

# Collaborative Inference of Coexisting Information Diffusions

Yanchao Sun<sup>1</sup>, Cong Qian<sup>1</sup>, Ning Yang<sup>1\*</sup>, Philip S. Yu<sup>2</sup>

<sup>1</sup>*School of Computer Science, Sichuan University, Chengdu, China*

*Email: {sunyc, congqian}@stu.scu.edu.cn, yangning@scu.edu.cn*

<sup>2</sup>*Department of Computer Science, University of Illinois at Chicago, Chicago, USA*

*Email: psyu@uic.edu*

**Abstract**—The purpose of *diffusion history inference* is to reconstruct the missing traces of information diffusion according to incomplete observations. Existing methods, however, often focus only on single diffusion trace, while in a real-world social network, there often coexist multiple information diffusions. In this paper, we propose a novel approach called Collaborative Inference Model (CIM) for the problem of the inference of coexisting information diffusions. CIM can holistically model multiple information diffusions without any prior assumption of diffusion models, and collaboratively infer the histories of the coexisting information diffusions via low-rank approximation with a fusion of heterogeneous constraints generated from additional data sources. We also propose an optimized algorithm called Time Window based Parallel Decomposition Algorithm (TWPDA) to speed up the inference without compromise on the accuracy. Extensive experiments are conducted on real-world datasets to verify the effectiveness and efficiency of CIM and TWPDA.

**Keywords**—Social network, Information diffusion, Sparse tensor approximation

## I. INTRODUCTION

The purpose of *diffusion history inference* is to reconstruct the missing information diffusion histories according to incomplete observations, which plays an important role in many applications [1]. For example, we are not likely to identify a rumor propagation over a network unless a significant number of users who transmitted it are noticed. In this case, it is essential to learn the diffusion history of the rumor for stopping its future spreading. Some approaches [1–3] have been proposed for the problem of diffusion history inference. The existing methods, however, often focus only on single information diffusion trace, while different information diffusions often coexist in real-world social networks, and a node often participates in multiple information diffusions at the same time period.

In this paper, we investigate the problem of *using sparse observations to infer the detailed histories of multiple information diffusions coexisting in a given network*, which is not easy due to the following challenges:

- (1) **Data Sparsity** Although massive data for information diffusions have been collected in the big data era, it is still likely that available observations of a single diffusion are sparse and insufficient, which makes it

extremely hard to infer the coexisting information diffusions separately.

- (2) **Lack of Prior Knowledge** In real world, information diffusion processes are so complicated that we seldom exactly know how information propagates and it is not appropriate to assume information diffusion follows a parametric model like Independent Cascade (IC) model and Susceptible-Infected (SI) model [3].
- (3) **Efficiency** In the big data era, a network often consists of billions of nodes and edges. Therefore, efficient algorithms are indispensable for the inference of coexisting information diffusions on a huge network.

In this paper, we propose a novel model, called Collaborative Inference Model (CIM), which exploits the synergism between the coexisting information diffusions to holistically model them as a 4th-order tensor, called Coexisting Diffusions Tensor (CDT), without priori assumptions of diffusion models, and fulfills the collaborative inference via a low-rank approximation of sparse CDT. To improve the inference accuracy, four heterogeneous constraints generated from additional data sources are fused with the decomposition. To make the inference efficient, we further propose an optimized decomposition algorithm, called Time Window based Parallel Decomposition Algorithm (TWPDA), which can speed up the decomposition in a parallel way without loss of approximation accuracy by making use of the temporal locality of information diffusions.

## II. PRELIMINARIES

Let  $C = \{c_1, \dots, c_M\}$  be the set of  $M$  memes who coexist over a network  $G$  consisting of  $N$  nodes  $\{v_1, \dots, v_N\}$  during  $Q$  time points  $\langle t_1, t_2, \dots, t_Q \rangle$ , where the term *meme* represents anything that can propagate over a social network, for example, it can be a label, a key word, or a news, etc.

**Definition 1:** An *infection*  $e$  is a 4-tuple  $(e.s, e.d, e.c, e.t)$ , which represents an infection of meme  $c \in C$  from source node  $s$  to destination node  $d$  ( $s, d \in V$ ) at time point  $t$ .

In our case, term *infection* is a generalized concept which represents the smallest blocks constituting a diffusion trace. For example, when one user retweets or comments on a tweet of another user, we can claim that the user is infected by the meme of the tweet. Note that an infection can happen

\* Ning Yang is the corresponding author.

more than once. For example, one may post tweets of the same meme many times in the same week.

*Definition 2:* An *information diffusion* of meme  $c_m$  ( $m = 1, 2, \dots, M$ ) is a set  $E^{(m)}$  consisting of infections whose meme is  $c_m$ , i.e.,  $E^{(m)} = \{e | e.c = c_m\}$ .

$E^{(m)}$  can be viewed as a sequence of temporal snapshots, i.e.,  $E^{(m)} = \bigcup_{q=1}^Q E_q^{(m)}$ , where  $E_q^{(m)}$  is the snapshot at time point  $t_q$ , i.e.,  $E_q^{(m)} = \{e | e \in E^{(m)} \text{ and } e.t = t_q\}$ .

### III. COLLABORATIVE INFERENCE MODEL

#### A. Coexisting Diffusions Tensor

A CDT  $\mathcal{A} \in \mathbb{R}^{N \times N \times M \times Q}$  consists of 4 modes which respectively represent  $N$  source nodes,  $N$  destination nodes,  $M$  memes that concurrently diffuse, and  $Q$  time points. A cell of the CDT,  $\mathcal{A}_{ijmq}$ , stores the frequency of the occurrences of an infection  $e$  of meme  $c_m$  from source node  $v_i$  to destination node  $v_j$  at time point  $t_q$ . Note that in a CDT, a node can be a source and a destination as well.

#### B. Sparse CDT Approximation

As CDT codes all the information, the problem of the inference of coexisting information diffusions can be reduced to the approximation of the sparse CDT  $\mathcal{A}$ . We construct a dense tensor  $\hat{\mathcal{A}}$  as the estimate of  $\mathcal{A}$  by a Tucker decomposition [4],

$$\mathcal{A} \approx \hat{\mathcal{A}} = \mathcal{G} \times_1 \mathbf{S} \times_2 \mathbf{D} \times_3 \mathbf{C} \times_4 \mathbf{T}, \quad (1)$$

i.e., a core tensor  $\mathcal{G} \in \mathbb{R}^{R \times R \times R \times R}$  multiplied by 4 latent factor matrices,  $\mathbf{S} \in \mathbb{R}^{N \times R}$ ,  $\mathbf{D} \in \mathbb{R}^{N \times R}$ ,  $\mathbf{C} \in \mathbb{R}^{M \times R}$ , and  $\mathbf{T} \in \mathbb{R}^{Q \times R}$ , along its 4 modes respectively, where  $R$  is the target rank, and the symbol  $\times_i$  ( $1 \leq i \leq 4$ ) stands for the tensor multiplication along the  $i$ th mode.

To achieve a higher approximation accuracy,  $\hat{\mathcal{A}}$  is produced by solving the following optimization problem:

$$\underset{\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T}}{\operatorname{argmin}} \mathcal{L}(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T}). \quad (2)$$

The objective function  $\mathcal{L}(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T})$  is defined as

$$\begin{aligned} \mathcal{L}(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T}) = & \frac{1}{2} \epsilon(\hat{\mathcal{A}}) + \frac{\lambda_1}{2} \gamma(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T}) \\ & + \frac{\lambda_2}{2} \phi(\mathbf{S}, \mathbf{D}) + \frac{\lambda_3}{2} \psi(\mathbf{D}, \mathbf{C}) \\ & + \frac{\lambda_4}{2} \xi(\mathbf{C}) + \frac{\lambda_5}{2} \tau(\mathbf{T}), \end{aligned} \quad (3)$$

where  $\lambda_1 \sim \lambda_5$  are the nonnegative parameters used to control the respective contributions of the terms. The first term  $\epsilon(\hat{\mathcal{A}})$  is the reconstruction error of the observable cells, which is defined as  $\epsilon(\hat{\mathcal{A}}) = \|\hat{\mathcal{A}}_\Omega - \mathcal{A}_\Omega\|_F^2$ , where  $\|\cdot\|_F$  stands for Frobenius norm of tensor, and  $\Omega$  represents the set of the indices of the observable tensor cells. The second term  $\gamma(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T})$  is the regularization constraints for avoiding overfitting, which is defined as  $\gamma(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T}) = \|\mathcal{G}\|_2^2 + \|\mathbf{S}\|_2^2 + \|\mathbf{D}\|_2^2 + \|\mathbf{C}\|_2^2 + \|\mathbf{T}\|_2^2$ ,

where  $\|\cdot\|_2$  stands for 2-norm of matrix. The last four terms of Equation (3) respectively represent four heterogeneous constraints which are described in the following subsection.

#### C. Heterogeneous Constraints

1) *Source-Destination Affinity Constraint:* An infection partly depends on the affinity between the source node and destination node, and the affinity strength is not reciprocal [5–7]. Based on this idea, we build a Source-Destination Affinity (SDA) matrix  $\mathbf{X} \in \mathbb{R}^{N \times N}$ .  $\mathbf{X}$  is asymmetric, and an element  $\mathbf{X}_{ij}$  indicates how likely node  $v_i$  will react to its neighbor node  $v_j$ ,  $1 \leq i, j \leq N$ , which is defined as

$$\mathbf{X}_{ij} = \frac{|\{v_p | f_{ip} < f_{ij}\}| + 0.5|\{v_p | f_{ip} = f_{ij}\}|}{|F_i|}, \quad (4)$$

where  $F_i$  is the neighbor set of  $v_i$ ,  $v_p \in F_i$ , and  $f_{ij}$  is the number of reactions from  $v_i$  to  $v_j$ .

$\mathbf{X}$  can be factorized as  $\mathbf{X} = \mathbf{S} \times \mathbf{D}^T$ , where  $\mathbf{S}$  and  $\mathbf{D}$  are exactly the latent feature matrices of source nodes and destination nodes of infections, and shared with the CDT decomposition shown in Equation (1). Thus the SDA constraint is defined as

$$\phi(\mathbf{S}, \mathbf{D}) = \|\mathbf{X} - \mathbf{S}\mathbf{D}^T\|_2^2. \quad (5)$$

The insight here is that the knowledge of coexisting information diffusions, which is represented by the CDT  $\hat{\mathcal{A}}$ , can fuse, through  $\mathbf{S}$  and  $\mathbf{D}$ , with the social affinity features of the nodes, which is represented by  $\mathbf{X}$ .

2) *Node-Meme Affinity Constraint:* Intuitively, one is likely to care more about specific memes than other memes. Based on this idea, we build a Node-Meme Affinity (NMA) matrix  $\mathbf{Y} \in \mathbb{R}^{N \times M}$ . An element  $\mathbf{Y}_{ij}$  ( $1 \leq i \leq N$ ,  $1 \leq j \leq M$ ) is defined as the proportion of the infections with meme  $c_j$  in the infections with destination  $v_i$ , i.e.,

$$\mathbf{Y}_{ij} = \frac{|\{e | e.d = v_i \wedge e.c = c_j\}|}{|\{e | e.d = v_i\}|}, \quad (6)$$

where  $e$  is an infection. As  $\mathbf{Y}$  carries the information of destination nodes and memes of infections, we can factorize it as  $\mathbf{Y} = \mathbf{D} \times \mathbf{C}^T$ , where  $\mathbf{D}$  and  $\mathbf{C}$  are the latent feature matrices of destination nodes and memes, respectively. Note that  $\mathbf{Y}$  shares  $\mathbf{D}$  and  $\mathbf{C}$  with Equation (1), so to fuse  $\mathbf{Y}$  with the decomposition, the NMA constraint is defined as

$$\psi(\mathbf{D}, \mathbf{C}) = \|\mathbf{Y} - \mathbf{D}\mathbf{C}^T\|_2^2. \quad (7)$$

3) *Meme-Correlation Constraint:* We often observe that related memes tend to share similar diffusion pattern. To capture the correlation between memes, we build a Meme Correlation (MC) matrix  $\mathbf{Z} \in \mathbb{R}^{M \times M}$ . An element  $\mathbf{Z}_{ij}$  is defined as the proportion of co-occurrence of meme  $c_i$  and  $c_j$ , i.e.,

$$\mathbf{Z}_{ij} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}, \quad (8)$$

where  $N_x = \{v | v \in V \wedge e.d = v \wedge e.c = c_x\}$  is the set of nodes infected by meme  $c_x$ ,  $x = i, j$ . Reasonably, if  $\mathbf{Z}_{ij}$  is large, the distance between meme latent feature vectors  $\mathbf{C}_{i*}$  and  $\mathbf{C}_{j*}$  (the  $i$ -th and  $j$ -th row vector of  $\mathbf{C}$ ) should be small, which leads to the following MC constraint:

$$\xi(\mathbf{C}) = \text{tr}(\mathbf{C}^T(\mathbf{K} - \mathbf{Z})\mathbf{C}), \quad (9)$$

where  $\mathbf{K} \in \mathbb{R}^{M \times M}$  is a diagonal matrix with diagonal elements  $\mathbf{K}_{ii} = \sum_j \mathbf{Z}_{ij}$ .

4) *Temporal Smoothness Constraint*: Like heat diffusion, information always diffuses gradually on networks without drastic change of node status [8–10], which exhibits a property of Temporal Smoothness (TS). Therefore we impose on the CDT decomposition the TS constraint  $\tau(\mathbf{T})$  which is defined as:

$$\tau(\mathbf{T}) = \|\mathbf{T} - \mathbf{U}\mathbf{T}\|_2^2, \quad (10)$$

where  $\mathbf{U} \in \mathbb{R}^{Q \times Q}$  is the temporal smoothing matrix with 0 elements except for  $\mathbf{U}_{i,i+1} = 1$ ,  $1 \leq i \leq Q - 1$ . By minimizing  $\tau(\mathbf{T})$ , it is guaranteed that the temporal latent features at two successive time points are similar.

#### IV. DECOMPOSITION ALGORITHMS

##### A. Native Decomposition Algorithm

As there is no closed-form solution to Equation (2), we first propose a Native Decomposition Algorithm (NDA) based on gradient descent to find a local minima. The outline of NDA is given in Algorithm 1, where nonzero cells of the sparse CDT are updated along the gradient direction until the objective function converges (Lines 6 to 16). NDA performs the decomposition on the whole tensor (that is why it is called 'native'), which makes it unable to scale up to a big CDT with millions cells.

##### B. Time Window based Parallel Decomposition Algorithm

Recent researches reveal that information diffusions in social networks exhibit a property of *temporal locality* [11], which implies that in a CDT, a slice may has less to do with the slices at time points far away. Inspired by this idea, we propose a Time Window based Parallel Decomposition Algorithm (TWPDA) for solving Equation (2). TWPDA employs a sliding time window scheme, where the width of each time window is determined adaptively and separately. With respect to the time windows, TWPDA splits a CDT into a series of sub-tensors with different sizes along time dimension.

1) *Sliding Time Window and Sub-tensor*: Let  $T = \langle t_1, t_2, \dots, t_Q \rangle$  be a sequence of  $Q$  time points considered, and then a sliding time window over  $T$  and its corresponding sub-tensor can be defined as follows:

*Definition 3*: A *sliding time window* is an interval, denoted by  $[s, q]$ , starting in time point  $t_s$  and ending in time point  $t_q$ ,  $1 \leq s \leq q \leq Q$ , such that if  $[s', q']$  is the successive time window of  $[s, q]$ ,  $s' = s + 1$  and  $q' > q$ .

---

##### Algorithm 1 NDA( $\mathcal{A}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U}, R, \epsilon$ )

---

**Input:**  $\mathcal{A}$ : the sparse CDT;  $\mathbf{X}$ : the SDA matrix;  $\mathbf{Y}$ : the NMA matrix;  $\mathbf{Z}$ : the MC matrix;  $R$ : the target rank;  $\epsilon$ : the threshold of error;

**Output:**  $\hat{\mathcal{A}}$ : the approximated CDT;

```

1: Randomly initialize matrices  $\mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T}$ , and core tensor  $\mathcal{G}$ ;
2: Set  $\eta$  as step size;
3: Set  $\mathbf{K}_{ii} = \sum_{j=1}^M \mathbf{Z}_{ij}$  for  $1 \leq i \leq M$ ;
4:  $\mathbf{L}_Z = \mathbf{K} - \mathbf{Z}$ ;
5:  $loss_a = \mathcal{L}(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T})$ ;
6: repeat
7:   for all  $\mathcal{A}_{ijkl} \neq 0$  do
8:      $\mathcal{B}_{ijkl} = \mathcal{G} \times_1 \mathbf{S}_{i*} \times_2 \mathbf{D}_{j*} \times_3 \mathbf{C}_{k*} \times_4 \mathbf{T}_{l*}$ ;
9:      $\mathbf{S}_{i*} = \mathbf{S}_{i*} - \eta \lambda_5 \mathbf{S}_{i*} - \eta \lambda_1 (\mathbf{S}_{i*} \times \mathbf{D}^T - \mathbf{X}_{i*}) \times \mathbf{D}$ 
        $- \eta (\mathcal{B}_{ijkl} - \mathcal{A}_{ijkl}) \times \mathcal{G} \times_2 \mathbf{D} \times_3 \mathbf{C} \times_4 \mathbf{T}$ ;
10:     $\mathbf{D}_{j*} = \mathbf{D}_{j*} - \eta \lambda_5 \mathbf{D}_{j*} - \eta \lambda_1 (\mathbf{S} \times \mathbf{D}_{j*}^T - \mathbf{X}_{*j})^T \times \mathbf{S}$ 
        $- \eta \lambda_2 (\mathbf{D}_{j*} \times \mathbf{C}^T - \mathbf{Y}_{*j}) \times \mathbf{C}$ 
        $- \eta (\mathcal{B}_{ijkl} - \mathcal{A}_{ijkl}) \times \mathcal{G} \times_1 \mathbf{S} \times_3 \mathbf{C} \times_4 \mathbf{T}$ ;
11:     $\mathbf{C}_{k*} = \mathbf{C}_{k*} - \eta \lambda_5 \mathbf{C}_{k*} - \eta \lambda_2 (\mathbf{D} \times \mathbf{C}_{k*}^T - \mathbf{Y}_{*k})^T \times \mathbf{D}$ 
        $- \eta \lambda_3 (\mathbf{L}_Z \times \mathbf{C})_{k*}$ 
        $- \eta (\mathcal{B}_{ijkl} - \mathcal{A}_{ijkl}) \times \mathcal{G} \times_1 \mathbf{S} \times_2 \mathbf{D} \times_4 \mathbf{T}$ ;
12:     $\mathbf{T}_{l*} = \mathbf{T}_{l*} - \eta \lambda_5 \mathbf{T}_{l*} - \eta \lambda_4 (\mathbf{T}_{l*} - \mathbf{T}_{l*} \times \mathbf{U}) \times (\mathbf{I} - \mathbf{U})^T$ 
        $- \eta (\mathcal{B}_{ijkl} - \mathcal{A}_{ijkl}) \times \mathcal{G} \times_1 \mathbf{S} \times_2 \mathbf{D} \times_3 \mathbf{C}$ ;
13:     $\mathcal{G} = \mathcal{G} - \eta \lambda_5 \mathcal{G}$ 
        $- \eta (\mathcal{B}_{ijkl} - \mathcal{A}_{ijkl}) \times \mathbf{S}_{i*} \circ \mathbf{D}_{j*} \circ \mathbf{C}_{k*} \circ \mathbf{T}_{l*}$ ;
14:   end for
15:    $loss_b = \mathcal{L}(\mathcal{G}, \mathbf{S}, \mathbf{D}, \mathbf{C}, \mathbf{T})$ ;
16: until  $|loss_a - loss_b| < \epsilon$ 
17:  $\hat{\mathcal{A}} = \mathcal{G} \times_1 \mathbf{S} \times_2 \mathbf{D} \times_3 \mathbf{C} \times_4 \mathbf{T}$ ;

```

---

*Definition 4*: The *sub-tensor* corresponding to time window  $[s, q]$  ( $1 \leq s \leq q \leq Q$ ), denoted by  $\mathcal{A}^{[s,q]}$ , consists of the  $q - s + 1$  slices  $\mathcal{A}_{***i}$  ( $i = s, s + 1, \dots, q$ ) concatenated along the time dimension.

For example, in Fig. 1, there are 3 sliding time windows,  $[1, 3]$ ,  $[2, 5]$ , and  $[3, 6]$ , and their corresponding sub-tensors are  $\mathcal{A}^{[1,3]}$ ,  $\mathcal{A}^{[2,5]}$ , and  $\mathcal{A}^{[3,6]}$ . Note that successive sub-tensors are overlapped.

2) *Determining Time Window Width*: As the observations are sparse and distributed nonuniformly, if the time windows share a fixed width, some sub-tensors may have insufficient observed data for inference while some ones may possess an excessive amount of data leading to overfitting. To address this issue, TWPDA adaptively determines the width of a time window with respect to the number of non-zero cells that the corresponding sub-tensor has, so that the denser (sparser) the observed infections, the narrower (wider) the time window.

Let  $nnz(\mathcal{A}^{[s,q]})$  be the number of non-zero cells of sub-tensor  $\mathcal{A}^{[s,q]}$ . Then the width of the time window starting in time point  $t_i$ , denoted by  $\omega_i$ , is evaluated by the following

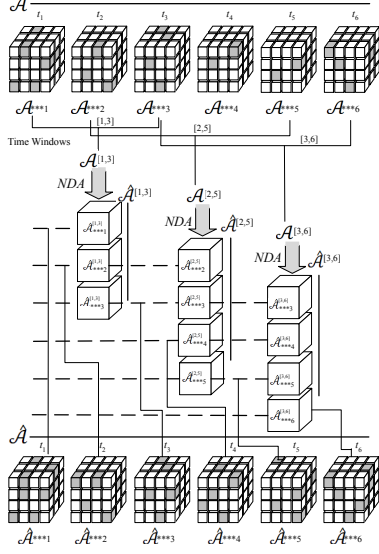


Figure 1. Illustration of TWPDA.

function:

$$\omega_i = f(\alpha_i) = \begin{cases} Q - i + 1, & \alpha_i \geq Q - i + 1 \\ \alpha_i, & nnz(\mathcal{A}^{[i, (i+\alpha_i-1)]}) \geq \beta, \\ f(\alpha_i + 1), & nnz(\mathcal{A}^{[i, (i+\alpha_i-1)]}) < \beta \end{cases} \quad (11)$$

where  $\alpha_i = \omega_{i-1}$  is the initial width of the time window starting in time point  $t_i$ , and  $\beta$  is the given threshold of the number of non-zero cells.

Fig. 1 shows some time windows with different widths. In Fig. 1, the time window starting in  $t_2$  and the time window starting in  $t_3$  are stretched, since the slices  $\mathcal{A}^{***4}$  and  $\mathcal{A}^{***5}$  are pretty sparse.

3) *Generating Final Result*: As sub-tensors are overlapped, so to reasonably generate the final approximated CDT, we take the weighted average of the multiple approximations of a shared slice as its final result, and according to the temporal locality, it is reasonable that the wider the time window, the smaller the weight of the approximation produced by the sub-tensor corresponding to that time window. Based on this idea, the final approximation of a time slice  $\mathcal{A}^{***k}$  is evaluated as

$$\hat{\mathcal{A}}^{***k} = \frac{1}{W} \sum_{\substack{\mathcal{A}^{[s,q]} \\ \text{s.t. } s \leq k \leq q}} w^{[s,q]} \times \hat{\mathcal{A}}^{[s,q]}, \quad (12)$$

where  $\hat{\mathcal{A}}^{[s,q]}$  is the approximation of  $\mathcal{A}^{***k}$  that is contained in the approximated sub-tensor  $\hat{\mathcal{A}}^{[s,q]}$ ,  $w^{[s,q]} = \frac{1}{2^{[q-s+1]}}$  is the weight of  $\hat{\mathcal{A}}^{[s,q]}$ , and  $W = \sum w^{[s,q]}$  is the regularization factor.

4) *Outline of TWPDA*: The outline of TWPDA is given in Algorithm 2, where the major computation is invoking

---

**Algorithm 2** TWPDA( $\mathcal{A}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{U}, R, \epsilon, \alpha_1, \beta$ )

---

**Input:**  $\mathcal{A}$ : the sparse CDT;  $\mathbf{X}$ : the SDA matrix;  $\mathbf{Y}$ : the NMA matrix;  $\mathbf{Z}$ : the MC matrix;  $\mathbf{U}$ : the TS matrix;  $R$ : the target rank;  $\epsilon$ : the threshold of error;  $\alpha_1$ : the initial width of first time window;  $\beta$ : the threshold of the number of non-zero cells;

**Output:**  $\hat{\mathcal{A}}$ : the approximated CDT;

- 1: Initialize the cells of  $\hat{\mathcal{A}}$  with zero;
  - 2: Generate the sliding time windows  $\{[s, q]\}$  by using Equation (11) with  $\alpha_1$  and  $\beta$ ;
  - 3: Construct the sub-tensors  $\{\mathcal{A}^{[s,q]}\}$ ;
  - 4: Generate  $\{\hat{\mathcal{A}}^{[s,q]}\}$  in parallel, by invoking *NDA* with  $\mathcal{A}^{[s,q]}$ ,  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$ ,  $\mathbf{U}^{[s,q]}$ ,  $R$  and  $\epsilon$ , where  $\mathbf{U}^{[s,q]}$  is the sub-matrix consisting of rows from  $\mathbf{U}_{s*}$  to  $\mathbf{U}_{q*}$ ;
  - 5: Generate in parallel the final result  $\hat{\mathcal{A}}^{***k}$ ,  $1 \leq k \leq Q$ , according to Equation (12);
  - 6: Generate  $\hat{\mathcal{A}}$  by concatenate the approximated time slices  $\{\hat{\mathcal{A}}^{***k}\}$  along the time dimension;
- 

*NDA* (shown in Algorithm 1) in parallel for each sub-tensor to generate the approximation (Line 4).

## V. EXPERIMENTS

The experiments are conducted on a Spark cluster consisting of 3 PCs where each PC equipped with a 2.7 GHz INTEL CPU of 4 cores and 128 GB RAM, and all the programs are written with MATLAB 2016b.

### A. Experiment Setting

1) *Datasets*: **Twitter** We choose the Twitter dataset released by Zhang *et al.* [13], which consists of 8 million tweets posted by 5,000 users during the period from Jan. 2009 to Dec. 2012. From the Twitter dataset, we extract 5 memes and construct a CDT of  $\mathbb{R}^{5000 \times 5000 \times 5 \times 48}$  for test, where the time dimension consists of 48 months. We use the Foursquare dataset also released by Zhang *et al.* [13] as the additional data source, which shares the users with the Twitter dataset and from which we build the heterogeneous constraint matrices  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$ .

**Weibo** The Sina Weibo network is the largest microblogging website in China. The first Weibo dataset is crawled with the method proposed Kong *et al.* [14], which consists of 80,000 tweets posted by 3,000 users in May, 2014. From the firsts Weibo dataset, we also extract 5 memes and construct a CDT of  $\mathbb{R}^{3000 \times 3000 \times 5 \times 31}$  for test, where the time dimension consists of 31 days. We also crawled the second Weibo dataset for the same users as the additional data source from which we build the heterogeneous constraint matrices  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$ . In the second Weibo dataset, the tweets are posted in April 2014, which are before the time of the tweets in the first Weibo dataset.

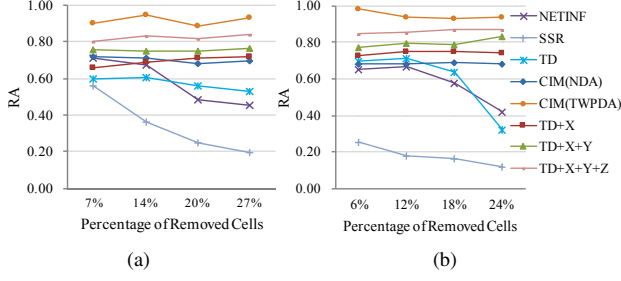


Figure 2. RA on (a) Twitter and (b) Weibo

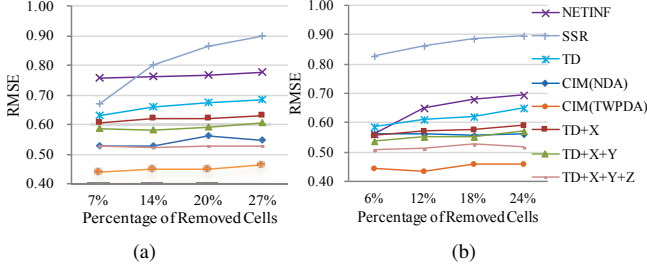


Figure 3. RMSE on (a) Twitter and (b) Weibo

2) *Metrics*: To evaluate the effectiveness of the proposed CIM, we use two metrics, recovery accuracy (RA) and root mean square error (RMSE). Let  $\mathcal{A}_{ijmq}$  be a non-zero cell whose original value is removed as the ground truth and  $\hat{\mathcal{A}}_{ijmq}$  be its estimate, and then RA and RMSE are defined as follows:

$$RA = \frac{1}{S} \sum_{i,j,m,q} \mathbb{I}(\mathcal{A}_{ijmq} \hat{\mathcal{A}}_{ijmq}), \quad (13)$$

$$RMSE = \sqrt{\frac{1}{S} \sum_{i,j,m,q} (\mathcal{A}_{ijmq} - \hat{\mathcal{A}}_{ijmq})^2}, \quad (14)$$

where  $S$  is the number of cells removed for test, and  $\mathbb{I}(x) = 1$  if  $x > 0$ , otherwise  $\mathbb{I}(x) = 0$ .

3) *Baseline Methods*: We use two existing representative algorithms, NETINF [15] and SSR [2], as the baseline methods. Since they are designed for the reconstruction of a single diffusion trace, we apply them to reconstruct the multiple testing diffusion traces one by one, and evaluate their respective performance by the average of the results over the testing diffusion traces. To verify the effectiveness of the heterogeneous constraints ( $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$  and  $\mathbf{T}$ ), we also compare CIM to Tucker Decomposition (TD) [4] and the combinations of TD and subsets of the constraints, i.e., TD+X, TD+X+Y, and TD+X+Y+Z, where each method adds one more constraint to the previous one.

## B. Experiment Results

1) *Effectiveness of CIM*: We randomly remove different percentages of cells and use their original values as the ground truth, and the setting of parameters is  $R = 3$ ,

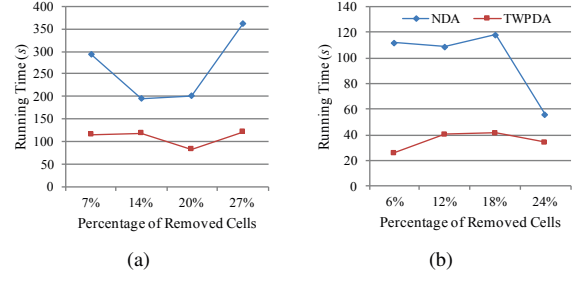


Figure 4. Time Cost of NDA and TWPDA on (a) Twitter and (b) Weibo

$\epsilon = 0.01$ ,  $\lambda_1 = \lambda_2 = 1$ ,  $\lambda_3 = 0.3$ ,  $\lambda_4 = \lambda_5 = 0.05$ , and  $\eta = 0.001$ . Fig. 2 and Fig. 3 show the RA and RMSE of CIM and the baseline methods, respectively, from which we can find that on each scale of the removed cells of both Twitter and Weibo, the RA and RMSE of CIM (with TWPDA or NDA) are obviously higher and lower than those of the baseline methods, respectively, which indicates that CIM can recover unobserved infections more correctly and also estimate their frequency more accurately. This is because CIM takes advantage of the synergism between the coexisting information diffusions and collaboratively infers them via a low-rank approximation of a CDT.

What is more, the curves of TD, TD+X, TD+X+Y, TD+X+Y+Z, and CIM show that the more number of constraints is taken into consideration, the higher the RA and the lower the RMSE, due to the fact that by fusing the additional heterogeneous constraints into the decomposition of CDT, CIM reduces the uncertainty of CDT significantly.

We can also note that with the increase of the scale of the removed cells, the performance of CIM keeps approximately stable, while the performance of alternative methods drops markedly. Since the increasing number of the removed cells leads to the sparser data, this result indicates that CIM outperforms the alternative methods on sparse data due to its collaborative reconstruction which utilizes the correlation of coexisting information diffusions via the low-rank approximation of sparse CDT.

2) *Efficiency of TWPDA*: From Fig. 4 we can see that the running time of TWPDA is significantly less than that of NDA on both Twitter and Weibo. This efficiency gain of TWPDA is again due to the sliding time window scheme which enables a parallel decomposition of the sub-tensors. One can refer to the full paper [12] for more details about the parallelizability of TWPDA.

## VI. RELATED WORK

Gomez-Rodriguez *et al* [15], propose an approach called NETINF, which models an information cascade as a tree, and builds a generative model to recover the underlying diffusion network structure with a greedy strategy. Sefer *et al.* [1] reduce the problem of diffusion history inference to the problem of determining the maximum likelihood

history given diffusion snapshots and propose an algorithm called DHR-sub (sub-modular history reconstruction on discrete dynamics) which reconstructs the history by greedily maximizing the non-monotone sub-modular log-likelihood at each time step. Chen *et al.* [2] formulate the diffusion history inference problem as a maximum a posteriori (MAP) estimate problem, and propose a greedy and step-by-step reconstruction algorithm called SSR to infer the most likely historical diffusion trace. Rozenshtein *et al.* [3] consider this problem in a different way, which models an information diffusion as a temporal Steiner-tree, and recover its historical spread flow by searching a temporal Steiner-tree with minimum cost defined in advance. However, the existing methods often focus on single diffusion trace and cannot model multiple diffusion traces holistically and recover their missing histories in a synergistic way.

## VII. CONCLUSION

In this paper, we propose an approach called Collaborative Inference Model (CIM), which can holistically model multiple information diffusion traces as a sparse Coexisting Diffusions Tensor (CDT), and can collaboratively infer the histories of the coexisting information diffusions. To speed up the inference without compromise on the accuracy, we further propose an optimized algorithm called Time Window based Parallel Decomposition Algorithm (TWPDA) as the core component of CIM. The extensive experiments conducted on real world datasets and synthetic datasets verify the effectiveness and the efficiency of CIM and TWPDA.

## ACKNOWLEDGMENT

This work is supported by NSFC through grant 61173099, and in part by NSF through grants IIS-1526499 and CNS-1626432.

## REFERENCES

- [1] E. Sefer and C. Kingsford, "Diffusion archaeology for diffusion progression history reconstruction," in *Proceedings of 2014 IEEE International Conference on Data Mining*, 2014, pp. 530–539.
- [2] Z. Chen, H. Tong, and L. Ying, "Full diffusion history reconstruction in networks," in *Proceedings of 2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 707–716.
- [3] P. Rozenshtein, A. Gionis, B. A. Prakash, and J. Vreeken, "Reconstructing an epidemic over time," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1835–1844.
- [4] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [5] N. J. Yuan, Y. Zhong, F. Zhang, X. Xie, C.-Y. Lin, and Y. Rui, "Who will reply to/retweet this tweet?: The dynamics of intimacy from online social interactions," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, pp. 3–12.
- [6] J. Schantl, R. Kaiser, C. Wagner, and M. Strohmaier, "The utility of social and topical factors in anticipating repliers in twitter conversations," in *Proceedings of the 5th Annual ACM Web Science Conference*, 2013, pp. 376–385.
- [7] R. Xiang, J. Neville, and M. Rogati, "Modeling relationship strength in online social networks," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 981–990.
- [8] H. Gui, Y. Sun, J. Han, and G. Brova, "Modeling topic diffusion in multi-relational bibliographic information networks," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 649–658.
- [9] S. Segarra, W. Huang, and A. Ribeiro, "Diffusion and superposition distances for signals supported on networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 1, pp. 20–32, 2015.
- [10] J. Zhang, C. Wang, J. Wang, and J. X. Yu, "Inferring continuous dynamic social influence and personal preference for temporal behavior prediction," *Proc. VLDB Endow.*, vol. 8, no. 3, pp. 269–280, 2014.
- [11] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng, "Spatio-temporal dynamics of online memes: A study of geo-tagged tweets," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 667–678.
- [12] Y. Sun, C. Qian, N. Yang, and P. S. Yu, "Collaborative inference of coexisting information diffusions," *arXiv preprint arXiv:1708.06890*, 2017.
- [13] J. Zhang, X. Kong, and P. S. Yu, "Predicting social links for new users across aligned heterogeneous social networks," in *Proceeding of 2013 IEEE 13th International Conference on Data Mining*, 2013, pp. 1289–1294.
- [14] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, 2013, pp. 179–188.
- [15] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," *ACM Transaction on Knowledge Discovery from Data*, vol. 5, no. 4, pp. 21:1–21:37, 2012.