Received February 2, 2020, accepted February 27, 2020, date of publication March 9, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979163

Joint Learning of Embedding-Based Parent Components and Information Diffusion for Social Networks

QING BAO[1], WILLIAM K. CHEUNG[2], BENYUN SHI[3], HONGJUN QIU[1], AND LIJIA MA[4]

[1]School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[2]Department of Computer Science, Hong Kong Baptist University, Hong Kong
[3]School of Computer Science and Technology, Nanjing Tech University, Nanjing 211800, China
[4]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding author: Qing Bao (qbao@hdu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806061 and Grant 61672358, in part by the General Research Fund from the Research Grants Council of Hong Kong under Grant 12202117, and in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ19F030011.

ABSTRACT Diffusion on social networks refers to the phenomena that opinions are spread via connected nodes. Given a set of observed cascades, the underlying diffusion process can be inferred for social network analysis. Earlier studies for modeling the diffusion process often assume that the activation of a node depends independently on the activations of its neighbors (or called parent nodes). Nevertheless, the activation of a node also depends on the connectivity of its neighbors. For instance, the opinions from the neighbors of the same closely connected social group are often similar, and thus those neighbors exhibit similar influence. Some recent studies incorporate the structural dependency of neighbors as connected components, which allow more accurate diffusion models to be inferred. However, the effectiveness of such component-based models often depends on how the components are identified. Existing methods are not designed to directly preserve the local connectivity of neighbors. In this paper, we propose to incorporate network embedding to enhance the performance of component-based diffusion models in social networks. In particular, we embed nodes in a social network in a latent vector space with local connectivity of the nodes preserved. Parent component identification then becomes a clustering task in the embedding space. A united probabilistic framework is proposed so that the parent components and the component-based diffusion models can be inferred simultaneously using a two-level EM algorithm based on observed information cascades. For performance evaluation, we apply the proposed model to both synthetic and real world data sets with promising results obtained. The empirical results also show how the use of the embedding-based framework can enhance both the component identification and the diffusion model.

INDEX TERMS Diffusion networks, independent cascade model, social networks, local connectivity information, network embedding.

I. INTRODUCTION

Diffusion on social networks refers to the phenomena that actions or information spreads among connected nodes, resulting in information cascades. Given a set of observed information cascades, the underlying diffusion process can be inferred [1], [2] for social network analysis, such as influence maximization [3]–[5], personalized recommendation [6], [7], authoritative user identification [8], and *et al.*

How actions or information spreads is well known to be highly related to the interactions of connected nodes. Various diffusion models have been proposed [1], [2], [9], [10] in the literature. Among them, the independent cascade (IC) model [1] and the linear threshold (LT) model [2] are two widely used diffusion models. The IC model assumes that a node can be influenced by any of its neighbors independently with some chosen probability, while the LT model assumes that whether a node will be influenced requires the social affirmation from multiple neighbors. Here we focus on the IC model and its extensions for more accurate modeling of

The associate editor coordinating the review of this manuscript and approving it for publication was Jeonghwan Gwak.

diffusion processes. Since the basic IC model was first proposed, there have been different variants of the IC model to uncover temporal dynamics [11], take continuous time [12], consider topic-aware [13] and role-aware [14], [15] diffusion, incorporate node embeddings [16] and *et al.*

The long-standing framework associates the probability of adopting a behavior with the number of network neighbors who have already adopted the behavior [17], [18]. Nevertheless, in social networks, the network neighbors could exhibit different forms of influence depending on their connectivity. For example, neighbors of the same social group are closely connected and communicate through their connectivity, thus those neighbors often have similar opinions and exhibit similar influence. In particular, Ugander *et al.* observed that the user engagement behavior in Facebook was affected by the connected components of Facebook users in the contact neighborhood instead of the individual users [19]. In addition, Zhang *et al.* proposed a related notion called social influence locality for modeling retweeting behaviors [20]. The structure of neighbors has been regarded as the resources they hold, which is termed social capital in [21]. Related perspectives have also been explored for mobile communication network and social network analysis [22], [23]. Based on that, Bao *et al.* [24] first proposed a component-based diffusion model for social networks, which assumes that the influence of the neighbors is not exerted individually but by connected components of the neighbors (called parent components in [24]). However, the effectiveness of component-based models often depends on how the components of the neighbors are identified. In [24], the components in the contact neighborhood are assumed to form different communities and are detected with community detection algorithms by optimizing quantities such as modularity [25]. As those quantities not designed to directly preserve local connectivity information among neighbor nodes, not all nodes in the detected components are closely connected.

In this paper, we propose a component-based diffusion model with *local connectivity information* of neighbors preserved. Recently, network embedding approaches have been proposed to encode individual nodes as low-dimensional embedding vectors that summarize the rich network properties [26]–[28], such as network structure [29]–[31], node content [32], [33], heterogeneous information networks [34], [35], temporal networks [27], [36], dynamics on the network [28] and *et al.* The learnt node embeddings can be used as feature inputs for downstream machine learning tasks such as node classification, node clustering and community detection. Among them, LINE [37], DeepWalk [29] and Node2vec [30] are commonly adopted to preserve the neighborhood structure of nodes via optimizing local connectivity information. The resulting node embeddings make the connected nodes near in the embedding space. In the context of component-based diffusion model, given the learnt node embeddings, parent component detection can then become a clustering task in the embedding space and probabilistic clustering algorithms such as Gaussian Mixture Models (GMM) can be naturally adopted. This further hints that the component detection and diffusion network inference problems both can readily be integrated under a unified probabilistic framework. We here formulate *the component identification* and *diffusion network inference* problems based on an overall likelihood function so that both can be solved at the same time in a disciplined manner. The corresponding learning algorithm is derived and the effectiveness of our proposed model is evaluated using both synthetic and real data sets. The results show the use of the embedding-based framework can enhance both component identification and diffusion prediction, and our model can be applied to support dependency analysis of different online news media.

The contributions of this paper are summarized as follows.

1) We model the *local connectivity information* of neighbors using network embedding to enhance both component identification and diffusion network inference. In particular, we propose a component-based diffusion model with embedding-based parent components, and formulate the *component identification* and *diffusion network inference* problems based on an overall likelihood function. To the best of our knowledge, this is the first work to solve both problems simultaneously within a unified framework.
2) We make use of the expectation maximization (EM) [38] algorithm and derive a corresponding two-level EM algorithm for obtaining maximum likelihood (ML) estimates of the model parameters. Both parent components and underlying diffusion network can be inferred simultaneously based on the observed information cascades.

The remainder of this paper is organized as follows. Section II gives the detailed formulation of our proposed model, followed by the EM algorithm for learning the model. Experimental results and related discussion can be found in Section III. Section IV concludes the paper and provides pointers for future work.

II. FORMULATION

In this section, we propose a novel component-based diffusion model where the parent components are identified with local connectivity information well preserved, with the objective to better model the underlying information spreading process. We first apply node embedding to the neighborhood network of each node to obtain the parent node representations. We then model the communities of parents for each node based on their node representations using Gaussian Mixture Models. Each mixture component tries to capture a community, and then the component-based diffusion model is inferred accordingly. We formulate it as a maximum likelihood estimation problem so that the components and the diffusion model can be inferred simultaneously given sufficient observed cascades.

A. PRELIMINARY

We represent a social network as a directed graph $G = (V, E)$ where V is the set of nodes and E is the set of edges.

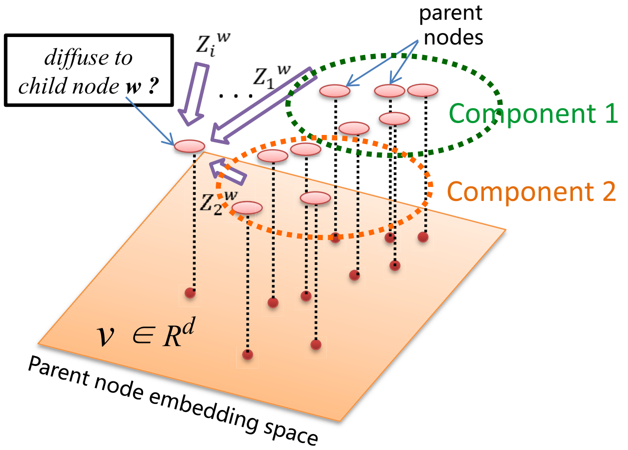


FIGURE 1. **Illustration of our proposed component-based model.**

Let $e = (v, w)$ be an edge from node v to node w, and $f(w)$ and $b(w)$ be the sets of child nodes and parent nodes of node w respectively, given as: $f(w) = \{u : (w, u) \in E\}$ and $b(w) = \{v : (v, w) \in E\}$. Also, we denote by $D_s = \{D_s(0), D_s(1) \cdots D_s(T_s)\}$ the s^{th} observed information cascade where $D_s(t)$ is the set of nodes activated at time step t and T_s is the final time step for the cascade D_s.

B. PARENT COMPONENTS BASED ON LOCAL CONNECTIVITY INFORMATION

For learning the embeddings of parent nodes, we consider neighborhood network of each child node w and adopt the widely used node2vec algorithm [30]. The node2vec algorithm introduces biased random walks on a graph to optimize the sampling strategies so that nonlinear graph structure can be turned into linear sequences for learning embeddings under the Skip-gram architecture [39], [40]. We assign each parent node $v \in b(w)$ with a continuous real-valued K dimension vector $\boldsymbol{v}^{\boldsymbol{w}}$. For each node $w \in V$, we define $N_S(w) \subset V$ as a *network neighborhood* of node w that are nearby in the generated sequence through the sampling strategies S. Given the generated sequences by the random walks, we seek to optimize the following objective function, which maximizes the log-probability of observing network neighborhood $N_S(w)$ of a node w conditioned on its representation, give by f:

$$\max_{f} \sum_{w \in V} \log P(N_S(w)|f(w))$$

Under the Skip-gram architecture, we can approximate it using negative sampling. We optimize the objective function using the stochastic gradient ascent over the model parameters defining the embedding.

Based on the learnt parent node embeddings for each child node w, parent component identification essentially becomes a clustering task in the embedding space, and here we adopt the Gaussian mixture model. But note that we propose to do the "clustering" and the diffusion network inference together under a unified probabilistic framework to be detailed in the sequel. For each node $w \in V$, we assume the presence of $N_z(w)$ different *latent components* consisting of w's parent nodes $b(w)$, which will make node w likely to be activated. We denote by $z^w \in \{1, \ldots, N_z(w)\}$ the index of the latent components, and model the set of parent components of a child node w as a mixture of Gaussians. We use a Gaussian distribution $\mathcal{N}(\boldsymbol{v}^{\boldsymbol{w}}|\boldsymbol{\mu}_{\boldsymbol{z}^{\boldsymbol{w}}}, \boldsymbol{\Sigma}_{\boldsymbol{z}^{\boldsymbol{w}}})$ to model each component z^w, and each component z^w has a mixing proportion α_{z^w} where $\sum_{z^w=1}^{N_z(w)} \alpha_{z^w} = 1$ and $\forall \alpha_{z^w}(\alpha_{z^w} \geq 0)$. Thus, the probability of observing a parent node $v \in b(w)$ of child node w in the embedding space is given as

$$p(v^w) = \sum_{z^w=1}^{N_z(w)} \alpha_{z^w} \mathcal{N}(\boldsymbol{v}^{\boldsymbol{w}}|\boldsymbol{\mu}_{\boldsymbol{z}^{\boldsymbol{w}}}, \boldsymbol{\Sigma}_{\boldsymbol{z}^{\boldsymbol{w}}})$$

And thus the likelihood of observing the parent components for all the nodes $w \in V$ is given as

$$L_1(\boldsymbol{\Theta}) = \sum_{w \in V} \sum_{v \in b(w)} \log(\sum_{z^w=1}^{N_z(w)} \alpha_{z^w} \mathcal{N}(\boldsymbol{v}^{\boldsymbol{w}}|\boldsymbol{\mu}_{\boldsymbol{z}^{\boldsymbol{w}}}, \boldsymbol{\Sigma}_{\boldsymbol{z}^{\boldsymbol{w}}}))$$

where $\boldsymbol{\Theta}$ is to represent the model parameters defined above.

C. COMPONENT-BASED DIFFUSION PROCESS

Component-based diffusion models assume that nodes within a parent component have similar influence on their common child. We here associate each parent component z^w of the child node w with a diffusion probability $p(w|z^w) = \tau_{z^w,w}$. When a parent node in component z^w is activated at time t, there will be a probability $\tau_{z^w,w}$ that node w will be activated due to the component z_w. Note that we allow overlapping components. Thus, $p(w|v)$, the probability of a parent node v to activate a child node w, becomes an expected value of the diffusion probabilities $\{\tau_{z^w,w}\}$ over all the latent components, that is

$$\begin{aligned} p(w|v) &= \sum_{z^w=1}^{N_z(w)} p(z^w|v)p(w|z^w) = \sum_{z^w=1}^{N_z(w)} \frac{p(v|z^w)p(z^w)}{p(v^w)} \tau_{z^w,w} \\ &= \sum_{z^w=1}^{N_z(w)} \tau_{z^w,w} \alpha_{z^w} \mathcal{N}(\boldsymbol{v}|\boldsymbol{\mu}_{\boldsymbol{z}^{\boldsymbol{w}}}, \boldsymbol{\Sigma}_{\boldsymbol{z}^{\boldsymbol{w}}})/p(v^w). \end{aligned}$$

Given the proposed diffusion model, the diffusion process of a particular cascade proceeds as follows. Let $D_s = \{D_s(0), D_s(1) \cdots D_s(T_s)\}$ be the s^{th} observed information cascade where $D_s(t)$ is the set of nodes activated at time step t and T_s is the final time step for the cascade D_s. Given the initial set of activated nodes in the s^{th} cascade $(D_s(0))$, we assume that each of them will try to activate its child nodes. Note that we allow a parent node to be able to activate its child node not just for the next immediate time step but also for the subsequent time steps up to a limit. To explain that, we denote by $C_s(w, t)$ the set of nodes which have at least one activation within the interval between the latest activation of the child node w in the s^{th} cascade denoted as $L_w^{(s)}(t+1)$ and the time step t. This assumes that we only pay attention to recent news and that the posts earlier than our latest post have little influence on our future posting behaviour. $b(w) \cap C_s(w, t)$ then gives the subset of $C_s(w, t)$ which are parents of w. Then, the probability that

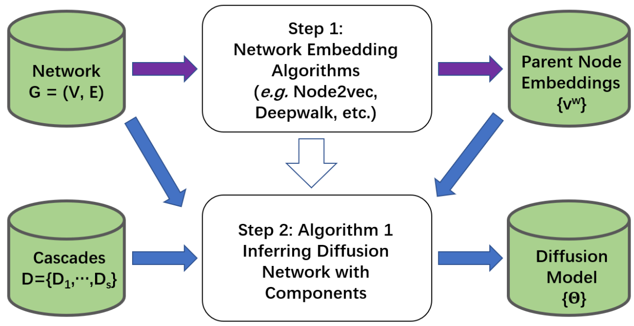


FIGURE 2. **Framework of our learning algorithm.**

the child node w will be activated at time $t+1$ is given as:

$$P(w(s, t+1)) = 1 - \prod_{v \in b(w) \cap C_s(w,t)} (1 - p(w|v)),$$

and whether node w will be activated is determined accordingly. The process proceeds until there is no more node being activated and then the cascade will stop.

We formulate the likelihood function under the delay agnostic setting as in [41], which counts a successful case for a parent node regardless of how many time steps it had tried before its child node got activated. We denote the set of nodes that were not activated by parent nodes as U_s, and the likelihood function of the observed cascades D_s can be formulated as follow:

$$\begin{aligned} L_2(\boldsymbol{\Theta}) &= \sum_{s=1}^{S} logP(D_s|\boldsymbol{\Theta}, D_0^{(s)}) \\ &= \sum_{s=1}^{S}\Bigg(\sum_{t=0}^{T_s-1} \sum_{w \in D_s(t+1)} logP(w(s, t+1)) \\ &\quad + \sum_{w \in U_s} \sum_{v \in b(w) \cap C_s(w,T_s-1)} log\Big(1 \\ &\quad - \sum_{z^w=1}^{N_z(w)} \tau_{z^w,w} \alpha_{z^w} \mathcal{N}(\boldsymbol{v}|\boldsymbol{\mu}_{\boldsymbol{z^w}}, \boldsymbol{\Sigma}_{\boldsymbol{z^w}})/p(v^w)\Big)\Bigg). \end{aligned}$$

To solve the component detection and diffusion network inference problems under a probabilistic framework, we can optimize the overall likelihood function, given as

$$L(\boldsymbol{\Theta}) = L_1(\boldsymbol{\Theta}) + L_2(\boldsymbol{\Theta}).$$

D. LEARNING ALGORITHM

We first adopt the widely used network embedding algorithm node2vec [30] to learn the embeddings of parent nodes of each node in the network. Then, a two-level expectation maximization (EM) algorithm is derived to obtain the ML estimates of the model parameters based on the observed cascades. The framework of our learning algorithm is shown in Fig. 2.

1) EM ALGORITHM

We make use of the expectation maximization (EM) algorithm [38], and derive a two-level EM algorithm to infer the latent parent components and the component-based diffusion probabilities simultaneously. We maximize the likelihood function $L(\boldsymbol{\Theta})$ with respect to the parameters $\boldsymbol{\Theta} = \{\{\alpha_{z^w}\}, \{\boldsymbol{\mu}_{\boldsymbol{z^w}}\}, \{\boldsymbol{\Sigma}_{\boldsymbol{z^w}}\}, \{\tau_{z^w,w}\}\}$. The inferred components will be those making the information diffusion mostly likely to happen, as well as grouping similar parent nodes.

a: FIRST LEVEL EM

Let I_{v,z^w} be a latent variable that takes the value of 1 when a parent node v belongs to the latent component z^w, and 0 otherwise, given the constraint $\sum_{z^w=0}^{N_z(w)} I_{v,z^w} = 1$. Let $I = \{I_{v,z^w}\}$ denote the whole set of the latent variables. We then derive the corresponding Q-function and obtain ML estimates via EM iterations consisting of an E-step and an M-step.

If we assume that I is known, the complete likelihood function can be written as $P(D, I|\boldsymbol{\Theta}) = P(D|I, \boldsymbol{\Theta})P(I|\boldsymbol{\Theta})$, where

$$P(I|\boldsymbol{\Theta}) = \prod_{w} \prod_{v \in b(w)} \prod_{z^w=0}^{N_z(w)} \left(\alpha_{z^w} \mathcal{N}(\boldsymbol{v^w}|\boldsymbol{\mu}_{\boldsymbol{z^w}}, \boldsymbol{\Sigma}_{\boldsymbol{z^w}})/p(v^w)\right)^{I_{v,z^w}},$$

and

$$\begin{aligned} P(D|I, \boldsymbol{\Theta}) &= L(\boldsymbol{\Theta}|I) \\ &= \sum_{s=1}^{S}\Bigg(\sum_{t=0}^{T_s-1} \sum_{w \in D_s(t+1)} \log p(w(s, t+1), I) \\ &\quad + \sum_{w \in U_s} \sum_{v \in b(w) \cap C_s(w,T_s-1)} \log(1 - \sum_{z^w=0}^{N_z(w)} I_{v,z^w} \tau_{z^w,w})\Bigg). \\ &\quad + \sum_{w \in V} \sum_{v \in b(w)} \log(\sum_{z^w=0}^{N_z(w)} I_{v,z^w} \mathcal{N}(\boldsymbol{v^w}|\boldsymbol{\mu}_{\boldsymbol{z^w}}, \boldsymbol{\Sigma}_{\boldsymbol{z^w}})) \end{aligned}$$

where

$$p(w(s, t+1)) = 1 - \prod_{v \in b(w) \cap C_s(w,t)} (1 - \sum_{z^w=0}^{N_z(w)} I_{v,z^w} \tau_{z^w,w}).$$

(a) E-step: As I is missing in most of the cases, we can do the E-step by first computing the posterior probabilities of I with the current parameter estimates $\hat{\tau}_{z^w,w}$, $\hat{\alpha}_{z^w}$, $\hat{\boldsymbol{\mu}}_{z^w}$ and $\hat{\boldsymbol{\Sigma}}_{z^w}$, given as

$$\begin{aligned} \eta_{v,z^w} &= P(I_{v,z^w} = 1|D, \hat{\boldsymbol{\Theta}}) \\ &= \frac{\hat{\alpha}_{z^w} \mathcal{N}(\boldsymbol{v^w}|\hat{\boldsymbol{\mu}}_{z^w}, \hat{\boldsymbol{\Sigma}}_{z^w}) P(D|I_{v,z^w} = 1, \hat{\boldsymbol{\Theta}})}{\sum_{z^w=0}^{N_z(w)} \hat{\alpha}_{z^w} \mathcal{N}(\boldsymbol{v^w}|\hat{\boldsymbol{\mu}}_{z^w}, \hat{\boldsymbol{\Sigma}}_{z^w}) P(D|I_{v,z^w} = 1, \hat{\boldsymbol{\Theta}})} \end{aligned}$$

where

$$\begin{aligned} &P(D|I_{v,z^w} = 1, \hat{\boldsymbol{\Theta}}) \\ &= \prod_{n:s \in S_w^+} \prod_{t \in T_{w,s}^+} E_{I \setminus I_{v,z^w}}[P(w|D, I_{v,z^w} = 1, \hat{\boldsymbol{\Theta}})] \\ &\quad \times \prod_{n:s \in S_w^-} I_{v \in b(w) \cap C_s(w,T_s-1)}(1 - \hat{\tau}_{z^w,w}). \end{aligned}$$

Here $I_{v \in b(w) \cap C_s(w,T_s-1)}$ is an indicator function which equals 1 if $v \in b(w) \cap C_s(w, T_s - 1)$. We define $T_{w,s}^+$ as the set of time steps $\{t\}$ in the s^{th} cascade satisfying that $w \in D_s(t+1)$ and $b(w) \cap C_s(w, T_s - 1) \neq \emptyset$. Moreover, we define $S_w^+ = \{D_s : \exists t\big(b(w) \cap C_s(w, t) \neq \emptyset \wedge w \in D_s(t+1)\big)\}$, and $S_w^- = \{D_s : U_s \neq \emptyset\}$.

Then, we take expectation of all possible assignments of I which can explain the observed cascades. The corresponding expected likelihood function, *i.e.,* Q-function, can be defined as

$$
\begin{aligned}
&\mathcal{Q}(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}})\\
&= \sum_{s=1}^{S}\Big(\sum_{t=0}^{T_s-1}\sum_{w\in D_s(t+1)} E_I[\log p(w(s,t+1),I)]\\
&\quad + \sum_{w\in U_s}\sum_{v\in b(w)\cap C_s(w,T_s-1)}\sum_{z^w=0}^{N_w(w)} \eta_{v,z^w}\log(1-\tau_{z^w,w})\Big)\\
&\quad + \sum_{w\in V}\sum_{v\in b(w)}\sum_{z^w=0}^{N_z(w)} \eta_{v,z^w}\log\mathcal{N}(\boldsymbol{v^w}|\boldsymbol{\mu}_{\boldsymbol{z^w}},\boldsymbol{\Sigma}_{\boldsymbol{z^w}})\\
&\quad + \sum_{w\in V}\sum_{v\in b(w)}\sum_{z^w=0}^{N_z(w)} \eta_{v,z^w}\log(\alpha_{z^w}\mathcal{N}(\boldsymbol{v^w}|\boldsymbol{\mu}_{\boldsymbol{z^w}},\boldsymbol{\Sigma}_{\boldsymbol{z^w}})/p(v^w))
\end{aligned}
$$

(b) M-step: For the M-step, we maximize $\mathcal{Q}(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}})$ by taking the derivative of $\mathcal{Q}$ with respect to $\boldsymbol{\Theta}$ to obtain the updating rule of the model parameters. Therefore,

$$
\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{z^w}} &= \frac{1}{\sum_{v\in b(w)}\eta_{v,z^w}}\sum_{v\in b(w)}\eta_{v,z^w}\boldsymbol{v^w}\\
\boldsymbol{\Sigma}_{\boldsymbol{z^w}} &= \frac{1}{\sum_{v\in b(w)}\eta_{v,z^w}}\sum_{v\in b(w)}\eta_{v,z^w}(\boldsymbol{v^w}-\boldsymbol{\mu}_{\boldsymbol{z^w}})(\boldsymbol{v^w}-\boldsymbol{\mu}_{\boldsymbol{z^w}})^T
\end{aligned}
$$

To update $\{\alpha_{z^w}\}$, according to the Lagrange multiplier method, maximizing $\mathcal{Q}(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}})$ with the constraint $\sum_{z^w=0}^{N_z(w)}\alpha_{z^w} = 1$ yields $\partial\Big(\mathcal{Q}(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}}) - \lambda(\sum_{z^w=0}^{N_z(w)}\alpha_{z^w} - 1)\Big)/\partial\alpha_{z^w} = 0$. Thus,

$$
\alpha_{z^w} = \frac{\sum_{v\in b(w)}\eta_{v,z^w}}{\sum_{v\in b(w)}\sum_{z^w=1}^{N_z(w)}\eta_{v,z^w}} = \frac{\sum_{v\in b(w)}\eta_{v,z^w}}{|b(w)|}.
$$

To update $\{\tau_{z^w}\}$, setting to zero the derivative for the first term $E_I[\log p(w(s,t+1),I)]$ in $\mathcal{Q}(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}})$ does not have a simple solution. So, within this M-step, we introduce another level of the EM algorithm.

b: SECOND LEVEL EM

Let $Y_{v,w}^{(s)}(t)$ denote a latent variable that indicates whether the activation of a node w at time step t in the s^{th} cascade is due to w's parent node v or not. We further define $Y_s = \{Y_s(0), Y_s(1)\cdots Y_s(T_s)\}$ where $Y_s(t) := \{Y_{v,w}^{(s)}(t)\}$ represents the set of latent variables corresponding to the activations at time step t in the s^{th} cascade. Then, we derive corresponding Q-function and infer parameters $\{\tau_{z^w}\}$ via second-level EM iterations.

(a) E-step: First, we compute the posterior probability of $Y_{v,w}^{(s)}(t)$, given as

$$
\begin{aligned}
\gamma_{v,w,s,t} &= P(Y_{v,w}^{(s)}(t+1)=1|D,\{\eta_{v,z^w}\},\hat{\boldsymbol{\Theta}})\\
&= \frac{\sum_{z^w=0}^{N_z(w)}\eta_{v,z^w}\hat{\tau}_{z^w,w}}{\hat{p}(w(s,t+1))}
\end{aligned}
$$

where $\hat{\tau}_{z^w,w}$ stands for the current estimate of $\tau_{z^w,w}$, and

$$
\hat{p}(w(s,t+1)) = 1 - \prod_{v\in b(w)\cap C_s(w,t)}(1-\sum_{z^w=0}^{N_z(w)}\eta_{v,z^w}\hat{\tau}_{z^w,w}).
$$

Then, we take expectation of all possible assignments of Y and the corresponding Q-function can then be defined as

$$
\begin{aligned}
&\mathcal{Q}'(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}})\\
&= \sum_{s=1}^{S}\Big(\sum_{t=0}^{T_s-1}\sum_{w\in D_s(t+1)}\sum_{v\in b(w)\cap C_s(w,t)}\sum_{z^w=0}^{N_z(w)}\eta_{v,z^w}\\
&\quad \big(\gamma_{v,w,s,t}\log\tau_{z^w,w} + (1-\gamma_{v,w,s,t})\log(1-\tau_{z^w,w})\big)\\
&\quad + \sum_{w\in U_s}\sum_{v\in b(w)\cap C_s(w,T_s-1)}\sum_{z^w=0}^{N_z(w)}\eta_{v,z^w}\log(1-\tau_{z^w,w})\Big)\\
&\quad + \sum_{w\in V}\sum_{v\in b(w)}\sum_{z^w=0}^{N_z(w)}\eta_{v,z^w}\log\mathcal{N}(\boldsymbol{v^w}|\boldsymbol{\mu}_{\boldsymbol{z^w}},\boldsymbol{\Sigma}_{\boldsymbol{z^w}})\\
&\quad + \sum_{w\in V}\sum_{v\in b(w)}\sum_{z^w=0}^{N_z(w)}\eta_{v,z^w}\log(\alpha_{z^w}\mathcal{N}(\boldsymbol{v^w}|\boldsymbol{\mu}_{\boldsymbol{z^w}},\boldsymbol{\Sigma}_{\boldsymbol{z^w}})/p(v^w)).
\end{aligned}
$$

(b) M-step: For the M-step, we maximize $\mathcal{Q}'(\boldsymbol{\Theta}|\hat{\boldsymbol{\Theta}})$ by taking the derivative of $\mathcal{Q}'$ with respect to $\{\tau_{z^w}\}$ to obtain the updating rule of $\{\tau_{z^w}\}$. Thus, $\partial\mathcal{Q}'/\partial\tau_{z^w,w} = 0$ yields:

$$
\begin{aligned}
\tau_{z^w,w} &= \frac{1}{N_{z^w,w}}\sum_{s\in S_w^+}\sum_{t\in T_{w,s}^+}\sum_{v\in b(w)\cap C_s(w,t)}\eta_{v,z^w}\gamma_{v,w,s,t}\\
N_{z^w,w} &= N_{z^w,w}^+ + N_{z^w,w}^-\\
N_{z^w,w}^+ &= \sum_{s\in S_w^+}\sum_{t\in T_{w,s}^+}\sum_{v\in b(w)\cap C_s(w,t)}\eta_{v,z^w}\\
N_{z^w,w}^- &= \sum_{s\in S_w^-}\sum_{v\in b(w)\cap C_s(w,T_s-1)}\eta_{v,z^w}.
\end{aligned}
$$

The E-step and M-step repeat until convergence. The detailed steps are summarized in Algorithm 1.

2) COMPUTATIONAL COMPLEXITY

Implementing the learning algorithm involves two main steps: 1) load the network data, the per-node neighbors' embeddings and the cascades related data, and 2) carry out the EM iterations.

For step 1), the cost for loading the network data is $O(|V|+|E|)$, and the cost for loading the cascades information is $O(S\times T)$ where T denotes the maximum length of a cascade record. The cost for loading the per-node neighbors' embedding is $O(|E|)$, since visiting the neighbors for each node is equivalent to that of visiting all edges. Thus, the overall complexity is $O(|V|+|E|+S\times T)$.

For step 2), we calculate the cost for each EM iteration. The E-step (lines 5-8) visits the neighbors $v\in b(w)$ for each node $w\in V$, which is equivalent to visiting all edges. By denoting I_{max} to be the maximum number of components set among all nodes, the worst case complexity becomes $O(I_{max}\times|E|)$.

Algorithm 1 Inferring Diffusion Network With Components

```
   Input: network $G = (V, E)$, cascades
          $D = \{D_1, \cdots, D_S\}$, parent node embeddings $\boldsymbol{v^w}$
          for each node $w \in V$, dimension $d$.
   Output: parent components for each node $w \in V$
          $\{\{\alpha_{z^w}\}, \{\boldsymbol{\mu}_{\boldsymbol{z^w}}\}, \{\boldsymbol{\Sigma}_{\boldsymbol{z^w}}\}\}$, component-based
          diffusion probabilities $\{\tau_{z^w,w}\}$
1  Assign initial values to
   $\hat{\boldsymbol{\Theta}} = \{\{\hat{\alpha}_{z^w}\}, \{\hat{\boldsymbol{\mu}}_{z^w}\}, \{\hat{\boldsymbol{\Sigma}}_{z^w}\}, \{\hat{\tau}_{z^w,w}\}\}$
2  //first level EM:
3  while not convergence do
4      //E-step:
5      for all $(w, v, z^w) : w \in V$ and $v \in b(w)$ do
6          $\eta_{v,z^w} \leftarrow$
7          $\frac{\hat{\alpha}_{z^w} \mathcal{N}(\boldsymbol{v^w} | \hat{\boldsymbol{\mu}}_{z^w}, \hat{\boldsymbol{\Sigma}}_{z^w}) P(D|I_{v,z^w}=1, \hat{\boldsymbol{\Theta}})}{\sum_{z^w=0}^{N_z(w)} \hat{\alpha}_{z^w} \mathcal{N}(\boldsymbol{v^w} | \hat{\boldsymbol{\mu}}_{z^w}, \hat{\boldsymbol{\Sigma}}_{z^w}) P(D|I_{v,z^w}=1, \hat{\boldsymbol{\Theta}})}$
8      end for
9      //M-step:
10     for all $(w, z^w) : w \in V$ do
11         $\alpha_{z^w} \leftarrow \frac{\sum_{v \in b(w)} \eta_{v,z^w}}{|b(w)|}$
12         $\boldsymbol{\mu}_{\boldsymbol{z^w}} \leftarrow \frac{1}{\sum_{v \in b(w)} \eta_{v,z^w}} \sum_{v \in b(w)} \eta_{v,z^w} \boldsymbol{v^w}$
13         $\boldsymbol{\Sigma}_{\boldsymbol{z^w}} \leftarrow \frac{1}{\sum_{v \in b(w)} \eta_{v,z^w}} \sum_{v \in b(w)} \eta_{v,z^w}$
14                 $(\boldsymbol{v^w} - \boldsymbol{\mu}_{\boldsymbol{z^w}})(\boldsymbol{v^w} - \boldsymbol{\mu}_{\boldsymbol{z^w}})^T$
15     end for
16     for all $(w, z^w) : w \in V$ do
17         $N^+_{z^w,w} \leftarrow \sum_{s \in S^+_w} \sum_{t \in T^+_{w,s}}$
18                 $\sum_{v \in b(w) \cap C_s(w,t)} \eta_{v,z^w}$
19         $N^-_{z^w,w} \leftarrow \sum_{s \in S^-_w} \sum_{v \in b(w) \cap C_s(w,T_s-1)} \eta_{v,z^w}$
20     end for
21     //second level EM:
22     while not convergence do
23         //E-step:
24         for all $(w, s, t) : w \in D_s(t+1)$ do
25             $\hat{p}(w(s, t+1)) \leftarrow 1 - \prod_{v \in b(w) \cap C_s(w,t)}$
26                         $(1 - \sum_{z^w=0}^{N_z(w)} \eta_{v,z^w} \hat{\tau}_{z^w,w})$
27         end for
28         for all $(v, w, s, t) : w \in D_s(t+1)$ and
   $v \in b(w) \cap C_s(w, t)$ do
29             $\gamma_{v,w,s,t} \leftarrow \frac{\sum_{z^w=0}^{N_z(w)} \eta_{v,z^w} \hat{\tau}_{z^w,w}}{\hat{p}(w(s,t+1))}$
30         end for
31         //M-step:
32         for all $(w, z^w) : w \in V$ do
33             $\tau_{z^w,w} \leftarrow \frac{1}{N^+_{z^w,w} + N^-_{z^w,w}} \sum_{s \in S^+_w} \sum_{t \in T^+_{w,s}}$
34                     $\sum_{v \in b(w) \cap C_s(w,t)} \eta_{v,z^w} \gamma_{v,w,s,t}$
35         end for
36         $\{\hat{\tau}_{z^w,w}\} \leftarrow \{\tau_{z^w,w}\}$
37     end while // end of second level EM
38     $\hat{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}$
39 end while //end of first level EM
```

For M-step, the worst case complexity for lines $10 - 15$ is also $O(I_{max} \times |E|)$, and $O(I_{max} \times |E| \times S \times T)$ for lines $16 - 20$. And by denoting the number of iterations as k, the worst case complexity for the part of second-level EM iterations becomes $O(k \times I_{max} \times |E| \times S \times T)$. Thus, the worst case complexity for each first-level EM iteration is $O(k \times I_{max} \times |E| \times S \times T))$.

III. EXPERIMENT

We compare our model with the basic models as well as the recently proposed models. We use both synthetic and real world data sets to evaluate the model accuracy and discuss how the use of the embedding-based framework can enhance both the component identification and diffusion prediction, and how our model can be applied to support dependency analysis of different online news media.

A. EXPERIMENTAL SETTINGS

In all our experiments, the number of parent components $N_z(w)$ of each child node w is set as the number of communities in the child node w's neighborhood network estimated by the CNM community detection algorithm [25] for all the component-based models. Also, for all the experiments, the initial values of $\tau_{z^w,w}$ are within [0, 1], and the initial values of α_{z^w} are generated within [0, 1] satisfying $\sum_{z^w} \alpha_{z^w} = 1$. For learning embeddings, the nodes within two hops are considered instead of just the immediate neighbors for more precisely learning of the embeddings of parent nodes, and only the embeddings of parent nodes are kept. The widely used random-walk based network embedding method node2vec [30] algorithm is adopted, which introduces biased random walks to balance between BFS and DFS sampling strategies using hyperparameters p and q. The parameters are set as $p = 0.125$, $q = 16$ that is biased to BFS-strategy sampling nearby nodes mostly in the same communities.

We compare our model with the basic IC model, a component-based diffusion model and an embedding-based IC model, given as:

1) *ICM*: The basic IC model. We extends the original IC model by considering the influence of all the parent nodes activated after the child node's latest activation instead of only those just activated at the previous time step. The main reason of using this modified IC model is to make sure that the comparison is only based on whether the components structure is adopted or not.
2) *Comp*: The component-based model proposed in [24]. The components of each child node are directly detected by the CNM community detection algorithm [25]. Parent nodes within two hops are considered to enhance the detection accuracy. We did not model redundancy in *Comp* for fair comparison about whether a particular way of identifying components is useful. Also, the parent nodes belonging to the same component have the same component-based diffusion

probability instead of independent values as in IC model.

3) *IC_Emb*: The embedding-based IC model in [16]. A set of sender embedding and receiver embedding is learnt for each node. And the diffusion probabilities are modeled as a function of sender embeddings and receiver embeddings, instead of independent values. However, neither local connectivity information nor the notion of parent components is considered.

4) *Comp_Emb*: The proposed component-based model with embedding-based parent components.

B. PERFORMANCE EVALUATION

As the ground-truth is unknown for real data, we adopt perplexity for evaluating the performance of various models on predicting unseen data. Perplexity is widely used for evaluation of language models [42], which calculates the average probability for each word to be generated by the trained model. For our case, the *perplexity* over the observed cascades is defined as

$$Perplexity = \frac{-\sum_{s=1}^{S} \ln P(D_s)}{W}.$$

where $P(D_s)$ is the probability for the s^{th} cascade to be generated, and W is the normalization term representing the number of activations due to the influence of the corresponding nodes' parents. The average probability is negatively correlated with perplexity. A lower perplexity score indicates the inferred model to be more probable, and thus indicates better performance. In particular, we denote the average probability for the first model as P_1, the second as P_2. Also the perplexity for the first model as PPL_1, the second as PPL_2. Then the relation between the improvement of average probability and the decrease in perplexity becomes $\ln \frac{P_2}{P_1} = PPL_1 - PPL_2$. When the decrease in perplexity equals 0, then $\frac{P_2}{P_1}$ equals 1, meaning the models give the same performance. When the decreased perplexity scores take values of 0.05, 0.1, 0.15 and 0.5, then $\frac{P_2}{P_1}$ becomes 0.05, 0.1, 0.16 and 0.65, which means there are 5%, 10%, 16% and 65% improvement of average probability respectively. Also, we divide the cascades into five folds and obtain the average performance using cross-validation.

C. EXPERIMENTS ON SYNTHETIC DATA

We generate synthetic cascades based on our model and anticipate that the inferred model with the same assumption for cascade generation should perform the best.

1) EXPERIMENTAL SETUP

We first generate two scale-free networks of 1000 nodes using the SNAP platform [43] as real networks are most scale-free. One network is generated with 5000 edges and the other with 10000 edges. For each network, 100 cascades are generated based on our model, where the diffusion probabilities are randomly assigned within [0, 1] and the embeddings are learnt from the network structure in advance. Note that the network with 10000 edges is denser and thus more activations in the generated cascades, providing more data for model training. We then investigate the connectivity of the nodes in the parent components detected by the CNM community detection algorithm in the basic component-based diffusion model *Comp*. Each component is measured with clustering coefficient which is defined as the network density of the corresponding subgraph of the component. Fig 3 shows the probability density function of the clustering coefficients of parent components in the two networks. The high percent of components with low clustering coefficients indicates that in scale-free networks, which is common for real networks, not all nodes in the detected components are closely connected, and thus we shall investigate local connectivity information of neighbors for better identification of components.

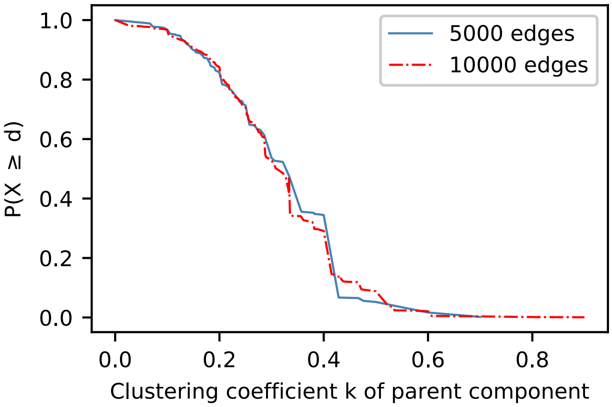


FIGURE 3. Probability density function of the clustering coefficients of parent components for synthetic networks with 5000 and 10000 edges.

2) GENERATIVE ABILITY

We apply *ICM*, *Comp*, *IC_Emb* and *Comp_Emb* to the synthetic networks. The performance comparison results in terms of perplexity are shown in Fig. 4. We observe that all the models perform better for the network with 10000 edges when compared with that with 5000 edges due to more training data. Also, for the network with 5000 edges, while *Comp* and *IC_Emb* can result in a comparable perplexity decrease of 0.74 and 0.79 in terms of *ICM* respectively, our model *Comp_Emb* apparently gets further improvement and achieves a perplexity decrease of 1.37 in terms of *ICM* and is most similar to the groundtruth value. Similar phenomenon can be observed for the network with 10000 edges. While both *Comp* and *IC_Emb* result in a perplexity decrease of 0.05 in terms of *ICM*, our model achieves a perplexity decrease of 0.08.

D. EXPERIMENTS ON REAL DATA

To validate if our model assumption indeed happens in online social and information networks, we apply the proposed model to real data sets.

1) DATA SET

Two real datasets are used for the evaluation, namely MemeTracker [44] and Digg [45] where both the network structure and the information cascades are available, given as:

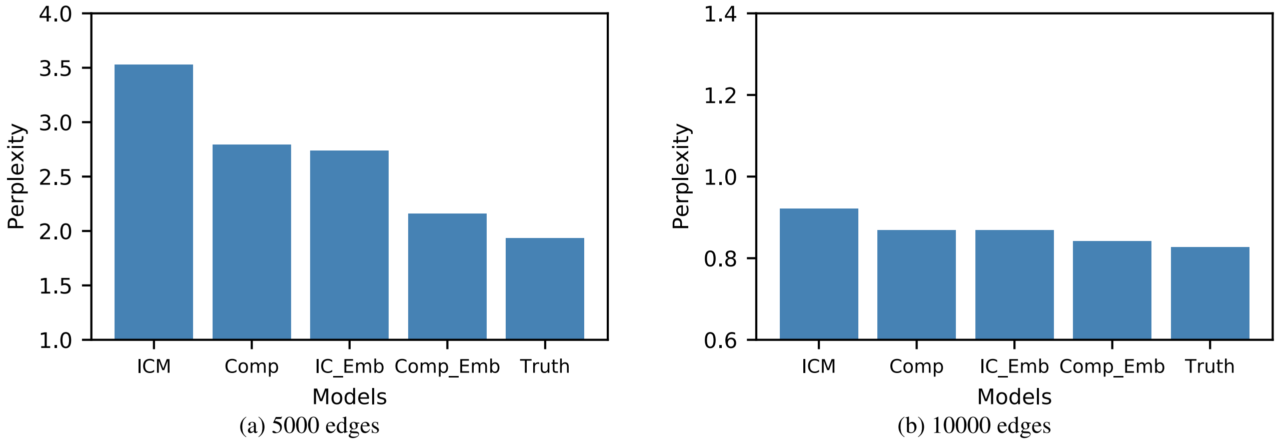


FIGURE 4. Performance comparison on synthetic data for networks with (a) 5000 edges and (b) 10000 edges.

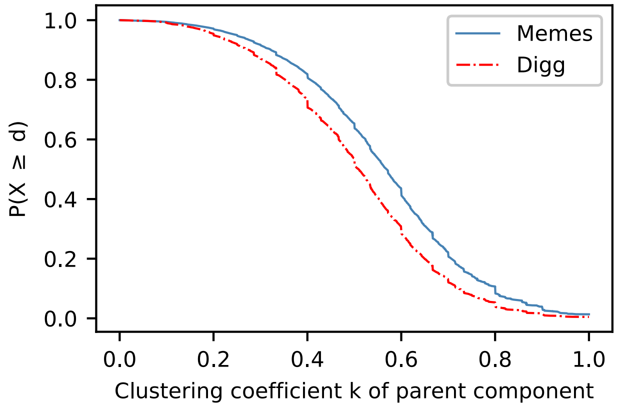


FIGURE 5. Probability density function of the clustering coefficients of parent components for real networks.

MemeTracker: The posts of mass media and weblogs from August 1 2008 to April 30 2009. Websites with news articles and blog posts are modeled as nodes which are further connected by directed edges (hyperlinks). Each cascade is defined based on a set of posts reporting the same event. We filter out the websites publishing less than 50 articles and get 11, 457 nodes and 71, 460 cascades.

Digg: The story voting process under a directed friendship network over one month in 2009. Users are modeled as nodes and following relations are modeled as edges. Each cascade is defined based on a particular frequently voted story. We filter out the websites publishing less than 50 articles and extract the corresponding cascades, resulting in 8, 954 nodes and 3, 553 cascades.

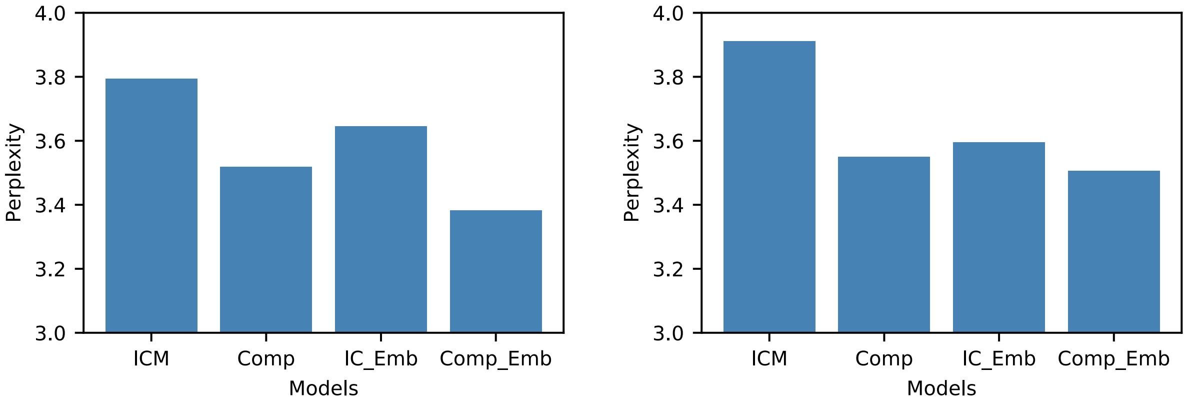
We also investigate the connectivity of the nodes in the parent components detected by the CNM community detection algorithm in the basic component-based diffusion model *Comp* for the two real networks. Fig 5 shows the probability density function of the clustering coefficients of parent components in both MemeTracker and Digg datasets. The high percent of components with low clustering coefficients indicates that not all nodes in the detected components are closely connected and thus we shall investigate local connectivity information of neighbors for better identification of components instead of using global quantities such as modularity.

2) GENERATIVE ABILITY

Figure 6 shows the experimental results. The results consistently show that our model *Comp_Emb* outperforms the basic IC model, the basic component-based model *Comp*, and the embedding-based IC model *IC_Emb*. In particular, our model outperforms *ICM* by a perplexity decrease of 0.41 and 0.40 on MemeTracker and Digg datasets respectively, and outperforms *Comp* by a decrease of 0.14 and 0.04 respectively. It indicates that incorporating local connectivity information in component-based models can further result in more accurate models to be inferred. Our model achieves more apparent improvement on MemeTracker dataset as the information sources of news media are from a variety of types and they form components with a wide range of influence ability. Thus, the identification of components plays a vital role. Also, our model outperforms *IC_Emb* by a decrease of 0.26 and 0.09 respectively on the two data sets. Therefore, node connectivity information as well as components considered in our model are important factors when learning node embeddings for modeling information diffusion. In addition, we test different numbers of dimensions for the node embeddings as shown in Fig 7 and the settings giving the best performance are adopted.

3) QUALITATIVE EVALUATION

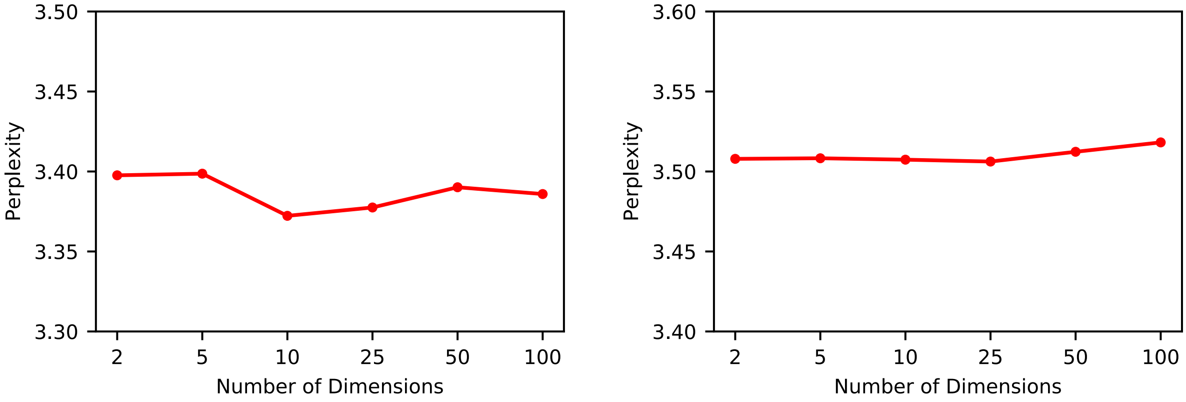
To illustrate how well the neighborhood network embeddings and the components are learnt, we visualize the neighbors' embeddings of the websites New York Times (abbreviated as NYTimes) and Seattle Local News (abbreviated as SEANews). Figure 8 shows (1) the learnt embeddings projected into two-dimensional space using the t-SNE [46] algorithm and (2) the mean of each component represented by red triangles. For comparison, the embeddings are colored based on the result of CNM community detection algorithm, with one color representing one cluster. Our model can detect three major components for NYTimes and two for SEANews, while CNM can only detect two major components for NYTimes. It is noted that the nodes in the two major parent components of NYTimes and SEANews identified by the CNM algorithm are also close in our embedding space. However, for NYTimes, near the boundary of the two



(a) MemeTracker

(b) Digg

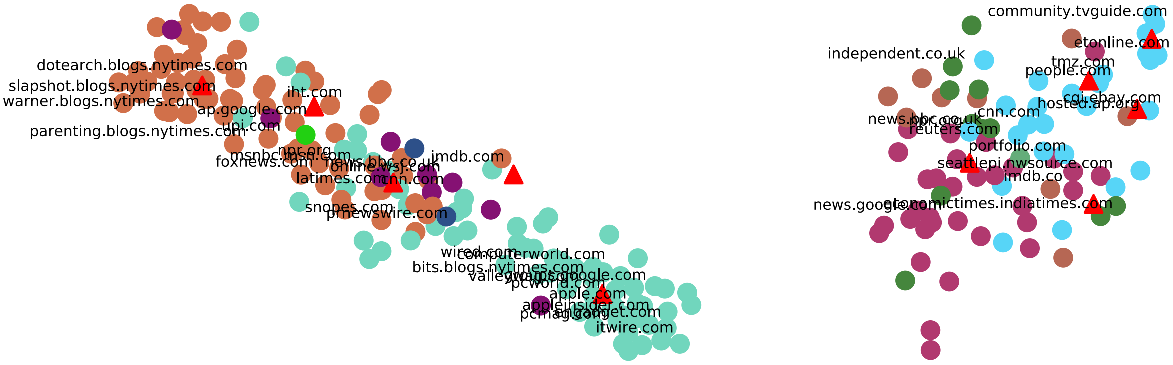
FIGURE 6. **Performance comparison on real data for (a) MemeTracker and (b) Digg datasets.**



(a) MemeTracker

(b) Digg

FIGURE 7. **Effect of embedding dimensions on model accuracy.**

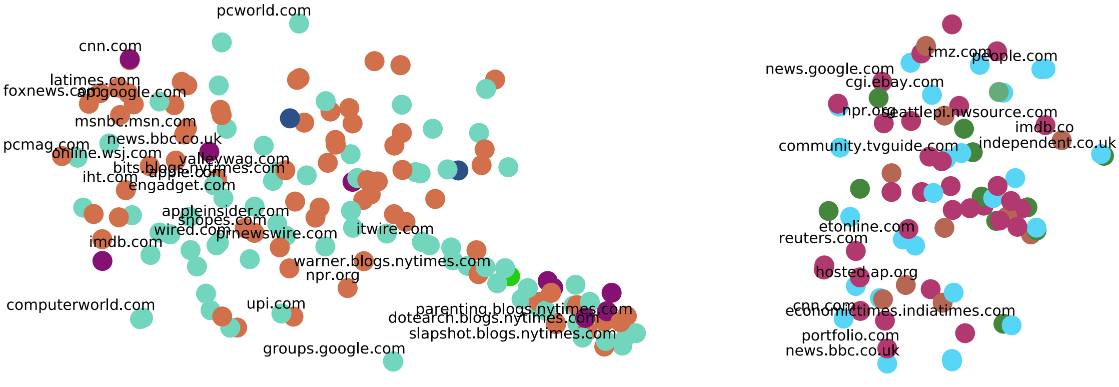


(a) NYTimes

(b) SEANews

FIGURE 8. **The neighborhood node embeddings and components learnt from our model for the website of New York Times (abbreviated as NYTimes) and Seattle Local News (abbreviated SEANews). It shows (1) the embeddings projected into two-dimensional space using the t-SNE [46] algorithm (2) the mean of each component represented as red triangles. For comparison, the embeddings are colored based on the result of CNM community detection algorithm (each color represents one cluster).**

components lie the mass media websites such as *news.bbc.co.uk* (BBC News), *foxnews.com* (Fox News), and *et al.* Although they are in the same color with **.blogs.nytimes.com*, the two sets of websites are relatively far away and indeed should form another category. They are recognized as another component by our model. In addition, the diffusion data helps learning more meaningful components. There is a red triangle around the websites reporting international news, including *iht.com* (International Herald Tribune), *ap.google.com* (Google News), and *upi.com* (United Press International). They should have different degrees of influence as compared to other mass media on NYTimes, even though they are near in the embedding space. This can also be identified by our model as a separate component.



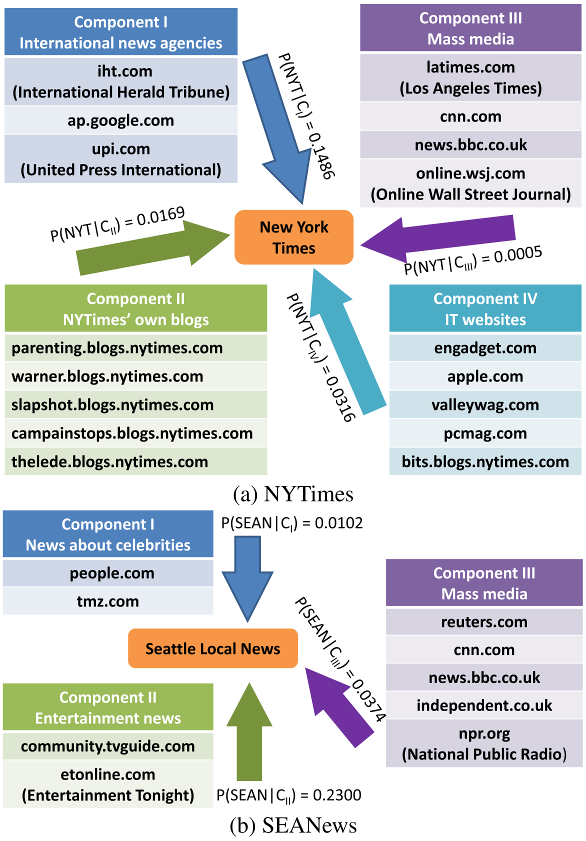
(a) NYTimes

(b) SEANews

FIGURE 9. The neighborhood node embeddings learnt from the baseline embedding-based IC model *IC_Emb* for the website of New York Times (abbreviated as NYTimes) and Seattle Local News (abbreviated as SEANews). The embeddings are projected into two-dimensional space using the t-SNE [46] algorithm, and are colored based on the result of CNM community detection algorithm for comparison (each color represents one cluster).

Similar phenomena can be found for SEANews. On the right corner, it shows our model can identify the entertainment news websites such as *etonline.com* (Entertainment Tonight) and *community.tvguide.com* (TV Guide), and the entertainment news websites specifically about celebrities such as *people.com* (People) and *tmz.com* (Thirty Mile Zone) as two separate components.

For comparison, Figure 9 shows the neighborhood node embeddings of NYTimes and SEANews learnt from the baseline embedding-based IC model *IC_Emb* by projecting the embeddings into two-dimensional space. The embeddings are also colored based on the result of CNM community detection algorithm for comparison. Apparently, the learnt node embeddings by *IC_Emb* are not well clustered. Nevertheless, some of the major nodes in the major components identified by our model are also relatively close in the embedding space learnt from *IC_Emb*. For NYTimes, there are some mass media websites on the left (*e.g.*, *online.wsj.com* (Wall Street Journal), *cnn.com* (CNN News), *latimes.com* (Los Angeles Times), *foxnews.com* (Fox News), and *news.bbc.co.uk* (BBC News)), information technology websites beside them (*e.g.*, *bits.blogs.nytimes.com*, *valleywag.com* (Valleywag), *computerworld.com* (Computerworld) and *pcmag.com* (PCMag)), and weblogs of NYTimes itself on the right corner (**.blogs.nytimes.com*). For SEANews, there are some mass media websites (*e.g.*, *cnn.com* (CNN News), *news.bbc.co.uk* (BBC News), and *portfolio.com* (Portfolio)). Also, *people.com* (People) and *tmz.com* (Thirty Mile Zone) are still relatively closer than other nodes, and the same for *etonline.com* (Entertainment Tonight) and *community.tvguide.com* (TV Guide). The main reason behind is that *IC_Emb* learns parent node embeddings based on observed cascades, and parent nodes with similar influence are close in the embedding space. As the major nodes within each parent component identified by our model have similar influence on its child node, the learnt embeddings are also close under *IC_Emb*. However, only parts of the nodes are clustered under *IC_Emb* as the observed cascades are not sufficient. Meanwhile, our model learns node embeddings based on local connectivity information and directly models closely connected nodes as components based on the embeddings, and thus can achieve better performance.



(a) NYTimes

(b) SEANews

FIGURE 10. The major components with corresponding component-based diffusion probabilities learnt from our model for the websites of New York Times (abbreviated as NYTimes) and Seattle Local News (abbreviated as SEANews).

Figure 10 lists the major components identified for NYTimes and SEANews with associated component-based diffusion probabilities under our model. It shows that the parent nodes for each component are well associated with NYTimes and SEANews respectively. For NYTimes, *Component I* is most influential and composes of websites of international news agencies. *Component II* are weblogs of NYTimes itself. They contain featured articles of different sections of NYTimes' own blogs, which are written by well-known experts. *Component III* are webistes of well-known mass media such as *cnn.com* (CNN News), *news.bbc.co.uk* (BBC News) and *et al*. *Component IV* are mainly information technology websites such as *engadget.com* (Engadget) and *pcworld.com* (PC World), and some related weblogs. For SEANews, *Component I* and *Component II* are websites about entertainment news, but *Component I* is specially on entertainment news about celebrities and less influential than *Component II*. *Component III* contains mass media websites such as *cnn.com* (CNN News), *news.bbc.co.uk* (BBC News) and *et al.*

IV. CONCLUSION

In this paper, we proposed to model local connectivity information of neighbors to enhance both component identification and diffusion network inference in social networks. In particular, we embedded nodes in a latent vector space using network embedding and proposed a component-based diffusion model with embedding-based parent components. A united probabilistic framework was proposed so that the parent components and the component-based diffusion models can be inferred jointly. Also, a two-level EM algorithm was derived for model inference based on the observed information cascades. For performance evaluation, we applied the proposed model to both synthetic and real world data sets with promising results obtained. We also discussed how the use of the embedding-based framework can enhance both component identification and the diffusion model.

This paper has some limitations. For instance, unlike some related work where network structure is unknown [47], our work is applied to the situations where network structure is known. And also for simplicity, the diffusion rate is assumed to be static and topic-independent, and the activations only occur at discrete time steps. For future work, the above assumptions can be further relaxed. The proposed model can also be applied to other network analysis tasks such as influence maximization.

REFERENCES

[1] J. Goldenberg, B. Libai, and E. Muller, "Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata," *Acad. Marketing Sci. Rev.*, vol. 9, no. 3, pp. 1–18, 2001.

[2] D. Kempe, J. Kleinberg, and E. Tardos, "Influential nodes in a diffusion model for social networks," in *Proc. 32nd Int. Conf. Automata, Lang. Program.*, Berlin, Germany, 2005, pp. 1127–1138.

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2003, pp. 137–146.

[4] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2009, pp. 199–208.

[5] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "A data-based approach to social influence maximization," *Proc. VLDB Endowment*, vol. 5, no. 1, pp. 73–84, Sep. 2011.

[6] X. Song, Y. Chi, K. Hino, and B. L. Tseng, "Information flow modeling based on diffusion rate for prediction and ranking," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, New York, NY, USA, 2007, pp. 191–200.

[7] X. Song, B. L. Tseng, C.-Y. Lin, and M.-T. Sun, "Personalized recommendation driven by information flow," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, New York, NY, USA, 2006, pp. 509–516.

[8] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "Pagerank with priors: An influence propagation perspective," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Beijing, China, 2013, pp. 2740–2746.

[9] S. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 1–9.

[10] M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrishnan, "DeepDiffuse: Predicting the 'who' and 'when' in cascades," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1055–1060.

[11] W. Lee, J. Kim, and H. Yu, "CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing," in *Proc. 12th Int. Conf. Data Mining*, Brussels, Belgium, 2012, pp. 960–965.

[12] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "Learning influence probabilities in social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, 2010, pp. 241–250.

[13] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," *Knowl. Inf. Syst.*, vol. 37, no. 3, pp. 555–584, Dec. 2013.

[14] Y. Yang, J. Tang, C. W. K. Leung, Y. Sun, Q. Chen, J. Li, and Q. Yang, "Rain: Social role-aware information diffusion," in *Proc. 19th AAAI Conf. Artif. Intell.*, 2015, pp. 367–373.

[15] H. Xu, J. Wei, Z. Yang, J. Ruan, and J. Wang, "Probabilistic topic and role model for information diffusion in social network," in *Proc. 22nd Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2018, pp. 3–15.

[16] S. Bourigault, S. Lamprier, and P. Gallinari, "Representation learning for information diffusion through social networks: An embedded cascade model," in *Proc. 9th ACM Int. Conf. Web Search Data Mining (WSDM)*, San Francisco, CA, USA, 2016, pp. 573–582.

[17] M. Granovetter, "Threshold models of collective behavior," *Amer. J. Sociol.*, vol. 83, no. 6, pp. 1420–1443, May 1978.

[18] M. O. Jackson and L. Yariv, "Diffusion of behavior and equilibrium properties in network games," *Amer. Econ. Rev.*, vol. 97, no. 2, pp. 92–98, Apr. 2007.

[19] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg, "Structural diversity in social contagion," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 16, pp. 5962–5966, Apr. 2012.

[20] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, "Social influence locality for modeling retweeting behaviors," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, Beijing, China, 2013, pp. 2761–2767.

[21] R. S. Burt, *Structural Holes: The Social Structure of Competition*. Cambridge, MA, USA: Harvard Univ. Press, 1992.

[22] J.-P. Onnela, J. Saramaki, J. Hyvonen, G. Szabo, D. Lazer, K. Kaski, J. Kertesz, and A.-L. Barabasi, "Structure and tie strengths in mobile communication networks," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 18, pp. 7332–7336, May 2007.

[23] J. Tang, T. Lou, and J. Kleinberg, "Inferring social ties across heterogenous networks," in *Proc. 5th ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, 2012, pp. 743–752.

[24] Q. Bao, W. K. Cheung, Y. Zhang, and J. Liu, "A component-based diffusion model with structural diversity for social networks," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1078–1089, Apr. 2017.

[25] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, pp. 066111–066116, Dec. 2004.

[26] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2019.

[27] Y. Lu, X. Wang, C. Shi, P. S. Yu, and Y. Ye, "Temporal network embedding with micro- and macro-dynamics," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2019, pp. 469–478.

[28] B. Shi, J. Zhong, Q. Bao, H. Qiu, and J. Liu, "EpiRep: Learning node representations through epidemic dynamics on networks," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, 2019, pp. 486–492.

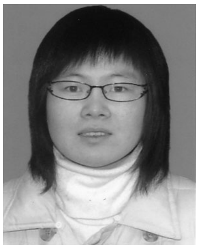
[29] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2014, pp. 701–710.

[30] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2016, pp. 855–864.
[31] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1145–1152.
[32] C. Tu, W. Zhang, Z. Liu, and M. Sun, "Max-margin deepwalk: Discriminative learning of network representation," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, 2016, pp. 3889–3895.
[33] D. Shen, X. Zhang, R. Henao, and L. Carin, "Improved semantic-aware network embedding with fine-grained word alignment," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium, 2018, pp. 1–10.
[34] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (EOE): Joint embedding for coupled heterogeneous networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., 2017, pp. 741–749.
[35] X. Wang, Y. Zhang, and C. Shi, "Hyperbolic heterogeneous information network embedding," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, pp. 5337–5344.
[36] Y. Zuo, G. Liu, H. Lin, J. Guo, X. Hu, and J. Wu, "Embedding temporal network via neighborhood formation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2857–2866.
[37] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web (WWW)*, 2015, pp. 1067–1077.
[38] J. Bilmes, "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Int. Comput. Sci. Inst., Univ. Berkeley, Berkeley, CA, USA, Tech. Rep. TR-97-021, 1997.
[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. 1st Int. Conf. Learn. Represent.*, Scottsdale, AZ, USA, 2013, pp. 1–12.
[40] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.* Stateline, NV, USA: Harrah's Lake Tahoe, 2013, pp. 3111–3119.
[41] S. Lamprier, S. Bourigault, and P. Gallinari, "Extracting diffusion channels from real-world social data: A delay-agnostic learning of transmission probabilities," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, 2015, pp. 178–185.
[42] P. F. Brown, V. J. D. Pietra, R. L. Mercer, S. A. D. Pietra, and J. C. Lai, "An estimate of an upper bound for the entropy of english," *Comput. Linguistics*, vol. 18, no. 1, pp. 31–40, 1992.
[43] J. Leskovec. *SNAP: Stanford Network Analysis Platform*. Accessed: Mar. 10, 2020. [Online]. Available: http://snap.stanford.edu/snap/index.html
[44] J. Leskovec, L. Backstrom, and J. Kleinberg. *MemeTracker: Download MemeTracker Data*. Accessed: Mar. 10, 2020. [Online]. Available: http://www.memetracker.org/data.html
[45] K. Lerman and R. Ghosh, "Information contagion: An empirical study of the spread of news on Digg and Twitter social networks," in *Proc. 4th AAAI Int. Conf. Weblogs Social Media*, Washington, DC, USA, 2010, pp. 90–97.
[46] G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2002, vol. 15, no. 4, pp. 833–840.
[47] M. G. Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2010, pp. 1019–1028.

QING BAO received the B.Sc. degree from the Department of Computer Science and Technology, East China Normal University, Shanghai, China, in 2011, and the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong, in 2016. She was a Postdoctoral Research Fellow with Hong Kong Baptist University. She is currently an Associate Professor with the School of Cyberspace, Hangzhou Dianzi University, China. Her research interests include Graph data mining, social network analysis and health informatics. She was a recipient of the Best Student Paper Award in the 2013 IEEE/WIC/ACM International Conference on Web Intelligence.

WILLIAM K. CHEUNG received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, in 1999. He is currently the Head and an Associate Professor of the Department of Computer Science, Hong Kong Baptist University. His current research interests include artificial intelligence, data mining, collaborative information filtering, social network analysis, and healthcare informatics. He has served as the Co-Chair and a Program Committee Member for a number of international conferences, as well as a Guest Editor of journals on areas, including *Artificial Intelligence*, *Web Intelligence*, *Data Mining*, *Web Services*, *E-Commerce Technologies*, and *Health Informatics*. He was on the Editorial Board of the IEEE Intelligent Informatics Bulletin, since 2002.

BENYUN SHI received the B.Sc. degree in mathematics from Hohai University, Nanjing, China, in 2003, and the M.Phil. and Ph.D. degrees in computer science from Hong Kong Baptist University, Hong Kong, in 2008 and 2012, respectively. He was a Full Professor with Hangzhou Dianzi University, China, and a Research Assistant Professor with Hong Kong Baptist University. He is currently a Professor with the School of Computer Science and Technology, Nanjing Tech University, China. His research interests include in the areas of data-driven modeling and analytics, machine learning, complex systems/networks, multiagent autonomy-oriented computing, computational epidemiology, and health informatics.



HONGJUN QIU received the B.Sc. degree in computer science and technology from Beijing Forestry University, Beijing, China, in 2003, and the Ph.D. degree in computer application from the Beijing University of Technology, Beijing, in 2010. She is currently a Lecturer with the School of Cyberspace, Hangzhou Dianzi University, China. Her research interests include complex systems/networks, autonomy-oriented computing, health informatics, and computational epidemiology.

LIJIA MA received the Ph.D. degree from Xidian University, Xi'an, China, in 2015. From October 2015 to October 2016, he was a Postdoctoral Researcher with Hong Kong Baptist University, Hong Kong, and from November 2016 to December 2017, he was also a Postdoctoral Researcher with Nanyang Technological University, Singapore. He is currently an Assistant Professor with the College of Computer and Software Engineering, Shenzhen University. His research interests mainly include evolutionary computation, machine learning, and complex networks.