Opinion Dynamics Incorporating Higher-Order Interactions

Zuobai Zhang*† Wanyue Xu*† Zhongzhi Zhang*† ‡¶ and Guanrong Chen§
*School of Computer Science, Fudan University, Shanghai 200433, China
§ Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China
¶Corresponding Author
Email: {17300240035,xuwy, zhangzz}@fudan.edu.cn; eegchen@cityu.edu.hk

***Abstract*—The issue of opinion sharing and formation has received considerable attention in the academic literature, and a few models have been proposed to study this problem. However, existing models are limited to the interactions among nearest neighbors, ignoring those second, third, and higher-order neighbors, despite the fact that higher-order interactions occur frequently in real social networks. In this paper, we develop a new model for opinion dynamics by incorporating long-range interactions based on higher-order random walks. We prove that the model converges to a fixed opinion vector, which may differ greatly from those models without higher-order interactions. Since direct computation of the equilibrium opinions is computationally expensive, which involves the operations of huge-scale matrix multiplication and inversion, we design a theoretically convergence-guaranteed estimation algorithm that approximates the equilibrium opinion vector nearly linearly in both space and time with respect to the number of edges in the graph. We conduct extensive experiments on various social networks, demonstrating that the new algorithm is both highly efficient and effective.**

***Keywords*-Opinion dynamics; social network; computational social science; random walk; spectral graph theory**

I. Introduction

Recent years have witnessed an explosive growth in social media and online social networks, which have increasingly become an important part of our lives [1]. For example, online social networks can increase the diversity of opinions, ideas, and information available to individuals [2]. At the same time, people may use online social networks to broadcast information on their lives and their opinions about some topics or issues to a large audience. It has been reported that social networks and social media have resulted in a fundamental change of ways that people share and shape opinions [3]. Recently, there have been a concerted effort to model opinion dynamics in social networks, in order to understand the effects of various factors on the formation dynamics of opinions [4].

One of the popular opinion dynamics models is the Friedkin-Johnsen (FJ) model [5]. Although simple and succinct, the FJ model can capture complex behavior of real social groups by incorporating French's "theory of social power" [6], and thus has been extensively studied. A sufficient condition for the stability of this standard model was obtained in [7], the average innate opinion was estimated in [8], and the unique equilibrium expressed opinion vector was derived in [8], [9]. Some explanations of this natural model were consequently explored from different perspectives [9], [10]. In addition, some optimization problems [11] for the FJ model were also investigated, such as opinion maximization [12].

Other than studying the properties and interpretations, many extensions or variants of this popular model have been developed [13]. In [11], the impact of susceptibility to persuasion on opinion dynamics were analyzed by introducing a resistance parameter to modify the FJ model. In [14], a varying peer-pressure coefficient was introduced to the FJ model, aiming to explore the role of increasing peer pressure on opinion formation. In [15], the FJ model was augmented to include algorithmic filtering, to analyze the effect of filter bubbles on polarization. Some multidimensional extensions were developed for the FJ model [16], [17], extending the scalar opinion to vector-valued opinions corresponding to several settings, either independent [16] or interdependent [17].

The above related works for opinion dynamic models provide deep insights into the understanding of opinion formulation, since they grasped various important aspects affecting opinion shaping, including individual's attributes, interactions among individuals, and opinion update mechanisms. However, existing models consider only the interactions among the nearest neighbors, neglecting those interactions among second-order, third-order, and higher-order nearest neighbors, in spite of the fact that this situation is commonly encountered in real natural [18] and social [19], [20] networks. For example, in social networks an individual can make use of the local, partial, or global knowledge corresponding to his direct, second-order, and even higher-order neighbors to search for opinions about a concerned issue or to diffuse information and opinions in an efficient way. To date, there still lack a comprehensive higher-order opinion dynamics model on social networks, although it has been observed that long-range non-nearest-neighbor interactions could play a fundamental role in opinion dynamics.

In this paper, we make a natural extension of the classical FJ opinion dynamics model to incorporate the higher-order interactions between individuals and their non-nearest neighbors by leveraging higher-order random walks. We prove that the higher-order model converges to a unique

equilibrium opinion vector, provided that each individual has a non-zero resistance parameter measuring his susceptibility to persuasion. We show that the equilibrium opinions of the higher-order FJ model differ greatly from those of the classical FJ model, demonstrating that higher-order interactions have a significant impact on opinion dynamics.

Basically, the equilibrium opinions of the higher-order FJ model on a graph are the same as those of the standard FJ model on a corresponding dense graph with a loop at each node. That is, at each time step, every individual updates his opinion according to his innate opinion, as well as the currently expressed opinions of his nearest neighbors on the dense graph. Since the transition matrix of the dense graph is a combination of the powers of that on the original graph, direct construction of the transition matrix for the dense graph is computationally expensive. To reduce the computation cost, we construct a sparse matrix, which is spectrally close to the dense matrix, nearly linearly in both space and time with respect to the number of edges on the original graph. This sparsified matrix maintains the information of the dense graph, such that the difference between the equilibrium opinions on the dense graph and the sparsified graph is negligible.

Based on the obtained sparsifed matrix, we further introduce an iteration algorithm, which has a theoretical convergence and can approximate the equilibrium opinions of the higher-order FJ model quickly. Finally, we perform extensive experiments on different networks of various scales, and show that the new algorithm achieves high efficiency and effectiveness. Particularly, this algorithm is scalable, which can approximate the equilibrium opinions of the second-order FJ on large graphs with millions of nodes.

II. Preliminaries

In this section, some basic concepts in graph and matrix theories, as well as the Friedkin-Johnsen (FJ) opinion dynamics model are briefly reviewed.

A. Graphs and Related Matrices

Consider a simple, connected, undirected social network (graph) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, ..., n\}$ is the set of n agents and $\mathcal{E} = \{(i, j)|i, j \in \mathcal{V}\}$ is the set of m edges describing relations among nearest neighbors. The topological and weighted properties of $\mathcal{G}$ are encoded in its adjacency matrix $\boldsymbol{A} = (a_{ij})_{n\times n}$, where $a_{ij} = a_{ji} = w_e$ if i and j are linked by an edge $e = (i, j) \in \mathcal{E}$ with weight w_e, and $a_{ij} = 0$ otherwise. Let $\mathcal{N}_i = \{j|(i, j) \in \mathcal{E}\}$ denote the set of neighbors of node i and $d_i = \sum_{j\in\mathcal{N}_i} w_{ij}$ denote the degree of i. The diagonal degree matrix of graph $\mathcal{G}$ is defined to be $\boldsymbol{D} = \mathrm{diag}(d_1, d_2, ..., d_n)$, and the Laplacian matrix of $\mathcal{G}$ is $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. Let $\boldsymbol{e}_i$ denote the i-th standard basis vector of appropriate dimension. Let $\mathbf{1}$ ($\mathbf{0}$) be the vector with all entries being ones (zeros). Then, it can be verified that $\boldsymbol{L}\mathbf{1} = \mathbf{0}$. The random walk transition matrix for $\mathcal{G}$ is defined as $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}$, which is row-stochastic (i.e., each row-sum equals 1).

B. Friedkin-Johnsen Opinion Dynamics Model

The Friedkin-Johnsen (FJ) model is a classic opinion dynamics model [5]. For a specific topic, the FJ model assumes that each agent $i \in \mathcal{V}$ is associated with an *innate opinion* $s_i \in [0, 1]$, where higher values signify more favorable opinions, and a *resistance parameter* $\alpha_i \in (0, 1]$ quantifying the agent's *stubbornness*, with a higher value corresponding to a lower tendency to conform with his neighbors' opinions. Let $\boldsymbol{x}^{(t)}$ denote the opinion vector of all agents at time t, with element $x_i^{(t)}$ representing the opinion of agent i at that time. At every timestep, each agent updates his opinion by taking a convex combination of his innate opinion and the average of the expressed opinion of his neighbors in the previous timestep. Mathematically, the opinion of agent i evolves according to the following rule:

$$x_i^{(t+1)} = \alpha_i s_i + (1 - \alpha_i)\frac{\sum_{j\in\mathcal{N}_i} w_{ij} \cdot x_j^{(t)}}{d_i}. \tag{1}$$

The evolution rule can be rewritten in matrix form as

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{\Lambda}\boldsymbol{s} + (\boldsymbol{I} - \boldsymbol{\Lambda})\boldsymbol{P}\boldsymbol{x}^{(t)}, \tag{2}$$

where $\boldsymbol{\Lambda}$ denotes the diagonal matrix $\mathrm{diag}(\alpha_1, \alpha_2, ..., \alpha_n)$, and $\boldsymbol{I}$ is the identity matrix.

It has been proved [8] that the above opinion formation process converges to a unique equilibrium $\boldsymbol{z}$ when $\alpha_i > 0$ for all $i \in \mathcal{V}$. The equilibrium vector $\boldsymbol{z}$ can be obtained as the unique fixed point of equation (2), i.e.,

$$\boldsymbol{z} = (\boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{\Lambda})\,\boldsymbol{P})^{-1}\,\boldsymbol{\Lambda}\boldsymbol{s}\,. \tag{3}$$

The ith entry z_i of $\boldsymbol{z}$ is the *expressed opinion* of agent i.

III. Higher-order Opinion Dynamics Model

In this section, we generalize the FJ model to a higher-order setting by using the random walk matrix polynomials describing higher-order random walks.

A. Random Walk Matrix Polynomial

For a network $\mathcal{G}$, its random walk matrix polynomial is defined as follows [21]:

Definition 1: Let $\boldsymbol{A}$ and $\boldsymbol{D}$ be, respectively, the adjacency matrix and diagonal degree matrix of a graph $\mathcal{G}$. For a non-negative vector $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_T)$ satisfying $\sum_{r=1}^{T} \beta_r = 1$, the matrix

$$\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G}) = \boldsymbol{D} - \sum_{r=1}^{T} \beta_r \boldsymbol{D}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r \tag{4}$$

is a T-degree random walk matrix polynomial of $\mathcal{G}$.

The Laplacian matrix $\boldsymbol{L}$ is a particular case of $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$, which can be obtained from $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$ by setting $T = 1$ and $\beta_1 = 1$. In fact, it can be proved that, for any $\boldsymbol{\beta}$, there

always exists a graph $\mathcal{G}'$ with loops, whose Laplacian matrix is $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$, as characterized by the following theorem.

Theorem 1 (Proposition 25 in [21]): The random walk matrix polynomial $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$ is a Laplacian matrix.

Define matrix $\boldsymbol{L}_{\mathcal{G}_r} = \boldsymbol{D} - \boldsymbol{D}\left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r$, which is a particular case of matrix $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$ corresponding to $T = r$ and $\beta_r = 1$. In fact, $\boldsymbol{L}_{\mathcal{G}_r}$ is the Laplacian matrix of graph $\mathcal{G}_r$, constructed from graph $\mathcal{G}$ by performing r-step random walks on graph $\mathcal{G}$. The ij-th element of the adjacency matrix $\boldsymbol{A}_{\mathcal{G}_r}$ for graph $\mathcal{G}_r$ is equal to the product of the degree d_i for node i in $\mathcal{G}$ and the probability that a walker starts from node i and ends at node j after performing r-step random walks in $\mathcal{G}$. Thus, the matrix polynomial $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$ is a combination of matrices $\boldsymbol{L}_{\mathcal{G}_r}$ for $r = 1, 2, ..., T$.

Based on the random walk matrix polynomials, one can define a generalized transition matrix $\boldsymbol{P}^* = \boldsymbol{P}^*_{\boldsymbol{\beta}}$ for graph $\mathcal{G}$ as follows.

Definition 2: Given an undirected weighted graph $\mathcal{G}$ and a coefficient vector $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_T)$ with $\sum_{r=1}^{T} \beta_r = 1$, the matrix

$$\boldsymbol{P}^*_{\boldsymbol{\beta}} = \sum_{r=1}^{T} \beta_r \boldsymbol{P}^r = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G}) \tag{5}$$

is a T-order transition matrix of $\mathcal{G}$ with respect to vector $\boldsymbol{\beta}$. Note that the generalized transition matrix $\boldsymbol{P}^*$ for graph $\mathcal{G}$ is actually the transition matrix for another graph $\mathcal{G}'$.

B. Higher-Order FJ Model

To introduce the higher-order FJ model, first modify the update rule in equation (2) by replacing $\boldsymbol{P}$ with $\boldsymbol{P}^*$. In other words, the opinion vector evolves as follows:

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{\Lambda}\boldsymbol{s} + (\boldsymbol{I} - \boldsymbol{\Lambda})\left[\beta_1 \boldsymbol{P} + ... + \beta_T \boldsymbol{P}^T\right]\boldsymbol{x}^{(t)}. \tag{6}$$

In this way, individuals update their opinions by incorpating those of their higher-order neighborhoods at each timestep. Moreover, by adjusting the coefficient vector $\boldsymbol{\beta}$, one can choose different weights for neighbors of different orders.

C. Convergence Analysis

The higher-order model has a unique equilibrium and will converge to that equilibrium after sufficiently many iterations, as established in the following theorems.

Theorem 2: The higher-order FJ model defined in (6) has a unique equilibrium if $\alpha_i > 0$ for all $i \in \mathcal{V}$.

Theorem 3: If $\alpha_i > 0$ for all $i \in \mathcal{V}$, then the higher-order FJ model converges to its unique equilibrium $\boldsymbol{z}^* = \left(\boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{\Lambda})\boldsymbol{P}^*\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{s}$.

IV. Fast Estimation of Equilibrium Opinions

Since direct computation of $\boldsymbol{P}^*$ is time consuming, here the spectral graph sparsification technique is utilized to obtain an approximation of matrix $\boldsymbol{P}^*$. Then, a fast convergent algorithm is developed to approximate the expressed opinion vector $\boldsymbol{z}^*$, which avoids matrix inverse operation. The pseudocode of this new algorithm is shown in Algorithm 1.

A. Random-Walk Matrix Polynomial Sparsification

Recall the random-walk matrix polynomial sparsification algorithm [21]. For a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, start from an empty graph $\tilde{\mathcal{G}}$ with the same node set $\mathcal{V}$ and an empty edge set. Then add M edges into the sparsifier $\tilde{\mathcal{G}}$ iteratively by a sampling technique. At each iteration, randomly pick an edge $e = (u, v)$ from $\mathcal{E}$ as an intermediate edge and an integer r from $\{1, 2, ..., T\}$ as the length of the random-walk path. To this end, run the PathSampling(e, r) algorithm in [21] to sample an edge by performing r-step random walks, and add the sample edge, together with its corresponding weight, into the sparsifier $\tilde{\mathcal{G}}$. Note that multiple edges will be merged into one single edges by summing up their weights together. Finally, the algorithm generates a sparsifier $\tilde{\mathcal{G}}$ for the original graph $\mathcal{G}$ with no more than M edges.

In [21], an algorithm is designed to obtain a sparsifier $\tilde{\mathcal{G}}$ with $O(n\epsilon^{-2}\log n)$ edges for $\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G})$, which consists of two steps: The first step uses random walk path sampling to get an initial sparsifier with $O(Tm\epsilon^{-2}\log n)$ edges. The second step utilizes the standard spectral sparsification algorithm proposed in [22] to further reduce the edge number to $O(n\epsilon^{-2}\log n)$. Since a sparsifier with $O(Tm\epsilon^{-2}\log n)$ edges is sparse enough for the present purposes, only the first step will be taken, while skipping the second step, to avoid unnecessary computations.

Algorithm 1: HODynamic$(\mathcal{G}, M, \boldsymbol{s}, \boldsymbol{\beta}, t)$

Input : $\mathcal{G}$: a connected undirected graph;
M: the number of edges in sparsifier;
$\boldsymbol{s}$: the innate opinion vector;
$\boldsymbol{\beta}$: the coefficient vector of random walk matrix polynomial;
t: the number of iterations;

Output : $\tilde{\boldsymbol{x}}^{(t)}$: the approximate equilibrium vector;

1 $\tilde{\mathcal{G}} = (\mathcal{V}, \emptyset)$
2 **for** $i = 1$ *to* M **do**
3 Randomly pick an edge $e = (u, v) \in \mathcal{E}$
4 Select an integer r from $\{1, 2, ..., T\}$ at uniform as the length of the random-walk path
5 Randomly pick an integer $k \in \{1, 2, ..., r\}$
6 Perform $(k-1)$-step random walk from u to u_0
7 Perform $(r-k)$-step random walk from v to u_r
8 Calculate $Z(\boldsymbol{p})$ along the length-r path $\boldsymbol{p}$ between node u_0 and node u_r according to (7)
9 Add an edge (u_0, u_r) of weight $\frac{2rm\beta_r}{MZ}$ to $\tilde{\mathcal{G}}$
10 $\tilde{\boldsymbol{P}} = \boldsymbol{I} - \boldsymbol{D}^{-1}\tilde{\boldsymbol{L}}(\tilde{\mathcal{G}})$
11 $\tilde{\boldsymbol{x}}^{(0)} = \boldsymbol{s}$
12 **for** $i = 1$ *to* t **do**
13 $\tilde{\boldsymbol{x}}^{(i)} = \boldsymbol{\Lambda}\boldsymbol{s} + (\boldsymbol{I} - \boldsymbol{\Lambda})\tilde{\boldsymbol{P}}\tilde{\boldsymbol{x}}^{(i-1)}$
14 **return** $\tilde{\boldsymbol{x}}^{(t)}$

To sample an edge by performing r-step random walks, the procedure of PathSampling algorithm in [21] is characterized in Lines 5-9 of Algorithm 1. To sample an edge,

first draw a random integer k from $\{1, 2, ..., r\}$ and then perform, respectively, $(k-1)$-step and $(r-k)$-step walks starting from two end nodes of the edge $e = (u, v)$. This process samples a length-r path $\boldsymbol{p} = (u_0, u_1, ..., u_r)$. At the same time, compute

$$Z(\boldsymbol{p}) = \sum_{i=1}^{r} \frac{2}{a_{u_{i-1},u_i}}. \tag{7}$$

The algorithm returns the two endpoints of path $\boldsymbol{p}$ as the sample edge (u_0, u_r) and the quantity $Z(\boldsymbol{p})$ for the calculation of weight.

Theorem 4: For a graph $\mathcal{G}$ with random-walk matrix polynomial

$$\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G}) = \boldsymbol{D} - \sum_{r=1}^{T} \beta_r \boldsymbol{D} \left(\boldsymbol{D}^{-1}\boldsymbol{A}\right)^r, \tag{8}$$

where $\sum_{r=1}^{T} \beta_r = 1$ and β_r are non-negative, one can construct, in time $O(T^2 m \epsilon^{-2} \log^2 n)$, a $(1+\epsilon)$-spectral sparsifier, $\tilde{\boldsymbol{L}}$, with $O(n\epsilon^{-2}\log n)$ non-zeros.

Now one can approximate the generalized transition matrix using the Laplacian $\tilde{\boldsymbol{L}}(\tilde{\mathcal{G}})$ of the sparse graph $\tilde{\mathcal{G}}$:

$$\boldsymbol{P}^* = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{L}_{\boldsymbol{\beta}}(\mathcal{G}) \approx \boldsymbol{I} - \boldsymbol{D}^{-1}\tilde{\boldsymbol{L}}(\tilde{\mathcal{G}}) = \tilde{\boldsymbol{P}}^*. \tag{9}$$

Therefore, it takes $O(MT\log n)$ time and $O(M)$ space to obtain the sparsifier $\tilde{\mathcal{G}}$, which is computable for appropriate size M.

B. Approximating the Equilibrium Opinions via Iteration

With the spectral graph sparsification technique, it is possible to approximate $\boldsymbol{P}^*$ with a sparse matrix. Nevertheless, directly computing the equilibrium still involves a matrix inverse operation, which is computationally expensive for large networks, such as those with millions of nodes. To approximate the equilibrium vector $\boldsymbol{z}^*$ using the recurrence defined in (6) and multiple iterations, in this section, we develop a convergent approximation algorithm. The approximation error of this algorithm can be summarized as stated in the following theorem.

Theorem 5 (Approximation Error): For every $t \geq 0$,

$$\left\| \tilde{\boldsymbol{x}}^{(t)} - \boldsymbol{z}^* \right\|_\infty \leq \frac{4\epsilon\sqrt{n} \cdot (1-\alpha_{\min}) \left[1 - (1-\alpha_{\min})^t\right] + (1-\alpha_{\min})^t}{\alpha_{\min}},$$

where $\alpha_{\min} = \min_{i=1,2,..,n}\{\alpha_i\}$.

In the sequel, this approximate iteration algorithm is referred to as APPROX. It should be mentioned that Theorem 5 provides only a rough upper bound. The experiments in Section V-C show that APPROX works well in practice, leading to very accurate results for real networks.

Table I
STATISTICS OF REAL NETWORKS USED IN EXPERIMENTS AND COMPARISON OF RUNNING TIME (SECONDS, s) BETWEEN EXACT AND APPROX FOR THE UNIFORM INNATE OPINION DISTRIBUTION AND MEAN ABSOLUTE ERRORS($\times 10^{-3}$).

Network	n'	m'	Running time (s)		Error
			EXACT	APPROX	
HamstersterFriends	1788	12476	0.174	0.974	3.718
HamstersterFull	2000	16098	0.303	1.540	3.134
PagesTVshow	3892	17239	1.204	1.530	4.552
Facebook (NIPS)	4039	88234	1.492	6.274	1.967
PagesGovernment	7057	89429	5.857	7.679	2.622
Anybeat	12645	49132	31.448	4.730	5.896
PagesCompany	14113	52126	39.348	4.269	4.788
Gplus	23613	39182	163.525	4.329	7.152
GemsecRO	41773	125826	885.069	15.758	4.787
GemsecHU	47538	222887	946.399	28.592	3.611
PagesArtist	50515	819090	1160.469	139.565	2.746
Brightkite	56739	212945	1913.246	27.351	5.717
Livemocha*	104103	2193083	—	538.730	—
Douban*	154908	327162	—	44.166	—
Gowalla*	196591	950327	—	138.222	—
TwitterFollows*	404719	713319	—	96.850	—
Delicious*	536108	1365961	—	209.371	—
YoutubeSnap*	1134890	2987624	—	663.090	—
Hyves*	1402673	2777419	—	648.906	—

V. EXPERIMENTS ON REAL NETWORKS

In this section, we conduct extensive experiments on real-world social networks to evaluate the performance of the algorithm APPROX.

A. Setup

Machine Configuration and Reproducibility. Our extensive experiments run on a Linux box with 16-core 3.00GHz Intel Xeon E5-2690 CPU and 64GB of main memory. All algorithms are programmed in *Julia v1.3.1*. The source code is publicly available at https://github.com/HODynamic/HODynamic.

Datasets. We test the algorithm on a large set of realistic networks, all of which are collected from the Koblenz Network Collection [23] and Network Repository [24]. For those networks that are disconnected originally, we perform experiments on their largest connected components. The statistics of these networks are summarized in the first three columns of Table I, where we use n' and m' to denote, respectively, the numbers of nodes and edges in their largest connected components. The smallest network consists of $1,788$ nodes, while the largest network has more than one million nodes. In Table I, the networks are listed in an increasing order of the number of nodes in their largest connected components.

Input Generation. For each dataset, we use the network structure to generate the input parameters in the following way. The innate opinions are generated according to the uniform distribution. For the uniform distribution, we generated the opinion s_i of node i at random in the range of $[0, 1]$. We generate the resistance parameters uniformly to be within the interval $(0, 1)$.

B. Comparison between Standard FJ Model and Second-Order FJ Model

To show the impact of higher-order interactions on the opinion dynamics, we compare the equilibrium expressed opinions between the second-order FJ model and the standard FJ model on four real networks: PagesTVshow, PagesCompany, Gplus, and GemsecRO. For both models, we generate innate opinions and resistance parameters for each node according to the uniform distribution. We set $\beta_1 = 1, \beta_2 = 0$ for the standard FJ model, and $\beta_1 = 0, \beta_2 = 1$ for the second-order FJ model.

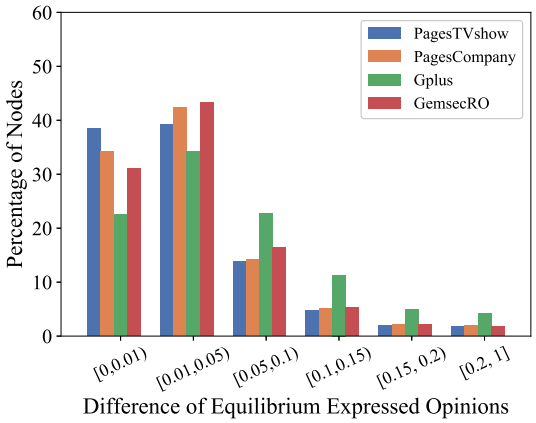

Figure 1. Distribution for difference of equilibrium expressed opinions between the standard FJ model and the second-order FJ model on four real networks.

Figure 1 illustrates the distribution for the difference of the final expressed opinions for each node between the classic and second-order FJ models on four considered real networks. It can be observed that for each of these four networks, there are more than half nodes, for which the difference of expressed opinions between the two models is larger than 0.01. Particularly, there are over 10% agents, for which the difference of equilibrium opinions is greater than 0.1. This possibly makes them stand on the opposite sides for different models. Thus, the opinion dynamics for the second-order FJ model differs largely from the classic FJ model, indicating that the effects of higher-order interactions are not negligible.

C. Performance Evaluation

To evaluate the performance of the new algorithm APPROX, we implement it on various real networks and compare the running time and accuracy of APPROX with those corresponding to the standard EXACT algorithm. For the EXACT, it computes the equilibrium vector by calculating the random-walk matrix polynomials via matrix multiplication and directly inverting the matrix $\boldsymbol{I} - (\boldsymbol{I} - \boldsymbol{\Lambda})\boldsymbol{P}^*$. Here, we use the second-order random-walk matrix polynomial to simulate the opinion dynamics with $\beta_1 = \beta_2 = 0.5$. For APPROX, we set the number M of samples as $10 \times T \times m$ and approximate the equilibrium vector with 100 iterations. To objectively evaluate the running time, we enforce the program to run on a single thread for both EXACT and APPROX on all considered networks, except the last seven marked with asterisks, for which we cannot run EXACT due to the very high cost for space and time.

Efficiency. We present the running time of algorithms APPROX and EXACT for all networks in Table I. For the last seven networks, we only run algorithm APPROX since Exact would take unfeasibly long time. For each of the three innate opinion distributions in different networks, we record the running time of APPROX and EXACT. From Table I, we observe that for small networks with less than 10,000 nodes, the running time of APPROX is a little longer than that of EXACT. Thus, APPROX shows no superiority for small networks. However, for those networks having more than twenty thousand nodes, APPROX significantly improves the computation efficiency compared with EXACT. For example, for the moderately large network GemsecRO with 41,773 nodes, APPROX is $60\times$ faster than EXACT. Finally, for large graphs APPROX shows a very obvious efficiency advantage. Table I indicates that for those network with over 100 thousand nodes, APPROX completes running within 12 minutes, whereas EXACT fails to run. We note that for large networks, the running time of APPROX grows nearly linearly with respect to m', consistent with the above complexity analysis, while the running time of EXACT grows as a cube power of n'.

Accuracy. In addition to the high efficiency, the new algorithm APPROX provides a good approximation for the equilibrim opinion vector $\boldsymbol{z}^* = (z_1^*, z_2^*, ..., z_{n'}^*)^\top$ in practice. To show this, we compare the approximate results of APPROX for second-order FJ model with exact results obtained by EXACT, for all the examined networks shown in Table I, except the last seven which are too big for EXACT to handle. For each of the three distributions of the innate opinions, Table I reports the mean absolute error $\sigma = \sum_{i=1}^{n'} |z_i^* - \tilde{z}_i^*|/n'$, where $\tilde{\boldsymbol{z}}^* = (\tilde{z}_1^*, \tilde{z}_2^*, ..., \tilde{z}_{n'}^*)^\top$ is the estimated vector obtained by APPROX. From Table I, we observe that the actual mean absolute errors σ are all less than 0.008, thus ignorable. Furthermore, for all networks we tested, the mean absolute errors σ are smaller than the theoretical ones provided by Theorem 5. Therefore, the new algorithm APPROX provides a very desirable approximation for the equilibrium opinion vector in applications.

VI. CONCLUSION

In this paper, we presented a significant extension of the classic Friedekin-Johnsen (FJ) model by considering not only nearest-neighbor interactions, but also long-range interactions via leveraging higher-order random walks. We showed that the proposed model has a unique equilibrium expressed opinion vector, provided that each individual

holds an innate opinion. We also demonstrated that the resultant expressed opinion vector of the new model may be significantly different from that of the FJ model, indicating the important impact of higher-order interactions on opinion dynamics.

The expressed opinion vector of the new model can be considered as an expressed opinion vector of the FJ model in a dense graph with a loop at every node, which has the transition matrix as a combination of powers of the transition matrix for the original graph. However, direct computation of the transition matrix for the dense graph is computationally expensive, which involves multiple matrix multiplication and inversion operations. As a remedy, we presented a nearly linear-time algorithm to obtain a sparse matrix, which is spectrally similar to the original dense matrix thereby presevering all basic information. Based on the obtained sparse matrix, we further proposed a convergent iteration algorithm, which approximates the equilibrium opinion vector in linear space and time. We finally conducted extensive experiments on diverse social networks, which demonstrate that the new algorithm achieves both good efficiency and effectiveness.

ACKNOWLEDGMENT

This work was in part supported by the Shanghai Municipal Science and Technology Major Project (No. 2018SHZDZX01), ZJLab, and the National Natural Science Foundation of China (No. 61872093).

REFERENCES

[1] K. P. Smith and N. A. Christakis, "Social networks and health," *Ann. Rev. Soc.*, vol. 34, no. 1, pp. 405–429, 2008.

[2] J. K. Lee, J. Choi, C. Kim, and Y. Kim, "Social media, network heterogeneity, and opinion polarization," *J. Commun.*, vol. 64, no. 4, pp. 702–722, 2014.

[3] A. Das, S. Gollapudi, and K. Munagala, "Modeling opinion dynamics in social networks," in *WSDM*. ACM, 2014, pp. 403–412.

[4] B. D. Anderson and M. Ye, "Recent advances in the modelling and analysis of opinion dynamics on influence networks," *Int. J. Autom. Comput.*, vol. 16, no. 2, pp. 129–149, 2019.

[5] N. E. Friedkin and E. C. Johnsen, "Social influence and opinions," *J. Math. Soc.*, vol. 15, no. 3-4, pp. 193–206, 1990.

[6] J. R. French Jr, "A formal theory of social power." *Psych. Rev.*, vol. 63, no. 3, pp. 181–194, 1956.

[7] C. Ravazzi, P. Frasca, R. Tempo, and H. Ishii, "Ergodic randomized algorithms and dynamics over networks," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 2, pp. 78–87, 2015.

[8] A. Das, S. Gollapudi, R. Panigrahy, and M. Salek, "Debiasing social wisdom," in *KDD*. ACM, 2013, pp. 500–508.

[9] D. Bindel, J. Kleinberg, and S. Oren, "How bad is forming your own opinion?" *Games Econ. Behav.*, vol. 92, pp. 248–265, 2015.

[10] J. Ghaderi and R. Srikant, "Opinion dynamics in social networks with stubborn agents: Equilibrium and convergence rate," *Automatica*, vol. 50, no. 12, pp. 3209–3215, 2014.

[11] R. Abebe, J. Kleinberg, D. Parkes, and C. E. Tsourakakis, "Opinion dynamics with varying susceptibility to persuasion," in *KDD*. ACM, 2018, pp. 1089–1098.

[12] A. Gionis, E. Terzi, and P. Tsaparas, "Opinion maximization in social networks," in *ICDM*. SIAM, 2013, pp. 387–395.

[13] P. Jia, A. MirTabatabaei, N. E. Friedkin, and F. Bullo, "Opinion dynamics and the evolution of social power in influence networks," *SIAM Rev.*, vol. 57, no. 3, pp. 367–397, 2015.

[14] J. Semonsen, C. Griffin, A. Squicciarini, and S. Rajtmajer, "Opinion dynamics in the presence of increasing agreement pressure," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1270–1278, 2019.

[15] U. Chitra and C. Musco, "Analyzing the impact of filter bubbles on social network polarization," in *WSDM*. ACM, 2020, pp. 115–123.

[16] N. E. Friedkin, "The problem of social control and coordination of complex systems in sociology: A look at the community cleavage problem," *IEEE Control Syst. Mag.*, vol. 35, no. 3, pp. 40–51, 2015.

[17] N. E. Friedkin, A. V. Proskurnikov, R. Tempo, and S. E. Parsegov, "Network science on belief system dynamics under logic constraints," *Science*, vol. 354, no. 6310, pp. 321–326, 2016.

[18] M. Schunack, T. R. Linderoth, F. Rosei, E. Lagsgaard, I. Stensgaard, and F. Besenbacher, "Long jumps in the surface diffusion of large molecules," *Phys. Rev. Lett.*, vol. 88, no. 15, p. 156102, 2002.

[19] D. Brockmann, L. Hufnagel, and T. Geisel, "The scaling laws of human travel," *Nature*, vol. 439, pp. 462–465, 2006.

[20] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the Levy-walk nature of human mobility," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 630–643, 2011.

[21] D. Cheng, Y. Cheng, Y. Liu, R. Peng, and S.-H. Teng, "Efficient sampling for Gaussian graphical models via spectral sparsification," in *COLT*, 2015, pp. 364–390.

[22] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM J. on Comput.*, vol. 40, no. 6, pp. 1913–1926, 2011.

[23] J. Kunegis, "Konect: The koblenz network collection," in *WWW*, ACM. New York, USA: ACM, 2013, pp. 1343–1350.

[24] R. Rossi and N. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*. AAAI, 2015, pp. 4292–4293.