah Etemadi and Jianguo Lu. 2017. Bias correction in clustering coefficient estimation. In *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 606–615.

[14] Kurt C Foster, Stephen Q Muth, John J Potterat, and Richard B Rothenberg. 2001. A faster katz status score algorithm. *Computational & Mathematical Organization Theory* 7, 4 (2001), 275–285.

[15] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. 2010. Walking in facebook: A case study of unbiased sampling of osns. In *2010 Proceedings IEEE Infocom*. Ieee, 1–9.

[16] Stephen J Hardiman and Liran Katzir. 2013. Estimating clustering coefficients and size of social networks via random walk. In *Proceedings of the 22nd international conference on World Wide Web (WWW 2013)*. ACM, 539–550.

[17] Douglas D Heckathorn and Christopher J Cameron. 2017. Network sampling: From snowball and multiplicity to respondent-driven sampling. *Annual review of sociology* 43 (2017), 101–119.

[18] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association* 58, 301 (1963), 13–30.

[19] Daniel G Horvitz and Donovan J Thompson. 1952. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association* 47, 260 (1952), 663–685.

[20] Shiyu Ji and Zenghui Yan. 2016. Refining approximating betweenness centrality based on samplings. *arXiv preprint arXiv:1608.04472* (2016).

[21] Leo Katz. 1953. A new status index derived from sociometric analysis. *Psychometrika* 18, 1 (1953), 39–43.

[22] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2003)*. ACM, 137–146.

[23] Sang Hoon Lee, Pan-Jun Kim, and Hawoong Jeong. 2006. Statistical properties of sampled networks. *Physical review E* 73, 1 (2006), 016102.

[24] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proceedings of the 12th international conference on Knowledge discovery and data mining (KDD 2006)*. ACM, 631–636.

[25] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. 2008. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web (WWW 2008)*. ACM, 695–704.

[26] Mingkai Lin, Wenzhong Li, Cam-tu Nguyen, Xiaoliang Wang, and Sanglu Lu. 2019. Sampling Based Katz Centrality Estimation for Large-Scale Social Networks. In *International Conference on Algorithms and Architectures for Parallel Processing(ICA3PP2019)*. Springer, 584–598.

[27] Arun S Maiya and Tanya Y Berger-Wolf. 2011. Benefits of bias: Towards better characterization of network sampling. In *Proceedings of the 17th international conference on Knowledge discovery and data mining (KDD 2011)*. ACM, 105–113.

[28] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering* 16, 1 (2010), 100–103.

[29] Toshiki Matsumura, Kenta Iwasaki, and Kazuyuki Shudo. 2018. Average path length estimation of social networks by random walk. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 611–614.

[30] Eisha Nathan and David A Bader. 2017. Approximating Personalized Katz Centrality in Dynamic Graphs. In *International Conference on Parallel Processing and Applied Mathematics (PPAM 2017)*. Springer, 290–302.

[31] Eisha Nathan, Geoffrey Sanders, James Fairbanks, David A Bader, et al. 2017. Graph ranking guarantees for numerical approximations to katz centrality. *Procedia Computer Science* 108 (2017), 68–78.

[32] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[33] Alireza Rezvanian, Behnaz Moradabadi, Mina Ghavipour, Mohammad Mehdi Daliri Khomami, and Mohammad Reza Meybodi. 2019. Social network sampling. In *Learning Automata Approach for Social Networks*. Springer, 91–149.

[34] Bernardete Ribeiro and Don Towsley. 2012. On the estimation accuracy of degree distributions from graph sampling. 8267, 1 (2012), 5240–5247.
[35] Matthew Richardson and Pedro Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2002)*. ACM, 61–70.
[36] Matteo Riondato and Evgenios M Kornaropoulos. 2016. Fast approximation of betweenness centrality through sampling. *Data Mining and Knowledge Discovery* 30, 2 (2016), 438–475.
[37] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International Scientific Conference and International Workshop Present Day Trends of Innovations*, Vol. 1.
[38] Alexander van der Grinten, Elisabetta Bergamini, Oded Green, David A. Bader, and Henning Meyerhenke. 2018. Scalable Katz Ranking Computation in Large Static and Dynamic Graphs. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA 2018)*. 42:1–42:14.
[39] Claudia Wagner, Philipp Singer, Fariba Karimi, Jürgen Pfeffer, and Markus Strohmaier. 2017. Sampling from Social Networks with Attributes. In *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. 1181–1190.
[40] David Eppstein Joseph Wang. 2006. Fast approximation of centrality. *Graph Algorithms and Applications* 5, 5 (2006), 39.
[41] Tomasz Was and Oskar Skibski. 2018. An Axiomatization of the Eigenvector and Katz Centralities.. In *In Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*.
[42] Jing Zhao, Ting-Hong Yang, Yongxu Huang, and Petter Holme. 2011. Ranking candidate disease genes from gene expression and protein interaction: a Katz-centrality based approach. *PloS one* 6, 9 (2011), e24306.
[43] Lin Zhu, Nicolas A Menzies, Jianing Wang, Benjamin P Linas, Steven M Goodreau, and Joshua A Salomon. 2020. estimation and correction of bias in network simulations based on respondent-driven sampling data. *Scientific reports* 10, 1 (2020), 1–11.

APPENDIX

The pseudocodes of baselines are shown as follows:

Algorithm 2 Foster(G, d, α, S)

Input:
G : The original network
d : The number of iterations
α : The attenuation factor
S : The objective vertices to estimate Katz centralities
Output:
M: The map of estimated Katz centralities

for v in G **do**
    $map[v] = 1$
for $k = 1, 2 \cdots, d$ **do**
    for v in G **do**
        $new\_map[v] = 0$
    for v in G **do**
        for nbr in $successor(v)$ **do**
            $new\_map[v] += map[nbr]$
    for v in G **do**
        $new\_map[v] = new\_map[v] \cdot \alpha + 1$
    $map = new\_map$
for i in S **do**
    $M[i] = map[i]$
return M

Algorithm 3 Nathan(G, d, α, S)

Input:
G : The original network
d : The maximum number of walks
α : The attenuation factor
S : The objective vertices to estimate Katz centralities
Output:
M: The map of estimated Katz centralities

```
1:  for i in S do
2:      walks = n × k array initialized to 0
3:      map[i] = 1
4:      while j < d do
5:          for v in map do
6:              count = map[v]
7:              for nbr in precursor(v) do
8:                  map[nbr] += count
9:          for v in map do
10:             walks[v][j] = map[v]
11:         j += 1
12:     c = 0
13:     for j = 1 : n do
14:         for k = 1 : d do
15:             c += α^(k+1) * walks[j][k]
16:     M[i] = c
17: return M
```