Under consideration for publication in Knowledge and Information
Systems

Diffusion Archaeology for Diffusion Progression History Reconstruction

Emre Sefer[1] and Carl Kingsford[2]

[1] Machine Learning Department, School of Computer Science,
[2] Computational Biology Department, Carnegie Mellon University, Pittsburgh PA, USA

Abstract. Diffusion through graphs can be used to model many real-world process, such as the spread of diseases, social network memes, computer viruses, or water contaminants. Often, a real-world diffusion cannot be directly observed while it is occurring — perhaps it is not noticed until some time has passed, continuous monitoring is too costly, or privacy concerns limit data access. This leads to the need to reconstruct how the present state of the diffusion came to be from partial diffusion data. Here, we tackle the problem of reconstructing a diffusion history from one or more snapshots of the diffusion state. This ability can be invaluable to learn when certain computer nodes are infected or which people are the initial disease spreaders to control future diffusions. We formulate this problem over discrete-time SEIRS-type diffusion models in terms of maximum likelihood. We design methods that are based on submodularity and a novel prize-collecting dominating-set vertex cover (PCDSVC) relaxation that can identify likely diffusion steps with some provable performance guarantees. Our methods are the first to be able to reconstruct complete diffusion histories accurately in real and simulated situations. As a special case, they can also identify the initial spreaders better than the existing methods for that problem. Our results for both meme and contaminant diffusion show that the partial diffusion data problem can be overcome with proper modeling and methods, and that hidden temporal characteristics of diffusion can be predicted from limited data.

Keywords: Diffusion; SEIRS Dynamics; Diffusion History Reconstruction; Social Networks

1. Introduction

Dynamic processes over networks are used to model and analyze properties of various social and biological systems. Diffusion is special case of those processes

Received xxx
Revised xxx
Accepted xxx

in which a spread (e.g., an infection) starts from some part of the graph and spreads to other portions over time via edges of the graph. Some examples are virus propagation in computer networks (Serazzi and Zanero, 2003), and idea and gossip spreading in social networks (Yang and Leskovec, 2011). A diffusion model, such as the commonly studied SIRS and SEIRS models, defines the set of possible states that the nodes of the graph can be in and rules for probabilistically switching between states. Recently, (Prakash, Chakrabarti, Faloutsos, Valler and Faloutsos, 2011) introduced the *VPM* (Virus Propagation Model) that generalizes all those Markovian diffusion models and defines the hierarchical relationships between them.

It is not always easy to know the whole diffusion progression, initial diffusion conditions, or the time it started due to several limitations. For example, existence of a computer virus diffusion over the computer network may only be noticed after a significant number of computers stop operating. A similar problem exists in detecting influenza diffusion (Salathé, Kazandjieva, Lee, Levis, Feldman and Jones, 2010). We may also not track the diffusion of a virus in email networks and a contaminant in a water distribution network due to privacy and physical limitations, respectively. In all these cases, it is essential to learn more about the past to take precautions to prevent future epidemics, to learn more about the true diffusion mechanics, to provide safer water, to break privacy and so on. However, given present-day diffusion data, it is not trivial to search for the most likely diffusion progression in the past as there will be many valid histories leading to the observed data.

In this paper, we tackle this problem of inferring a complete diffusion history from one or more diffusion snapshots for discrete-time SEIRS-type diffusion models that include SI, SIS, SIR, SIRS, SEIR, SEIS, SEIRS. Those models with their abstract states and independent cascade (IC) assumption (Kempe, Kleinberg and Tardos, 2003) have been used to model many forms of diffusion in different domains (Yang and Leskovec, 2011; Salathé et al., 2010; Leskovec, Adamic and Huberman, 2007).

Complete diffusion history reconstruction has not been previously studied but similar problems exist in the literature. The most relevant such problem is *Initial Spreader Identification* where we want to identify the most probable initial infected nodes that started a diffusion. Among approaches for this problem, *Keffectors* (Lappas, Terzi, Gunopulos and Mannila, 2010) identifies the k best-possible initial spreaders. However, it requires an estimate of the number of initial spreaders to be given as input. *Rumor* (Shah and Zaman, 2011) finds the most probable spreader by estimating the rumor centrality but it assumes a single initial spreader, and it is only defined for SI model. Lastly, *NetSleuth* (Prakash, Vreeken and Faloutsos, 2012) does not require the number of initial spreaders as an input. However, it works only for the restricted cases of SI model, and it is based on MDL principle without any provable performance guarantee. None of these methods infer the whole diffusion progression, as our approaches do. Another related problem is *Graph Inference* where we want to reconstruct the unknown graph from observed multiple diffusion traces over it. This problem is fundamentally different than the history reconstruction problem as graph inference methods (Gomez-Rodriguez, Leskovec and Krause, 2012; Gomez-Rodriguez, Leskovec and Schölkopf, 2013) search in the graph space assuming full observability of multiple traces whereas our methods search in temporal diffusion progression space as they try to complete the missing history of a single trace.

We formulate the diffusion history reconstruction problem as that of determining the maximum likelihood (ML) history given diffusion snapshots that may come from multiple time points. We designed an algorithm called *DHR-sub* (submodular history reconstruction on discrete dynamics) that reconstructs the history before the earliest measured time point by greedily maximizing the non-monotone submodular log-likelihood at each previous time step. It further reconstructs the history between the consecutive diffusion data time points by solving the problem as non-monotone submodular maximization under matroid base constraints.

Though accurate and practical for smaller graphs, *DHR-sub* can take some time to solve. To reconstruct diffusion history faster, we designed *DHR-pcdsvc* that solves the first-order Taylor approximation relaxation of the log-likelihood. We define this new problem as *Prize-Collecting Dominating-Set Vertex Cover*, and show that it can be approximated within a factor of $O(\log(|V|))$. This problem can be further relaxed by removing the covering constraints; it becomes *Prize-Collecting Vertex Cover*, and we design *DHR-pcvc* that approximates it by a factor of 2 for non-bipartite models and solves this newer relaxation optimally by transforming it to s-t mincut for bipartite models. We also design ensemble approaches for all of our methods that estimate the robust set of initial spreaders from multiple runs of the algorithm.

In summary, our main contributions are:

- Our methods reconstruct the whole diffusion history nonparametrically for all SEIRS-type models whereas the existing methods only identify the initial spreaders for certain models;
- Our methods formulate the problem in terms of diffusion likelihood, and we give some performance guarantees on the quality of the obtained solutions;
- Relaxation methods *DHR-pcdsvc* and *DHR-pcvc* scale well to history reconstruction over tens of thousands of nodes with provable performance guarantees;
- Our methods perform better by using the diffusion information from all the nodes (not just from infected nodes), and from multiple time points if available;
- We use reconstructed histories to predict several diffusion features such as speed and acceleration that are not apparent in the observed portion of the diffusion.

Our methods more accurately identify initial spreader sites on a water distribution network and on simulated networks. In terms of history reconstruction, we compared our methods with a baseline heuristic since there is no previous method. All our methods can accurately reconstruct several meme diffusion histories on blog networks. They also perform better on synthetic networks under different models. In general, all our methods reconstruct the diffusion history reasonably fast and accurately compared to the hardness of the problem (see Section 7.7). In many cases, relaxations of the original problem can reconstruct the diffusion history almost as good as the original formulations in a far shorter amount of time. Lastly, we also estimate the speed and acceleration dynamics of several memes over blog network from their reconstructed histories. In this case, estimated dynamics from quite a few diffusion snapshots match the true dynamics almost perfectly producing decent whole history reconstruction performance. Overall, our results for different types of diffusion show that many characteristics of complete diffusion history can be inferred with proper modeling and methods.

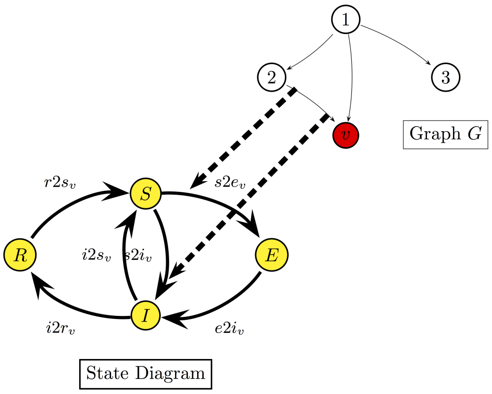


Fig. 1. SEIRS State Transition Diagram

Symbol	Definition and Description
$G = (V, E)$	directed graph G
$P(v)$, $S(v)$	set of predecessors, successsors of node v
$\mathcal{S}$	SEIRS states (S,E,I,R)
$\mathcal{M}$	diffusion model of SEIRS-type models
p_{uv}	probability of diffusion from infected node u to susceptible node v
$e2i_v$, $i2r_v$, $r2s_v$, $s2e_v$, $s2i_v$, $i2s_v$	probability of ($E \rightarrow I$, $I \rightarrow R$, $R \rightarrow S$, $S \rightarrow E$ $S \rightarrow I$, $I \rightarrow S$) transition for v
t^s_v, t^e_v, t^i_v, t^r_v	time v transitions into (S, E, I, R)
S_t, E_t, I_t, R_t	set of nodes that are in (S, E, I, R) at time t
D_t	diffusion snapshot at time t
$\mathcal{D}$	given diffusion data
l_D	length of diffusion D
T_D	ordered set of time points of $\mathcal{D}$
t_{min}, t_{max}	$\min(T_D)$, $\max(T_D)$
f_{min}, f_{max}	$\min(T_D)/l_D$, $\max(T_D)/l_D$

Table 1. Table of Symbols

2. SEIRS Diffusion Dynamics

SEIRS diffusion dynamics over directed graph $G = (V, E)$ with possible state transitions is shown in Figure 1. The SEIRS states are Susceptible (S), Exposed but not contagious (E), Infected and contagious (I), and previously infected but now Recovered (or immune to the infection) (R). Those states are general enough abstractions to model various forms of diffusion in different contexts (Leskovec et al., 2007; Salathé et al., 2010). For instance, the infected state models people having influenza symptoms in influenza diffusion over humans, and it represents creation of a blog entry about a topic in idea diffusion. Similarly, the recovered state could represent recovery of a person from influenza or the decontamination of a water tower from chemical contaminants depending on the context.

In SEIRS model, diffusion starts at time $t = 0$ from set of initially infected nodes and progresses over G in discrete time steps. Let S_t, E_t, I_t, R_t be the set of S, E, I, R nodes at time t respectively. At each time step, infected nodes spread the infection to the susceptible nodes with certain probability. This $S \rightarrow E$ transition is exogenous; it is affected by G and probability of exogenous transition for susceptible node v at time t is $1 - \prod_{u \in P(v) \cap I_t}(1 - p_{uv})$, where $P(v)$ is the set of nodes with edges into v and p_{uv} is the probability of transmission of the agent over edge (u, v). The remaining $E \rightarrow I$, $I \rightarrow R$, $R \rightarrow S$ transitions are endogenous; their transition probabilities are $e2i_v$, $i2r_v$, $r2s_v$ respectively, and they are not affected by G. For every node at each time step, if a transition succeeds, the node transitions to a new state. Otherwise, it follows similar procedure at next time step, independent of the previous trials. SEIRS-type models are Markovian since state of a node at time t depends on its state and its neighbors' states at previous time steps, and it obeys independent cascade (IC) (Kempe et al., 2003) assumption which states that a diffusion from one of nodes predecessor is enough for node to become exposed/infected. The symbols used in this text are given in Table 1 for reference.

SEIRS-type models include the well-known SI, SIR, SIS, SIRS, SEIR, SEIRS models (Hethcote, 2000). SEIRS is the most general model among these models, and some of its transitions disappear or change slightly in other models. For instance, in SIR, there is no exposed state; the exogenous transition is $S \rightarrow I$ since nodes proceed directly to the infected state, and there is no $R \rightarrow S$ transition. We can classify SEIRS-type models in various ways. SIRS, SEIRS are *loopy* models where $R \rightarrow S$ transition is available whereas SI, SIR, SEIR are *non-loopy* models. We can also split SEIRS-type models into *bipartite* and *non-bipartite* models: a node that gets the infection directly transitions into infected state in non-bipartite models such as SI, SIR, SIRS, SIS whereas it goes through the exposed state for bipartite models such as SEIR, SEIRS. The model may also be either *uniform* in which case all of the transition probabilities are the same for each edge and node, or *non-uniform* in which case the probabilities may vary over the edges and nodes. We discuss the general non-uniform case here; the uniform case is a simple specialization.

3. Diffusion History Reconstruction Problem

For diffusion D, let $D_t = (S_t, E_t, I_t, R_t)$ be the state of the nodes at the time t, where S_t is the set of susceptible nodes, etc. D_t is a *diffusion snapshot.* We define Problem 1 to reconstruct the diffusion history when the diffusion length is unknown:

Problem 1. We are given: a graph $G = (V, E)$, state transition probabilities (p_{uv}, $e2i_v$, $i2s_v$, $i2r_v$, $r2s_v$) that define an SEIRS-type model, a collection of time points T_D at which snapshots were taken, and a collection of diffusion snapshots $\mathcal{D} = \{D_t\}$ for $t \in T_D$. Each snapshot records the state of every node at a single time point, partitioning them into $V = S_t \cup E_t \cup I_t \cup R_t$.

Our goal is to infer the past states (*susceptible*, *exposed*, *infected* and *recovered*) of every node at every time $t \notin T_D$.

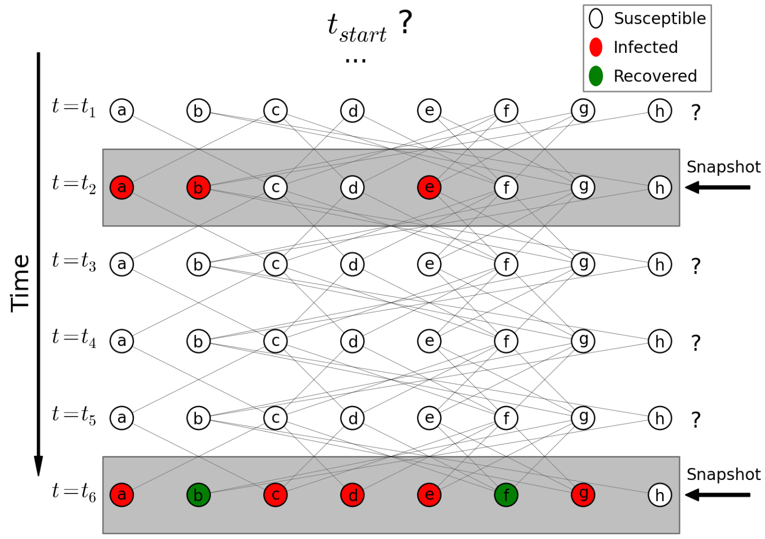


Fig. 2. Example Problem: SIR Diffusion over 8 node graph where we can only observe t_2 and t_6 without knowing the initial diffusion time t_{start}. We want to reconstruct the missing diffusion snapshots from t_{start} onwards

Figure 2 illustrates the history reconstruction problem for the SIR model. Each subsequent layer shows the progression of time, and we want to reconstruct the diffusion progression from unknown initial time t_{start} onwards given full state knowledge at subset of time points. The *initial spreader identification problem* is its special case where we want to identify only the initial *infected* nodes.

4. Non-monotone Submodular History Reconstruction (*DHR-sub*)

Let $t_{max} = \max(T_D)$, $t_{min} = \min(T_D)$, t_{start} be the unknown initial diffusion time, and $\mathcal{M}$ be the specific SEIRS type model that diffusion snapshots are collected over, SEIRS-type models are Markovian so the probability of diffusion $D = \{D_{t_{start}}, \dots, D_{t_{max}}\}$ that starts at t_{start} and progresses until t_{max} can be written as the multiplication of the probability of each time step in terms of previous time steps as in (1):

$$P(D) = \prod_{j=t_{start}+1}^{t_{max}} P(D_j | D_{j-1}, \dots, D_{t_{start}})\ P(D_{t_{start}}) \tag{1}$$

We assume the state transition probabilities (p_{uv}, $e2i_v$, $i2r_v$) to be same at each time step, so the overall diffusion probability in (1) simplifies to (2) under this memoryless property:

$$P(D) = \prod_{j=t_{start}+1}^{t_{max}} P(D_j | D_{j-1})\ P(D_{t_{start}}) \tag{2}$$

Let T_D be the ordered set of observed time points, $X_j = (S_j,\, E_j,\, I_j,\, R_j)$ be the unknown state knowledge at time $\forall j \notin T_D$, and $X = \{X_j : j \in \{t_{start}, \ldots, t_{max}\} \setminus T_D\}$. Given a collection of diffusion snapshots $\mathcal{D}$ for T_D, our goal is to reconstruct the most probable diffusion progression (X) by maximizing the log-likelihood as in Equation (3)–(5):

$$\begin{aligned}
&\operatorname*{argmax}_{X}\ \log(\mathcal{L}(X|\mathcal{D})) = \underbrace{\log(\mathcal{L}^{pre})}_{\textbf{\textit{DHR-sub-early}}} + \sum_{(j,k)\in\mathcal{P}(T_D)} \underbrace{\log(\mathcal{L}^{in}_{j,k})}_{\textbf{\textit{DHR-sub-between}}} && (3)\\
&\text{s.t. } \textbf{IntraConsistent}(X_j, \mathcal{M}),\ j \in t_{start}, \ldots, t_{max} - 1 && (4)\\
&\qquad \textbf{InterConsistent}(X_j, X_{j+1}, \mathcal{M}),\ j \in t_{start}, \ldots, t_{max} - 1 && (5)
\end{aligned}$$

where $\mathcal{L}^{pre}$ and $\mathcal{L}^{in}_{j,k}$ are defined in (6)–(7), $\mathcal{P}(T_D) = \{(t_j, t_{j+1}),\ j \in 1, \ldots, |T_D| - 1\}$, and maximum log-likelihood estimate is the same as the maximum likelihood estimate since the logarithm is a monotonically increasing function:

$$\begin{aligned}
\mathcal{L}^{in}_{j,k} &= P(X_{j+1}|D_j)\, P(D_k|X_{k-1}) \prod_{t\in j+1,..,k-2} P(X_{t+1}|X_t) && (6)\\
\mathcal{L}^{pre} &= \prod_{j=1}^{t_{min}-t_{start}} P(X_{t_{min}-j+1}|X_{t_{min}-j})\, P(X_{t_{start}}) && (7)
\end{aligned}$$

There are two types of constraints: **IntraConsistency** constraints (4) make sure that the variable assignments at each time step are valid under $\mathcal{M}$: every node belongs to a single state at each j, and **InterConsistency** constraints (5) make sure that the diffusion between each consecutive time steps is valid according to rules of $\mathcal{M}$: every node that got the infection at j has at least one infected predecessors at $j-1$, and node transitions are valid according to $\mathcal{M}$. For instance, recovered nodes cannot become susceptible if $\mathcal{M}$ is not loopy. These constraints are described in more detail below.

The diffusion history between consecutive observed T_D pairs is independent of each other since Eq. 2 is memoryless, and each D_j completely describes states of all nodes at time j. Thus, maximizing (3)–(5) can be partitioned into multiple independent subproblems of two types that can be optimized independently. The first type (***DHR-sub-early***) maximizes $\log(\mathcal{L}^{pre})$ under the consistency constraints to reconstruct the history before t_{min}. The second type (***DHR-sub-between***) maximizes $\log(\mathcal{L}^{in}_{j,k})$ to reconstruct the history between the snapshots from time j and time k under the consistency constraints. We define algorithms for both types of subproblems below. In the text, we use D_j and X_j interchangeably for $\forall j \in T_D$.

4.1. History Reconstruction Before the Earliest Observed Snapshot (*DHR-sub-early*)

To find the most likely diffusion history before t_{min}, we solve the problem:

$$\underset{X}{\operatorname{argmax}}\ \log(\mathcal{L}^{pre}) = \sum_{j=t_{start}}^{t_{min}-1} \log\Big(P(X_{j+1}|X_j)\Big) + \log\Big(P(X_{t_{start}})\Big) \tag{8}$$

$$\text{s.t. } \mathbf{IntraConsistent}(X_j, \mathcal{M}),\ j \in t_{start}, \ldots, t_{min}-1 \tag{9}$$

$$\mathbf{InterConsistent}(X_j, X_{j+1}, \mathcal{M}),\ j \in t_{start}, \ldots, t_{min}-1 \tag{10}$$

We assume a uniform prior $P(X_{t_{start}})$ over set of initially infected nodes since we do not have any extra information about them. We now discuss how to formulate the objective function and constraints above in terms of binary variables representing each node's state.

4.1.1. Expressing the objective function (8)

Given X_j, the probability of observing the diffusion snapshot X_{j-1} at time $j-1$ can be expressed as:

$$P(X_j|X_{j-1}) = \mathcal{L}(X_{j-1}|X_j) = \mathcal{L}_s^j\, \mathcal{L}_e^j\, \mathcal{L}_i^j\, \mathcal{L}_r^j \tag{11}$$

where $\mathcal{L}_s^j$, $\mathcal{L}_e^j$, $\mathcal{L}_i^j$, $\mathcal{L}_r^j$ are the likelihoods of the nodes in S_j, E_j, I_j, R_j states respectively in terms of X_{j-1}.

To define $\mathcal{L}_s^j$, $\mathcal{L}_e^j$, $\mathcal{L}_i^j$, $\mathcal{L}_r^j$, we introduce a single binary variable for each node to define its state at time $j-1$ given its state at time j. A binary variable is sufficient because there are only two possibilities for a node at time $j-1$ given its state at time j: either the node is in the same state as time j, or the node has made a state transition at time j, and when computing $\mathcal{L}(X_{j-1}|X_j)$, the state at time j is known.

We define a variable $s_{v,j-1}$ for every node $v \in E_j$, and $r_{v,j-1}$ for every node $v \in R_j$. For every node $v \in I_j$, we define variable for the incoming state of I ($e_{v,j-1}$ for bipartite $\mathcal{M}$ and $s_{v,j-1}$ for non-bipartite $\mathcal{M}$). Similarly, for every node $v \in S_j$, we define a variable for the incoming state of S if $\mathcal{M}$ is loopy, otherwise we do not need to define the variable, since we know that if a node is in S_j it must be in S_{j-1}.

With these variables, the likelihoods in (11) are explicitly defined as in (12)–(15) for SEIRS model as:

$$\mathcal{L}_e^j = \prod_{v \in E_j} \left(\mathcal{L}_{e2e}^{v,j-1} \left(1 - \mathcal{L}_{s2e}^{v,j,I}\ \mathcal{L}_{s2e}^{v,j,R}\right)^{s_{v,j-1}}\right) \tag{12}$$

$$\mathcal{L}_s^j = \prod_{v \in S_j} \left(\left(\mathcal{L}_{s2e}^{v,j,I}\ \mathcal{L}_{s2e}^{v,j,R}\right)^{s_{v,j-1}} (r2s_v)^{1-s_{v,j-1}}\right) \tag{13}$$

$$\mathcal{L}_r^j = \prod_{v \in R_j} \left((i2r_v)^{1-r_{v,j-1}} (1 - r2s_v)^{r_{v,j-1}}\right) \tag{14}$$

$$\mathcal{L}_i^j = \prod_{v \in I_j} \left(((e2i_v)^{e_{v,j-1}} (1 - i2r_v)^{1-e_{v,j-1}})\right) \tag{15}$$

where the sub-terms are defined as:

$$\mathcal{L}_{s2e}^{v,j,I} = \prod_{u \in P(v) \cap I_j} (1 - p_{uv})^{1-e_{u,j-1}}$$
$$\mathcal{L}_{s2e}^{v,j,R} = \prod_{u \in P(v) \cap R_j} (1 - p_{uv})^{1-r_{u,j-1}}$$
$$\mathcal{L}_{e2e}^{v,j} = \prod_{v \in E_j} (1 - e2i_v)^{1-s_{v,j-1}}$$

Each likelihood above for a given state ($\mathcal{L}_s^j$, $\mathcal{L}_e^j$, $\mathcal{L}_i^j$, $\mathcal{L}_r^j$) has two parts: the likelihood of the nodes staying at the given state, and the likelihood of the nodes transitioning towards the given state. For example, $\mathcal{L}_e^j$ is the likelihood for nodes $v \in E_{j-1}$ not to transition to infected state at time j, and nodes $v \in S_{j-1}$ to become exposed at time j. This gives us an explicit definition of objective function (8) in terms of a collection of binary variables.

Likelihoods (12)–(15) are defined for the most general model SEIRS, some of the likelihood terms disappear, or change slightly for models that are missing some of the states. For instance, parts including $r2s_v$ in $\mathcal{L}_s^j$ and $\mathcal{L}_r^j$ disappear for non-loopy $\mathcal{M}$, the likelihood representing the exogenous transition $\mathcal{L}_{s2e}$ is replaced by the similarly defined $\mathcal{L}_{s2i}$ for non-bipartite models, etc.

4.1.2. Expressing the constraints in equations (9) *and* (10)

The intra-consistency constraints (9) that require every node have a single state at each time step are already implied by the objective function since there is only a single variable for every node modeling the two possibilities. The inter-consistency constraints (10) can be modeled as packing constraints:

$$\sum_{u \in P(v) \cap I_j} e_{u,j-1} + \sum_{u \in P(v) \cap R_j} r_{u,j-1} + s_{v,j-1} \leq d_v, \forall v \in E_j \tag{16}$$

These constraints make sure every node that became exposed at time t ($v \in E_j, S_{j-1}$) has at least one incoming edge from node $u \in I_{j-1}$ ($u \in I_j \cup R_j$). However, these constraints (16) are already represented in the objective function (11) since the higher order term $\log(\mathcal{L}_e^j)$ takes the lowest possible value $\log(0) = -\infty$ when any of them are not satisfied.

4.1.3. Optimizing the likelihood under these constraints

Since t_{start} is unknown, we reconstruct the history using the above likelihoods and constraints by iteratively maximizing the likelihood at each time step $t_{start} \leq j < t_{min}$ backwards starting from $t_{min} - 1$, where the state is known. In each iteration, given X_j, we reconstructs the states at the previous time step $j - 1$ (X_{j-1}) by maximizing:

$$\max \text{ F} = \log(\mathcal{L}_s^j) + \log(\mathcal{L}_e^j) + \log(\mathcal{L}_i^j) + \log(\mathcal{L}_r^j) \tag{17}$$

Objective F for single step reconstruction is submodular as proven in Theorem 4.1 for all SEIRS-type models except SIS. For SIS model, history reconstruction can still be expressed as submodular maximization under packing and partition matroid constraints by modifying F as in Theorem 4.2.

Theorem 4.1. *F in Equation (17) is non-monotone submodular for all SEIRS-type models except SIS.*

Proof. F has three types of terms; higher order terms from $\log(\mathcal{L}_e^j)$, quadratic or linear terms from $\log(\mathcal{L}_s^j)$ depending on $\mathcal{M}$ and linear terms from $\log(\mathcal{L}_i^j)$ and $\log(\mathcal{L}_r^j)$. F is non-monotone since linear and quadratic terms are either positive or negative depending on $\mathcal{M}$, transition distribution parameters and the terms from $\log(\mathcal{L}_s^j)$ that model the probability of susceptible nodes not being infected/exposed.

F is submodular when $F(A+x)-F(A) \geq F(B+x)-F(B)$ for every $A \subset B$ and for every $x \in U \setminus (A \cup B)$. To prove submodularity of F, we prove the submodularity of each term in F since summation of submodular functions is also submodular. Linear terms of F are unimodular, so they are submodular. Quadratic terms show up in $\log(\mathcal{L}_s^j)$ when $\mathcal{M}$ is loopy and when the model is not SIS, each quadratic term is one of the following: $Q(r_{v,j-1}, r_{u,j-1}) = \log(1-p_{uv})(1-r_{v,j-1})(1-r_{u,j-1})$, $Q(r_{v,j-1}, e_{u,j-1}) = \log(1-p_{uv})(1-r_{v,j-1})(1-e_{u,j-1})$ or $Q(r_{v,j-1}, i_{u,j-1}) = \log(1-p_{uv})(1-r_{v,j-1})(1-i_{u,j-1})$. All those terms are submodular since they satisfy the inequality $Q(0,0)+Q(1,1) \leq Q(0,1)+Q(1,0)$.

Then, we need to prove the submodularity of the higher-order terms that depend on G to prove submodularity of F. Higher-order terms appear in either $\log(\mathcal{L}_e^j)$ for bipartite models or $\log(\mathcal{L}_i^j)$ for non-bipartite diffusion models. Depending on $\mathcal{M}$, we need to prove either $T = s_{v,j-1} \log\left(1.0 - \mathcal{L}_{s2e}^{v,j,I}\, \mathcal{L}_{s2e}^{v,j,R}\right)$ or $T = s_{v,j-1} \log\left(1.0 - \mathcal{L}_{s2i}^{v,j,I}\, \mathcal{L}_{s2i}^{v,j,R}\right)$. Each variable might appear at two positions of T; either inside or outside the logarithm. When $\mathcal{M}$ is bipartite, each variable can only appear in one of those positions whereas it can appear in both positions for non-bipartite $\mathcal{M}$. Let $V_e = \bigcup_{u \in P(v) \cap I_j} e_{u,j-1}$, $V_r = \bigcup_{u \in P(v) \cap E_j} r_{u,j-1}$, x be the variable to be added, X be the current set of added variables, $K = \Pi_{V_e \cup V_r}(1-p_{uv})$ and $P_t = (1-p_{tv})^t$ for every $t \in V_e \cup V_r$, T is submodular as proven below.

– If x is outside the logarithm, let $A = \{a,b\}$ and $B = \{a,b,c\}$. Then, $T(A+x) = \log\left(1 - \frac{K}{P_a P_b P_x}\right)$, $T(B+x) = \log\left(1 - \frac{K}{P_a P_b P_c P_x}\right)$ and $T(A+x) - T(A) \geq T(B+x) - T(B)$ will be satisfied since $T(A+x) \geq T(B+x)$ and $T(A) = T(B) = 0$.

– If x is inside the logarithm, when $s_{v,j-1} \notin X$, submodularity is trivially satisfied since $T(A) = T(A+x) = T(B) = T(B+x) = 0$. When $s_{v,j-1} \in X$, let $A = \{a\}$ and $B = \{a,c\}$ ($A \subset B$), submodularity is satisfied as shown in Equation (18)–(20).

$$T(A+x) - T(A) \geq T(B+x) - T(B) \tag{18}$$

$$\log\left(\frac{1 - \frac{K}{P_a P_x}}{1 - \frac{K}{P_a}}\right) \geq \log\left(\frac{1 - \frac{K}{P_a P_b P_x}}{1 - \frac{K}{P_a P_c}}\right) \tag{19}$$

$$K P_a P_b (1-P_b)(1-P_a) \geq 0 \tag{20}$$

Then, F is submodular since each summation term including the higher-order ones is submodular.

□

Theorem 4.2. Program (17)–(16) for SIS can be expressed as submodular maximization under both packing and partition matroid constraints.

Proof. Quadratic terms $Q(s_{v,j-1}, s_{u,j-1})$ from $\mathcal{L}_s^j$ are supermodular for SIS but they can be turned into submodular ones as follows: We define new varible $i_{v,j-1}$ for every node $v \in \{S_j \cup I_j\}$ to represent whether v is infected at time $j-1$. Then, we obtain the new objective function F^* by replacing each supermodular $Q(s_{v,j-1}, s_{u,j-1}) = log(1-p_{uv})s_{v,j-1}(1-s_{u,j-1})$ with $Q^*(s_{v,j-1}, s_{u,j-1}) = log(1-p_{uv})s_{v,j-1}i_{v,j-1}$. We also add assignment constraints of $s_{v,j-1}+i_{v,j-1} = 1$ for every node $v \in \{S_j \cup I_j\}$ to make sure node v is either infected or susceptible at $j-1$. Each $Q^*(s_{v,j-1}, s_{u,j-1})$ in F^* is submodular since it satisfies the inequality $Q^*(0,0)+Q^*(1,1) \leq Q^*(0,1)+Q^*(1,0)$. Then, F^* is submodular since the rest of the higher-order terms are submodular as proven in Theorem 4.1. Assignment constraints define partition matroid and the problem of reconstructing history at time $j-1$ becomes submodular maximization under both partition matroid and existing packing constraints for SIS model.

□

Therefore, optimizing (17) is a non-monotone submodular maximization problem. Non-monotone submodular maximization is NP-hard since its special cases such as *MAX DICUT* is NP-hard (Feige and Goemans, 1995). To solve this problem, we apply the deterministic non-monotone submodular maximization method by (Feige, Mirrokni and Vondrak, 2007) repeatedly between adjacent time steps and iterate until the estimates between the consecutive time steps are same, indicating that we have reached the initial t_{start} state. At each step, (Feige et al., 2007) maximizes a normalized F_n at every step that is obtained by adding $-F(\emptyset)$ to every $S \subset 2^N$ so $F_n(\emptyset) = 0$, where $F(\emptyset)$ is the value of the objective if no nodes change states between adjacent time points. As applied here, this is done by starting with an initial solution $X'_j = \emptyset$ that represents the same state assignments between the consecutive time steps j and $j-1$. For each time step, we add the node with the most increase in F_n to the set of nodes that have changed state. Algorithm 1 gives a schematic outline of the procedure.

Algorithm 1 *DHR-sub-early*

```
$j \leftarrow t_{min} - 1$
**repeat**
  $\{X'_j$ is the set of nodes that changed state at time $j\}$
  $X'_j \leftarrow \emptyset$
  **repeat**
    Add nodes to $X'_j$ according to the rule for non-monotone submodular
    maximization approximation (Feige et al., 2007)
  **until** no node can be added that increases the score
  $j \leftarrow j - 1$
**until** $X'_j = X'_{j+1}$
```

The method by (Feige et al., 2007) has a $\frac{1}{3}$ approximation ratio for normalized submodular functions, and we found it to perform in practice better than the randomized algorithm (Buchbinder, Feldman, Naor and Schwartz, 2012) with approximation ratio 0.5 due to the structure of our problem. The $\frac{1}{3}$ ratio for normalized F_n implies a data-dependent bound for F as proven in Theorem 4.3 where $F(\emptyset) = -S_0$, X_{opt} is the set of elements maximizing F and $F(X_{opt}) = -O$.

Theorem 4.3. Algorithm 1 has an approximation guarantee of $k + \frac{S_0}{O}(1-k)$ for $k = \frac{1}{3}$ in terms of minimization of supermodular $-F$ for each of its iteration.

Proof. Let X be the set of elements returned by the non-monotone submodular maximization algorithm and $F(X) = -M$. We are interested in upper-bounding the supermodular minimization ratio $(\frac{M}{O})$ for $-F$. Since F_n is obtained by adding S_0 to each set in F, $\frac{F_n(X)}{F_n(X_{opt})} = \frac{S_0 - M}{S_0 - O} \geq k$ and we obtain $\frac{M}{O} <= k + \frac{S_0}{O}(1-k)$. Here, $\frac{S_0}{O}(1-k)$ makes the approximation ratio data-dependent and this ratio is the best we can achieve when k is tight for non-monotone submodular maximization. This data-dependent bound is also the best we can achieve in terms of supermodular minimization perspective since non-negative supermodular minimization problem cannot be approximated in constant factor unless $P = NP$ (Wolsey and Nemhauser, 1999).
□

4.2. History Reconstruction Between Consecutive Snapshots (*DHR-sub-between*)

History reconstruction for every interval between by consecutive, observed T_D pairs is independent of other intervals. Therefore, we can solve each independently by solving the following problem:

$$\underset{X}{\text{argmax}}\ T = \log(P(X_{j+1}|D_j)) + \log(P(D_k|X_{k-1})) + \sum_{t=j+1}^{k-2} \log(P(X_{t+1}|X_t)) \tag{21}$$

$$\text{s.t}\ \ \textbf{IntraConsistent}(X_t, \mathcal{M}),\ t \in j+1, \ldots, k-1 \tag{22}$$

$$\textbf{InterConsistent}(X_t, X_{t+1}, \mathcal{M}),\ t \in j, \ldots, k-1 \tag{23}$$

where we only consider X_t that lay within the $[j,k]$ interval bracketed by diffusion snapshot observations D_j and D_k.

4.2.1. Expressing objective (21)

Objective (21) has three parts: $\log(P(D_k|X_{k-1}))$ is same as the single step backwards reconstruction of *DHR-sub-early*, and $\log(P(X_{j+1}|D_j))$ is a trivial forward diffusion expression with unknown X_{j+1} and known D_j. Lastly, both X_{t+1} and X_t are unknown in $\log(P(X_{t+1}|X_t))$, and it can be written explicitly as:

$$\log(P(X_{t+1}|X_t)) = \log(\mathcal{L}_s^{t+1}) + \log(\mathcal{L}_e^{t+1}) + \log(\mathcal{L}_i^{t+1}) + \log(\mathcal{L}_r^{t+1}) \tag{24}$$

where the likelihoods are defined as:

$$\mathcal{L}_e^{t+1} = \prod_{v \in V} ((1 - e2i_v)^{e_{v,t}\, e_{v,t+1}}\ (1.0 - \mathcal{L}_{exo})^{e_{v,t+1}}) \tag{25}$$

$$\mathcal{L}_s^{t+1} = \prod_{v \in V} (\mathcal{L}_{exo}^{s_{v,t+1}} (r2s_v)^{r_{v,t}\, s_{v,t+1}}) \tag{26}$$

$$\mathcal{L}_r^{t+1} = \prod_{v \in V} \left((i2r_v)^{i_{v,t}\, r_{v,t+1}} (1.0 - r2s_v)^{r_{v,t}\, r_{v,t+1}}\right) \tag{27}$$

$$\mathcal{L}_i^{t+1} = \prod_{v \in V} \left((e2i_v)^{e_{v,t}\, i_{v,t+1}} (1.0 - i2r_v)^{i_{v,t}\, i_{v,t+1}}\right) \tag{28}$$

and the term in (25) is:

$$\mathcal{L}_{exo} = \prod_{u \in P(v)} (1 - p_{uv})^{i_{u,t}\, s_{v,t}}$$

Objective (21) is non-monotone submodular as proven in Theorem 4.4.

Theorem 4.4. T in (21) is non-monotone submodular for all SEIRS-type models.

Proof. We prove the submodularity of $\log(\mathcal{L}_{j,k}^{in})$ by proving the submodularity of each of its summation terms. $\log(P(X_{j+1}|D_j))$ estimates the most probable diffusion snapshot at $j+1$ given D_j. It is a forward estimate and if we use the same variable naming as in Section 4.1, it becomes a linear function of X_{j+1} and thus submodular.

$\log(P(D_k|X_{k-1}))$ is same as F (17) in Section 4.1 and it is submodular as proven in Theorem 4.1.

Every $\log(P(X_{t+1}|X_t))$ involves the variables from both time steps t and $t+1$. Here, we do not know the exact node states at both time steps so we define all possible state variables for every node for both time steps ($s_{v,t}$, $e_{v,t}$, $i_{v,t}$, $r_{v,t}$, $s_{v,t+1}$, $e_{v,t+1}$, $i_{v,t+1}$, $r_{v,t+1}$, $\forall v \in V$). $\log(P(X_{t+1}|X_t))$ can be expressed as in Equation 29 where the likelihoods are defined as in Equation (30)–(33). Each term in $\log(P(X_{t+1}|X_t))$ is additive and log-likelihood terms of endogenous transitions are submodular since they are quadratic terms with negative coefficient. Log-Likelihood terms of exogenous transitions are also submodular by following the submodularity proof of the higher-order terms from Theorem 4.1.

$$\log(P(X_{t+1}|X_t)) = \log(\mathcal{L}_s^{t+1}) \ + \log(\mathcal{L}_e^{t+1}) \ + \log(\mathcal{L}_i^{t+1}) \ + \log(\mathcal{L}_r^{t+1}) \tag{29}$$

$$\mathcal{L}_{exo} = \prod_{u \in P(v)} (1 - p_{uv})^{i_{u,t} s_{v,t}}$$

$$\mathcal{L}_e^{t+1} = \prod_{v \in V} ((1 - e2i_v)^{e_{v,t} e_{v,t+1}} \ (1.0 - \mathcal{L}_{exo})^{e_{v,t+1}}) \tag{30}$$

$$\mathcal{L}_s^{t+1} = \prod_{v \in V} (\mathcal{L}_{exo}^{s_{v,t+1}} (r2s_v)^{r_{v,t} s_{v,t+1}}) \tag{31}$$

$$\mathcal{L}_r^{t+1} = \prod_{v \in V} \left((i2r_v)^{i_{v,t} r_{v,t+1}} (1.0 - r2s_v)^{r_{v,t} r_{v,t+1}}\right) \tag{32}$$

$$\mathcal{L}_i^{t+1} = \prod_{v \in V} \left((e2i_v)^{e_{v,t} i_{v,t+1}} (1.0 - i2r_v)^{i_{v,t} i_{v,t+1}}\right) \tag{33}$$

□

4.2.2. Expressing the inter- and intra-consistency constraints

The inter-consistency constraints (23) ensure the validity of diffusion, and they are explicitly written, using binary state variables $s_{v,t}$, $e_{v,t}$, $i_{v,t}$, $r_{v,t}$ for every $v \in V$ and $t \in j, \ldots, k$, as:

$$s_{v,t} + i_{v,t+1} + r_{v,t+1} \leq 1, \tag{34}$$

$$e_{v,t} + s_{v,t+1} + r_{v,t+1} \leq 1, \tag{35}$$

$$i_{v,t} + s_{v,t+1} + e_{v,t+1} \leq 1, \tag{36}$$

$$r_{v,t} + e_{v,t+1} + i_{v,t+1} \leq 1,\ t \in j, \ldots, k-1,\ v \in V \tag{37}$$

$$e_{v,t+1} - s_{v,t} \leq \sum_{u \in P(v)} i_{u,t},\ t \in j, \ldots, k-1,\ v \in V \tag{38}$$

Constraints (34)–(37) ensure that state transitions obey SEIRS dynamics rules such as a node infected at t cannot be susceptible or exposed at $t+1$. The remaining constraint (38) ensures that a newly exposed node must have at least one infected predecessor at previous time step. The constraints (38) are already represented in the objective function, since $(1.0 - \mathcal{L}_{exo})^{e_{v,t+1}}$ in $\mathcal{L}_e^{t+1}$, and (24) takes the lowest possible value $\log(0) = -\infty$ when any of them is not satisfied. So, we can remove the constraints (38) without affecting the results. (Some of these constraints are modified accordingly for subset of models. For example, (36) becomes $i_{v,t} + s_{v,t+1} \leq 1$ for SI.)

The intra-consistency constraints (22) ensure that every node belongs to a single state at each time step:

$$s_{v,t} + e_{v,t} + i_{v,t} + r_{v,t} = 1,\ t \in j+1, \ldots, k-1,\ v \in V \tag{39}$$

4.2.3. Optimizing (21) *in practice*

Let $E = \{s_{v,t}, e_{v,t}, i_{v,t}, r_{v,t} \,|\, v \in V,\ t \in j, \ldots, k\}$. Then the intra-consistency constraints (39) define base of a partition matroid over the ground set E (Lee, Mirrokni, Nagarajan and Sviridenko, 2009), and constraints (34)–(37) define 2-independence system over the same ground set: Its rank quotient is 2 since the ratio of cardinality of the largest base (maximal independent set) to the cardinality of the smallest base is at most 2. For more detailed information on matroid and independence system, see (Schrijver, 2003; Gupta, Roth, Schoenebeck and Talwar, 2010).

Combining (24) and (21) with the discussion above, the history reconstruction between time steps j and k can be then written as optimizing

$$\max\ T = \sum_{t=j+1}^{k} \log(\mathcal{L}_s^t) + \log(\mathcal{L}_e^t) + \log(\mathcal{L}_i^t) + \log(\mathcal{L}_r^t) \tag{40}$$

subject to inter-consistency constraints (34)–(37) and intra-consistency constraints (39). (39) cannot be removed as in Section 4.1 since each node may belong to any state at time t as we do not know the node states at $t-1$ or $t+1$ (except for boundary times j and k).

When considered together, constraints (34)–(37) and (39) are base of a new matroid defined by the intersection of the partition matroid and 2-independence system. Proof is as follows: Intersection of constraints (34)–(37) and (39) relaxed to $\leq$ define a matroid $\mathcal{M}_p = (E_p, \mathcal{I}_p)$ where $E_p = E$, and independent set $\mathcal{I}_p$ is subset of E satisfying (34)–(37) and relaxed (39). $\mathcal{M}_p$ defines a matroid since all its bases (maximal independent set) have the same cardinality $(k-j-1)|V|$; we can always find a state assignment for every node and every time step that satisfies the constraints, and we cannot assign multiple states to each node at each time step. Then, equality constraints in the original equations (39) force independent sets in $\mathcal{I}_p$ to be bases of $\mathcal{M}_p$, as cardinality of an independent set in $\mathcal{I}_p$ will now always be $(k-j-1)|V|$.

This problem becomes non-monotone submodular maximization under matroid base constraints. It is NP-hard (Lee et al., 2009), and its normalized version can be approximated by $\frac{1}{6}$ by modified local search (Lee et al., 2009). We run this method by (Lee et al., 2009) in *DHR-sub-between* to reconstruct the history between j and k. *DHR-sub-between* has three main steps: In the first step, it starts with a base of M_p that also satisfies (38), and it finds a base $B_1 \subseteq \mathcal{M}_p$ that is optimal under swap operations. In the second step, it removes B_1 from M_p and greedily finds independent set X_2 that is locally optimal under addition and deletion operations. In the third step, it contracts independent set X_2 from M_p and finds two disjoint bases B_a, B_b that are guaranteed to exist when the original matroid M_p has two disjoint bases. Lastly, it returns the best of three bases B_1, $X_2 \cap \mathcal{B}_a$ or $X_2 \cap \mathcal{B}_b$. The resulting solution always satisfies (38) without explicitly checking for them: Local search will not replace the current solution with a low-score invalid solution as the objective (40) takes the lowest possible value $-\infty$ if any of (38) are not satisfied.

5. Prize Collecting (Dominating Set) Vertex Cover Relaxations (*DHR-pcdsvc*, *DHR-pcvc*)

Although accurate and practical for smaller graphs, *DHR-sub* may take some time to solve for larger graphs. We can reconstruct the history before the earliest observed snapshot faster by relaxing *DHR-sub-early*. For a relaxed version of the problem, we define variables differently than above. When reconstructing the history at previous time $j-1$, we define $i_{v,j-1}$, $\forall v \in I_j \cup R_j$, $s_{v,j-1}$, $\forall v \in S_j$, and $e_{v,j-1}$, $\forall v \in E_j$. After this transformation, $\mathcal{L}_e^j$ (12) turns into

$$\mathcal{L}_e^j = \prod_{v \in E_j} \left(\mathcal{L}_{e2e}^{v,j-1} \left(1 - \mathcal{L}_{s2e}^{v,j}\right)^{1-e_{v,j-1}} \right) \tag{41}$$

where $\mathcal{L}_{s2e}^{v,j}$ and $\mathcal{L}_{e2e}^{v,j}$ are defined as in (42)–(43):

$$\mathcal{L}_{s2e}^{v,j} = \prod_{u \in P(v) \cap (I_j \cup R_j)} (1 - p_{uv})^{i_{u,j-1}} \tag{42}$$

$$\mathcal{L}_{e2e}^{v,j} = \prod_{v \in E_j} (1 - e2i_v)^{e_{v,j-1}} \tag{43}$$

The other likelihoods are transformed similarly.

The hardness of *DHR-sub-early* comes from higher-order terms $\left(1 - \mathcal{L}_{s2e}^{v,j}\right)^{1-e_{v,j-1}}$ in (41), so we replace them with their first-order Taylor expansion $\mathcal{T}_{s2e}^{v,j}$ at the point ($i_{u,j-1} = 1, \forall u \in P(v) \cap \{I_j \cup R_j\} \cup e_{v,j-1} = 1$) as in (44).

$$\mathcal{T}_{s2e}^{v,j} = \log(K_{s2e}) + \frac{1}{K_{s2e}} \sum_{u \in I_j \cup R_j} \frac{\partial \mathcal{L}_{s2e}^{v,j,(I,R)}}{\partial i_{u,j-1}} (i_{u,j-1} - 1) \tag{44}$$

In (44), $K_{s2e} = \mathcal{L}_{s2e}^{v,j}(1, \ldots, 1) \approx 1$, so the original reconstruction Problem 17 for single time step turns into minimizing $-F_r$ as in (45)–(46):

$$\begin{aligned} \min \; - F_r = & \sum_{(u,v) \in E^*} w_{uv} \bar{i}_{u,j-1} \, \bar{e}_{v,j-1} + \sum_{u \in I_j \cup R_j} w_u \, i_{u,j-1} + \sum_{v \in E_j} w_v \, e_{v,j-1} + \\ & \sum_{v \in S_j} w_v \, s_{v,j-1} \end{aligned} \tag{45}$$

$$\text{s. t.} \sum_{u \in P(v) \cap \{I_j \cup R_j\}} i_{u,j-1} + e_{v,j-1} \geq 1, \, v \in E_j \tag{46}$$

where $\bar{i}_{u,j} = 1 - i_{u,j}$ and $\bar{e}_{v,j} = 1 - e_{v,j}$. The covering constraints (46) are inter-consistency constraints ensuring the validity of the diffusion. Similar to *DHR-sub-early*, we do not need intra-consistency constraints since we are reconstructing the history step by step. This problem is *Prize Collecting Dominating Set Vertex Cover* (PCDSVC) over the graph $G^* = (V^*, E^*)$ where $V^* = V$ with weights w_v and directed edge from u to v with weight $w_{uv} = -\log(1 - p_{uv})$ exists when $v \in E_j, u \in P(v) \cap \{I_j \cup R_j\}$ for bipartite $\mathcal{M}$ and $v \in I_j, u \in P(v) \cap \{I_j \cup R_j\}$ for non-bipartite $\mathcal{M}$.

PCDSVC is different than *Vertex Cover* because (1) We may not cover an edge (u, v) if we pay its price w_{uv}, and (2) A feasible solution is a vertex dominating set. This problem has not been studied before, it is NP-hard, and it can be approximated by $O(\log(|V^*|))$ by formulating it as *Minimum Hitting Set* and running the greedy method for *Set Cover* as proven in Theorem 5.1.

Theorem 5.1. Prize Collecting Dominating Set Vertex Cover (PCDSVC) is NP-hard, and it can be approximated by $O(\log(|V^*|))$.

Proof. PCDSVC is NP-hard since its special case *Dominating Set* is NP-hard that is obtained when all edge weights are 0 ($w_{uv} = 0$).

Given PCDSVC problem over graph $G^* = (V^*, E^*)$, we construct *Minimum Hitting Set* instance (S, C) as follows: We define the set of elements as $S = \{v \in V^*\} \cup \{e \in E^*\}$ where the cost of each each item in E^* is w_u for every $u \in V^*$ and w_{uv} for every $(u, v) \in E^*$. Subsets $C = C_1 \cup C_2$ of S are defined as: $C_1 = \{e_u, e_v, e_{uv}\}$ $, \forall (u, v) \in E^*$ and $C_2 = \{e_u, u \in P(v) \cup \{v\}\}$ $, \forall v \in V^*$. This reduction is linear time, approximation preserving and the solution of this *Minimum Hitting Set* (MHS) gives us the solution for PCDSVC. Here $|S| = |E^*| + |V^*|$ and **Greedy** method for *Set Cover* approximates this problem by $\log(|S|) + 1 \approx O(\log(|E^*| + |V^*|)) + 1 \approx O(\log(|V^*|)) + 1$.

One can also easily show that each MHS instance can be reduced to PCDSVC and this reduction is approximation preserving. Then,MHS and PCDSVC are *equivalent* under linear reduction and this approximation ratio for PCDSVC is the best we can achieve unless P=NP (Feige, 1998).

□

We can relax this problem further by removing (46) and it becomes *Prize Collecting Vertex Cover* (PCVC). PCVC can be approximated by a factor of 2 using the LP relaxation (Hochbaum, 2000), and it can be solved optimally for *bipartite diffusion models* by expressing it as s-t mincut as proven in Theorem 5.2.

Theorem 5.2. The Taylor expansion relaxation of (17) for *bipartite diffusion models* can be expressed as *s-t mincut.*

Proof. Minimization problem for bipartite $\mathcal{M}$ has objective F_{bi} as seen in Equation 47. F_{bi} is a regular function (Kolmogorov and Zabih, 2004): when expressed as the summation of first and second-order terms as in Equation 48, each second order term $E^{u,v}(s_{v,j-1}, i_{u,j-1})$ satisfies $E^{u,v}(0,0) + E^{u,v}(1,1) \leq E^{u,v}(0,1) + E^{u,v}(1,0)$ in regular functions. Regular functions can be solved optimally by transforming it into s-t mincut as follows:

$$\begin{aligned}\min\ -F_{bi} = &\sum_{(u,v)\in E^*} \frac{1}{\log(1-p_{uv})}(1-i_{u,j-1})s_{v,j-1} + \sum_{v\in E_j \cup S_j} w_v s_{v,j-1} \\ &+ \sum_{v \in I_j \cup R_j} w_v i_{v,j-1}\end{aligned} \tag{47}$$

$$-F_{bi} = \sum_{\substack{u\in I_j\cup R_j,\\ v\in E_j\cup S_j}} E^{u,v}(i_{u,j-1}, s_{v,j-1}) + \sum_{v\in I_j\cup R_j} E^v(i_{v,j-1}) + \sum_{v\in S_j \cup E_j} E^v(s_{v,j-1}) \tag{48}$$

We define new directed graph $G' = (V', E')$ where $V' = V^* \cup \{s\} \cup \{t\}$. For every $v \in V^*$, we add edge (s, v) with weight $E^v(1)$ if $E^v(1) > 0$ and add edge (v, t) with weight $-E^v(1)$ if $-E^v(1) < 0$. For every $u \in I_j \cup R_j$ and $v \in S_j \cup E_j$, we add edge (u, v) with weight $E^{u,v}(0, 1)$. The s-t mincut solution of this graph gives us the resulting node partition: After the cut edges removed, variables of the nodes that are reachable from s are assigned 1 and the variables of the nodes that have a path to t are assigned 0.

□

The algorithms for these relaxed versions, *DHR-pcdsvc* and *DHR-pcvc*, are similar to *DHR-sub* except they run PCDSVC and PCVC respectively instead of submodular maximization at each iteration.

6. Ensemble Initial Spreader Identification

We define *DHR-sub*-ens, *DHR-pcdsvc*-ens and *DHR-pcvc*-ens for the ensemble versions of our methods: they estimate the most likely subset of nodes that explains the diffusion data from multiple runs. For each initial time point seen in the multiple runs, we greedily select the subset of nodes seen in that time point that best explains $\mathcal{D}$ in terms of minimum absolute difference F_{dif} defined as in:

$$F_{dif} = |S^e_t - S^t_t| + |E^e_t - E^t_t| + |I^e_t - I^t_t| + |R^e_t - R^t_t| \tag{49}$$

where $S^e_t, E^e_t, I^e_t, R^e_t$ are the set of estimated nodes whereas $S^t_t, E^t_t, I^t_t, R^t_t$ are the set of true nodes for S, E, I, R states at time t respectively. We keep adding the node that improves F_{dif} the most until there is no improvement. Lastly, we return set of nodes that has the minimum score among the all possible initial time points as our initial spreader prediction.

7. Experimental Results

7.1. Comparison and Evaluation

We compared our methods with *NetSleuth*, *Keffectors* and *Rumor* in identifying the initial spreaders. *Keffectors* and *Rumor* require estimates of the number of initial spreaders, so we provide them an estimate of the initial spreader count by the number of clusters in G estimated by modularity (Blondel, Guillaume, Lambiotte and Lefebvre, 2008). We compared our methods with the baseline heuristic *GreedyForward* for history reconstruction that reconstructs the history in each interval by simulating a forward trace starting from the interval's earlier time. We return the topmost k spreaders from *Rumor* sorted by its rumor centrality metric where k is the number of clusters in G.

We validated the history reconstruction performance by Kendall Tau-b statistic (Agresti, 2002) (τ_B) that measures the similarity between true and estimated node orderings defined in terms of infection times by also adjusting for ties:

$$\tau_B(T,O) = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{50}$$

Here, T and O are true and inferred node orderings respectively in terms of given state (such as *infected*). Let V_T be set of nodes seen in true ordering T, then τ_B, n_c are n_d are concordant and discordant pairs respectively, $n_0 = \frac{|V_T|(|V_T|-1)}{2}$, n_1 and n_2 are sum of tied quantities in the true and observed orderings respectively. τ_B adjusts for ties by subtracting n_1 and n_2 from n_0 in the denominator.

We validated the initial spreaders identification performance by graph-based average matching score ($\overline{M}_G$). Let $\hat{V}_t$ and $\hat{V}_o$ be true and estimated initial nodes respectively, and $G_b = (\hat{V}_t \cup \hat{V}_o, \hat{V}_t \times \hat{V}_o)$ be a weighted bipartite graph with weights $w_{ab} = \frac{1}{1+d_{ab}}$ for every $a \in \hat{V}_t$, $b \in \hat{V}_o$ where d_{ab} is the distance between a and b in G. $\overline{M}_G$ estimates the maximum bipartite matching score in G_b, and returns the average. When $|\hat{V}_o| \neq |\hat{V}_t|$, M_G is modified to account for the unmatched vertices by matching them independently to the best ones.

Both τ_B and $\overline{M}_G$ are normalized, and higher score means better performance in both. We implemented all our methods, synthetic trace generator and existing methods *NetSleuth*, *Rumor*, *Keffectors* in Python, solved LP relaxations by CPLEX (*IBM ILOG CPLEX Optimizer*, 2010), and modified and used C++ maximum flow code from (Boykov and Kolmogorov, 2004). We run all our experiments on Macbook Pro with 2.5 Ghz CPU and 8 GB memory. Our code and data are available on the web[1].

7.2. Reconstruction Performance on Synthetic Data

We generated 5 networks of 500 nodes and 5000 edges that are grown by *Erdös-Reyni* (Erdos and Rnyi, 1960), *Forest Fire* (Leskovec, Kleinberg and Faloutsos, 2005), *linear preferential attachment* (Barabási and Albert, 1999) network growth models. We generated each synthetic trace by choosing the given number of source nodes randomly, making them infected, and running the diffusion

[1] `http://www.cs.cmu.edu/~ckingsf/software/dhrec`

	Intial Spreader			History			
	FF			*LPA*		*RDS*	
	SI	SIR	SEIR	SI	SIR	SI	SIR
DHR-sub	0.8	0.83	0.81	0.97	0.88	0.69	0.77
DHR-sub-ens	0.87	0.88	0.89	-	-	-	-
DHR-pcdsvc	0.78	0.8	0.81	0.9	0.82	0.64	0.73
DHR-pcvc	0.76	0.76	0.79	0.88	0.77	0.59	0.72
Rumor	0.74	0.7	0.6	-	-	-	-
NetSleuth	0.75	0.8	0.64	-	-	-	-
Keffectors	0.77	0.74	0.7	-	-	-	-
GreedyForward	-	-	-	0.34	0.28	0.31	0.23

Table 2. $\overline{M}_G$, τ_B vs. growth and diffusion models for spreader identification (5 true spreaders) and history reconstruction from $|T_D| = 2$ snapshots

over the network until either all nodes become recovered (or infected under the SI model) or until the spread dies out. When multiple snapshots are given, we sample them uniformly in the range (t_{min}, t_{max}).

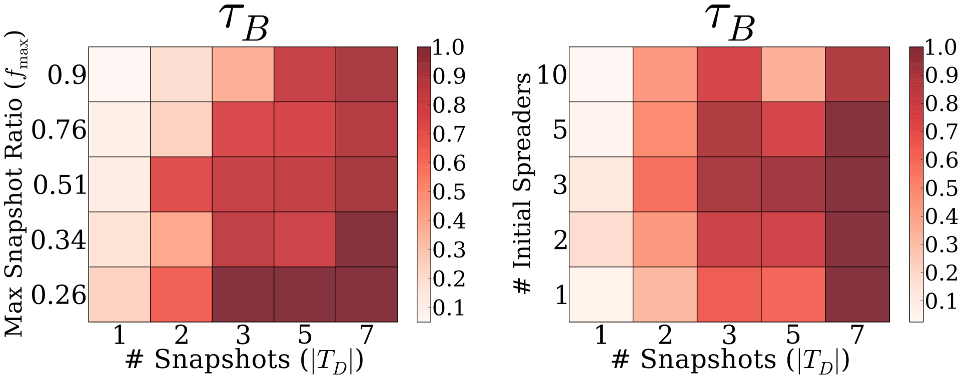
We test our methods on SI, SIR, SEIR by modeling the transition distributions from a geometric distribution with different parameters in each model in order to assess performance under various conditions. In SI, we selected p_{uv} for every $(u, v) \in E$ uniformly between 0.1 and 0.4. In SIR, we selected $p_{uv}, \forall(u, v) \in E$ uniformly in the range $(0.2, 0.6)$ and $i2r_v, \forall v \in V$ uniformly in the range $(0.5, 0.6)$. In SEIR, we selected $p_{uv}, \forall(u, v) \in E$, $e2i_v, \forall v \in V$, $i2r_v, \forall v \in V$ each uniformly in the range $(0.4, 0.8)$.

DHR-sub and its ensemble version *DHR-sub*-ens perform the best on all the models in terms of identifying the initial spreaders as in Table 2. Its relaxations *DHR-pcdsvc* and *DHR-pcvc* also perform better than the existing methods, and they are good alternatives to *DHR-sub* considering their faster running times. The performance difference between our methods and the existing methods become more apparent especially for SIR and SEIR models.

In terms of history reconstruction, all our methods perform much better than the greedy baseline *GreedyForward*. All our methods perform better when multiple snapshots are available as seen in Figure 3 for *DHR-sub* for both SI and SIR. *DHR-sub* reconstructs the histories more precisely when the interval to be reconstructed has lower maximum snapshot ratio ($f_{max} = t_{max}/l_D$) where l_D is the diffusion length, and its performance is not significantly affected by the number of initial spreaders given the same number of snapshots as in Figure 3. Lower reconstruction performance for higher f_{max} intervals is due to increasing number of similar quality diffusion histories. In its extreme, τ_B may become close to 0 when reconstructing histories of longer intervals from a single snapshot.

7.3. Reconstructing Meme Diffusion History from Blog Data

We used our methods to extract the diffusion history of memes that are defined as short textual phrases that travel through the Web. We inferred the diffusion progression of several memes in two blog networks under SI using the true



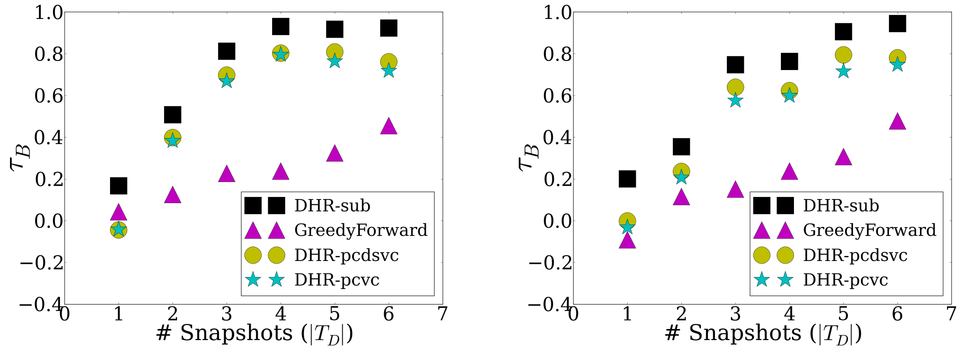
a) SI b) SIR

Fig. 3. τ_B vs. number of snapshots (x axis) and max snapshot ratio (y-axis), number of true initial spreaders (y-axis) for history reconstruction over *Forest Fire* for *DHR-sub* a) SI b) SIR

diffusion data from (Gomez-Rodriguez et al., 2013): *Top-Blog* has 5000 nodes and 30072 edges and it shows the connection between the topmost 5000 blogs, *Rand-Blog* has 250 nodes and 3342 edges, and it shows the connection between random 250 blogs. In both networks, nodes represent either personal blogs or mass media, and edges represent hyperlinks from one blog to the another one. We do not know the true p_{uv}, so we estimate them by a geometric distribution with p being 0.3 between mass media, 0.25 from mass media to bloggers, 0.15 between bloggers, and 0.05 from bloggers to media. Traces for several topics were obtained from the same source (Gomez-Rodriguez et al., 2013). When tracking the diffusion of a topic, if a blog publishes about it at multiple time points, we assume blog is infected at the earliest time point.

We reconstructed the diffusion history of the memes *Fukushima*, *Arab Spring* and *Nba* on both *Top-Blog* and *Rand-blog* as in Figures (4)–(5). Values inside the parentheses define the time scale for the meme progression ($1 : 5 = 1$ time unit for 5 days). τ_B are lower than the synthetic case especially when fewer than 2 snapshots are available but they are still reasonable since the true diffusion parameters are unknown. In Figure 4, *DHR-sub* performs the best, and all methods reconstruct the diffusion history better when more diffusion data is available. When run with multiple snapshots, *DHR-sub* better captures the diffusion direction and performs almost close to 1 whereas heuristic method *Greedy-Forward*'s τ_B never exceeeds 0.5. Although *Fukushima* and *Arab Spring* have different diffusion dynamics (Gomez-Rodriguez et al., 2013), both trajectories can be reconstructed precisely by *DHR-sub*. Similar to the synthetic case, performance of *DHR-sub* increases if more snapshots are available, and it decreases as f_{max} increases as in Figure 5. Overall, both *DHR-sub* and *DHR-pcdsvc* can nicely fill in the missing gaps of the meme diffusion history.

In another example, the order of diffusion estimated by *DHR-sub* matches the true order of the meme *Occupy* reasonably well ($\tau_B = 0.77$). In this case, most of the initial diffusion of *Occupy* happens between mass media, and diffusion at personal blogs start to show up later. However, the speed of the predicted diffusion trajectory is more uniform than true *Occupy* trajectory.



a) *Fukushima*

b) *Arab Spring*

Fig. 4. τ_B vs. number of snapshots for a) *Fukushima*$(1:5)$, b) *Arab Spring*$(1:5)$ on *Top-Blog*

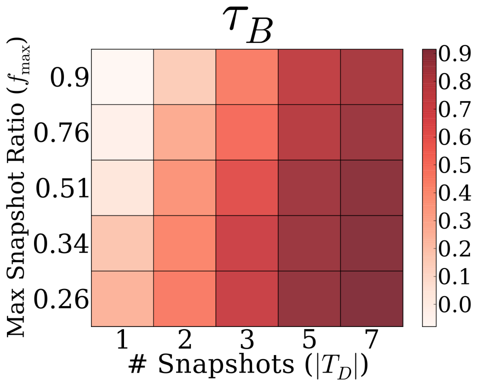


Fig. 5. τ_B vs. $|T_D|$ and f_{max} for *DHR-sub* of *Nba*$(1:10)$ on *Rand-Blog*

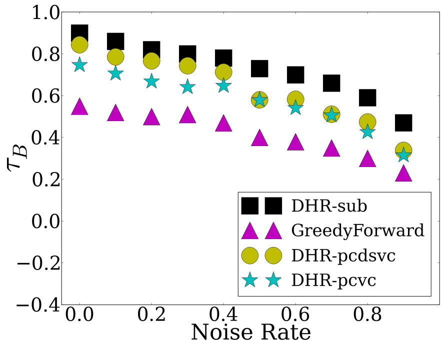
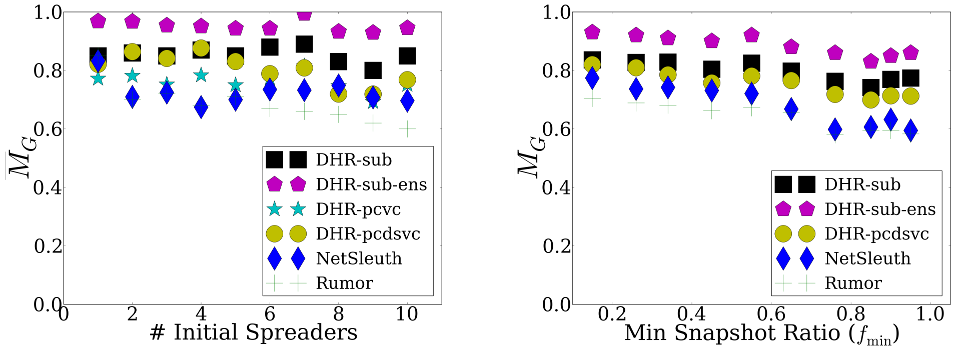


Fig. 6. τ_B vs noise ratio (p) over *Water-sm*

7.4. Identifying Initial Water Contamination Sites

We inferred the initial contaminant locations over two water distribution networks (Avi Ostfeld et al., 2008) where nodes are water demand-supply locations, and the edges represent the water pipes: *Water-sm* has 130 nodes and 173 edges, *Water-big* has 12527 nodes and 14595 edges. We used contaminant diffusion data generated by the water distribution simulator EPANET (Rossman, 1999).

We identified the initial contamination sites in *Water-sm* and *Water-big* under SIR where the *recovered* state models the dilution of the contaminant. We approximate the true hydraulic water diffusion dynamics by SIR as follows: we assume that $p_{uv} = K_1/l_{uv}$ and $i2r_{uv} = K_2/l_{uv}$ where K_1, K_2 are constants, and l_{uv} is the length of pipe (u, v). Ensemble methods perform the best as in Figure 7 on *Water-sm*, and *DHR-sub* (without ensemble) also performs better than the existing methods. Our methods' performance is consistent across different numbers of initial contamination sites whereas the existing methods' performance is affected by the number of initial sites. Our methods are nonparametric as they



a) *Water-sm* b) *Water-big*

Fig. 7. a) $\overline{M}_G$ vs. number of initial spreaders for *Water-sm*, b) $\overline{M}_G$ vs. f_{min} for *Water-big* (5 initial sites)

do not require number of initial spreaders as input, and our methods' performance consistency makes them the topmost candidates for application domains with multiple but unknown number of initial spreaders.

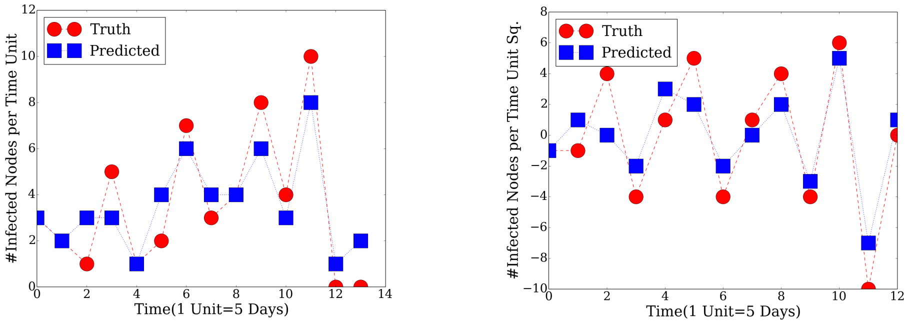
The performance of both *DHR-sub*-ens and *DHR-pcdsvc* decrease for higher f_{min} as in Figure 7 on *Water-big*, but they still perform at least 10% better than the best performing *NetSleuth*. This lower performance is due to both difficulty of differentiating between the initial spreader candidates with similar scores, and the decreasing ability to estimate the correct number of initial spreaders. Our methods may miss the true initial spreaders, but their estimates are within close distance to the original spreaders as reflected by higher performance in various cases.

7.5. Predicting Temporal Diffusion Features

We may answer questions related to temporal diffusion features from the reconstructed histories such as How quickly did it spread over time?, Did it spread faster at the beginning slowing down at later time steps?, etc. Here, we compared the speed (first-order) and acceleration (second-order) dynamics of *Unemployment* and *Fukushima* estimated from *DHR-sub* with the true ones from (Gomez-Rodriguez et al., 2013) as in Figures (8)–(9). We define the speed of a meme as the number of blogs that publishes about the meme for the first time per time unit, and acceleration as the diffusion speed change per time unit.

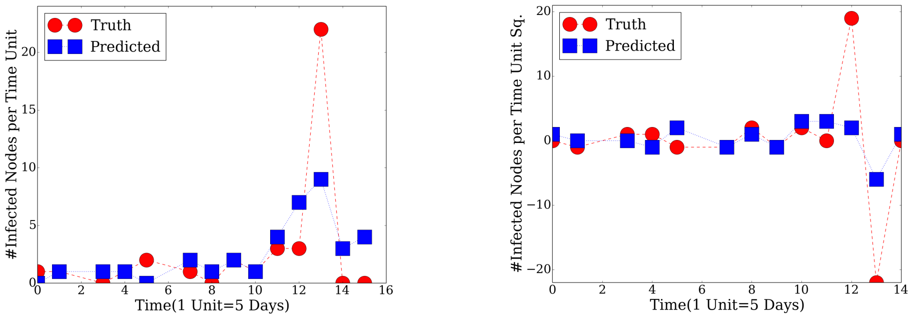
Unemployment is a more commonly-used meme than *Fukushima*, and such difference is reflected in their diffusion dynamics: *Unemployment*'s diffusion speed is more uniform over time whereas *Fukuhisma* shows more bursty dynamics. Diffusion speed of *Unemployment* has multiple local optima for the time points it peaks in news cycle whereas the speed of *Fukushima* has a single peak when it takes attention of the main media sites. However, such difference in diffusion dynamics does not make a difficulty for *DHR-sub* as *DHR-sub* predicted speed of both memes closely approximate their true ones even from 3 snapshots.

DHR-sub reconstructed histories of both memes also mimic closely their true acceleration dynamics. The change of diffusion speed for *Unemployment* is more uniform than the one for *Fukushima*, and its uniform dynamics are predicted al-



a) Speed b) Acceleration

Fig. 8. a) Speed, b) Acceleration Dynamics of True and Predicted Diffusion of *Unemployment* over Time from 3 snapshots



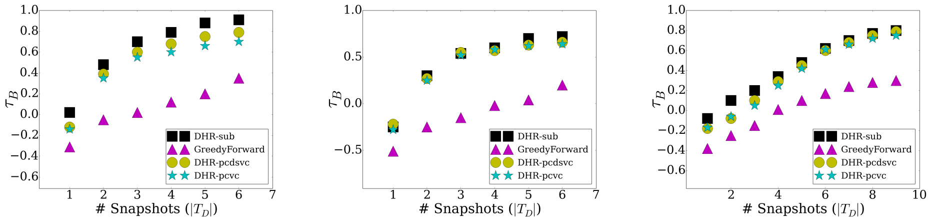
a) Speed b) Acceleration

Fig. 9. a) Speed, b) Acceleration Dynamics of True and Predicted Diffusion of *Fukushima* over Time from 3 snapshots

most perfectly by *DHR-sub* whereas the main peak of *Fukushima*'s acceleration dynamics was missed by *DHR-sub* except precise approximation at remaining time points. Overall, *DHR-sub* reconstructed histories from only 3 snapshots mimic the true speed and acceleration dynamics of both memes quite precisely even though the original prediction scores are below 0.75 ($\tau_B = 0.74$ for *Unemployment*, $\tau_B = 0.65$ for *Fukushima*).

7.6. Reconstructing Meme Diffusion by Training with Another Meme

We also evaluate how well diffusion history of a meme can be revealed by learning the diffusion model parameters over different memes by using the same blog dataset in the previous sections. In realistic scenarios, one may not always know the diffusion parameters but instead it can be learned over other possibly semantically closer memes which diffusion histories can be better recorded. We learn the diffusion parameters of meme by running InfoPath (Gomez-Rodriguez et al., 2013) over a number of cascades. We assume that diffusion model is SI, and distribution is exponential which corresponds to geometric distribution in



(a) Predicting *Fukushima* from *Earthquake*

(b) Predicting *Fukushima* from *Islamist*

(c) Predicting *Uprising* from *Unrest*

Fig. 10. Meme diffusion history prediction by training with another meme. a) Predicting *Fukushima* from cascades including *Earthquake*, b) Predicting *Fukushima* from *Islamist*, c) Predicting *Uprising* from *Unrest*.

our discrete setting. Figure 10a shows the performance of predicting the history of *Fuhushima* over increasing number of snapshots by training with 187 cascades including the keyword *Earthquake* that have spreaded to at least 100 blogs. Accurate prediction performance in the figure can be explained by semantic similarity of both memes which possibly leads to similar diffusion progression patterns.

If we train with same number of cascades including *Islamist* instead, there is a major performance decrease as in Figure 10b showing the importance of semantic similarity between trained and inferred memes. However, semantic similarity may not be enough on its own to achieve an accurate performance as in Figure 10c. Even though *Uprising* and *Unrest* are semantically similar, performance is lower than Figure 10a which shows the importance of factors other than semantic similarity in diffusion progression. In general, increasing the number of *Earthquake* training cascades leads to more accurate prediction of *Fuhushima* history until 25 cascades after which performance becomes almost stable. Overall, our analysis suggests that diffusion parameters of a semantically closer meme can be used to predict diffusion history of a meme even if its diffusion parameter estimates are not available a priori.

7.7. Scalability and Robustness of History Reconstruction

All our methods reconstruct the history on *Top-Blog* in less than 5 minutes, and our relaxation methods *DHR-pcdsvc*, *DHR-pcvc* reconstruct the history in less than 12 minutes on a large 2D grid graph having 90000 nodes and 179400 edges, with reasonable performance ($\tau_B = 0.71, 0.63$) as in Table 3 whereas *DHR-sub* takes more than an hour on a personal laptop. When combined with previous sections' results, running times in Table 3 suggest that *DHR-pcdsvc* and *DHR-pcvc* are nice alternatives to *DHR-sub* for scalable history reconstruction on large graphs. However, we still need faster methods for scalable reconstruction on million-node graphs.

Figure 6 shows the performance of contaminant diffusion history reconstruction over *Water-sm* under increasing noise levels. Let p be the noise ratio between 0.0 and 1.0, and we added the synthetic noise p as follows: For each node and each state, we randomly select a value m between 0 and pl_D where l_D is length of the diffusion and flip a coin to either add m to the current state transition

	Top-Blog		*2D-GRID*	
	$\lvert T_D\rvert = 1$	$\lvert T_D\rvert = 3$	$\lvert T_D\rvert = 1$	$\lvert T_D\rvert = 3$
DHR-sub	562.4	147.4	-	-
DHR-pcdsvc	215.2	112.9	592.1	199.2
DHR-pcvc	150.4	42.8	351.7	82.7

Table 3. History reconstruction time (in seconds) for *Top-Blog* and a 2D grid graph for different numbers of diffusion snapshots.

time t_v, or subtract it from t_v. If modified transition time $(t_v + m)$ is less than 0, we make it 0.

Our methods do not show a sudden performance drop by increasing noise levels, as *DHR-sub* can still reconstruct histories with performance over $\tau_B = 0.7$ even when the noise levels are 0.5. Similarly, *DHR-sub*-ens achieves $\overline{M}_G = 0.72$ in identifying the initial contaminant locations over *Water-sm* for $p = 0.5$ (results are not shown). In general, all our methods are robust to the noise in the data.

8. Conclusions

We designed several methods for estimating diffusion histories that either optimize the likelihood or its relaxations with provable performance guarantees for local steps. Our methods do not require the number of initial spreaders and diffusion length as parameters. They identify the initial spreaders better than the existing methods specially designed for this task. They reconstruct the history accurately in a number of scenarios. We also accurately estimated temporal diffusion characteristics of several semantically different memes from partial data. These findings suggest the reconstructability of diffusion history from partial data under several settings. Partial diffusion data is not an unsolvable bottleneck as missing diffusion history can be completed by our methods accurately.

Acknowledgements. We thank anonymous reviewers for their very useful comments and suggestions. This work has been partially funded by the US National Science Foundation (CCF-1256087, CCF-1319998) and US National Institutes of Health (R21HG006913 and R01HG007104). C.K. received support as an Alfred P. Sloan Research Fellow. A preliminary version of this paper appeared in ICDM2014 (Sefer and Kingsford, 2014).

References

Agresti, A. (2002), *Categorical Data Analysis*, Wiley Series in Probability and Statistics, 2nd edn, Wiley-Interscience.

Avi Ostfeld et al. (2008), 'The battle of water sensor networks (bwsn): A design challenge for engineers and algorithms', *Journal of Water Resources Planning and Management* **134**(6), 556–568.

Barabási, A.-L. and Albert, R. (1999), 'Emergence of scaling in random networks', *Science* **286**(5439), 509–512.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E. (2008), 'Fast unfolding of communities in large networks', *J. of Statistical Mechanics: Theory and Experiment* **2008**(10), P10008+.

Boykov, Y. and Kolmogorov, V. (2004), 'An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **26**(9), 1124–1137.

Buchbinder, N., Feldman, M., Naor, J. S. and Schwartz, R. (2012), A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization, *in* '2012 IEEE 53rd Annual Symposium on Foundations of Computer Science', IEEE, pp. 649–658.

Erdos, P. and Rnyi, A. (1960), On the evolution of random graphs, *in* 'Publication of the Mathematical Institute of the Hungarian Academy of Sciences', pp. 17–61.

Feige, U. (1998), 'A threshold of ln n for approximating set cover', *Journal of the ACM* **45**(4), 634–652.

Feige, U. and Goemans, M. (1995), Approximating the value of two power proof systems, with applications to max 2sat and max dicut, *in* 'Theory of Computing and Systems, 1995. Proceedings., Third Israel Symposium on the', pp. 182–189.

Feige, U., Mirrokni, V. S. and Vondrak, J. (2007), Maximizing non-monotone submodular functions, *in* 'Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science', FOCS '07, IEEE Computer Society, Washington, DC, USA, pp. 461–471.

Gomez-Rodriguez, M., Leskovec, J. and Krause, A. (2012), 'Inferring Networks of Diffusion and Influence', *ACM Transactions on Knowledge Discovery from Data* **5**(4), 1–37.

Gomez-Rodriguez, M., Leskovec, J. and Schölkopf, B. (2013), Structure and dynamics of information pathways in online media, *in* 'WSDM '13', ACM, New York, NY, USA, pp. 23–32.

Gupta, A., Roth, A., Schoenebeck, G. and Talwar, K. (2010), 'Constrained non-monotone submodular maximization: Offline and secretary algorithms', *CoRR* **abs/1003.1517**.

Hethcote, H. W. (2000), 'The mathematics of infectious diseases', *SIAM Rev.* **42**(4), 599–653.

Hochbaum, D. S. (2000), Instant recognition of polynominal time solvability, half integrality and 2-approximations, *in* 'APPROX '00', Springer-Verlag, pp. 2–14.

IBM ILOG CPLEX Optimizer (2010).
URL: *http://www.ilog.com/products/cplex/*

Kempe, D., Kleinberg, J. and Tardos, E. (2003), Maximizing the spread of influence through a social network, *in* 'Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, New York, NY, USA, pp. 137–146.

Kolmogorov, V. and Zabih, R. (2004), 'What energy functions can be minimized via graph cuts', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 65–81.

Lappas, T., Terzi, E., Gunopulos, D. and Mannila, H. (2010), Finding effectors in social networks, *in* 'KDD '10', ACM Press, New York, New York, USA, p. 1059.

Lee, J., Mirrokni, V. S., Nagarajan, V. and Sviridenko, M. (2009), Non-monotone submodular maximization under matroid and knapsack constraints, *in* 'Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing', ACM, New York, NY, USA, pp. 323–332.

Leskovec, J., Adamic, L. A. and Huberman, B. A. (2007), 'The dynamics of viral marketing', *ACM Trans. on the Web* **1**(1), 5.

Leskovec, J., Kleinberg, J. and Faloutsos, C. (2005), Graphs over time: densification laws, shrinking diameters and possible explanations, *in* 'KDD '05', ACM, New York, NY, USA, pp. 177–187.

Prakash, B. A., Chakrabarti, D., Faloutsos, M., Valler, N. and Faloutsos, C. (2011), Threshold conditions for arbitrary cascade models on arbitrary networks, *in* 'Proceedings of the 2011 IEEE 11th International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 537–546.

Prakash, B. A., Vreeken, J. and Faloutsos, C. (2012), Spotting culprits in epidemics: How many and which ones?, *in* 'ICDM', pp. 11–20.

Rossman, L. (1999), The epanet programmer's toolkit for analysis of water distribution systems, *in* 'WRPMD'99', pp. 1–10.

Salathé, M., Kazandjieva, M., Lee, J. W., Levis, P., Feldman, M. W. and Jones, J. H. (2010), 'A high-resolution human contact network for infectious disease transmission.', *Proc. Natl. Acad. Sci. USA* **107**(51), 22020–5.

Schrijver, A. (2003), *Combinatorial Optimization - Polyhedra and Efficiency*, Springer.

Sefer, E. and Kingsford, C. (2014), Diffusion archaeology for diffusion progression history reconstruction, *in* 'Data Mining (ICDM), 2014 IEEE International Conference on', pp. 530–539.

Serazzi, G. and Zanero, S. (2003), Computer virus propagation models, *in* 'In Tutorials of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS03)', Springer-Verlag.

Shah, D. and Zaman, T. (2011), 'Finding Rumor Sources on Random Graphs', *arXiv* p. 1110.6230.

Wolsey, L. A. and Nemhauser, G. L. (1999), *Integer and Combinatorial Optimization*, Wiley-Interscience.

Yang, J. and Leskovec, J. (2011), Patterns of temporal variation in online media, *in* 'ACM International Conference on Web Search and Data Minig (WSDM)', pp. 177–186.

Author Biographies

insert photo

Emre Sefer received a B.E. degree from Bogazici University, Istanbul, Turkey. He also received a M.S. degree from University of Maryland College Park, and Ph.D. from Carnegie Mellon University School of Computer Science. He is currently a post-doctoral researcher at Computer Science Department, Carnegie Mellon University, Pittsburgh PA. His main research interests include systems biology, data mining, machine learning applications on social networks.

insert photo

Carl Kingsford is an Associate Professor in Computational Biology Department in Carnegie Mellon University. He has received NSF Career Award, UMD Department of Computer Science "Teaching Excellence Award for a Professor", and Alfred P. Sloan Fellowship. He has served as a member of program committee of Recomb and ISMB; he has published over 50 refereed papers. His research interests include large-scale genomics and systems biology.

Correspondence and offprint requests to: Emre Sefer, Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Email: esefer@cs.cmu.edu