Not Enough Data? Joint Inferring Multiple Diffusion Networks via Network Generation Priors

Xinran He
University of Southern California
Los Angeles, CA ,United States
xinranhe@usc.edu

Yan Liu
University of Southern California
Los Angeles, CA ,United States
yanliu.cs@usc.edu

ABSTRACT

Network Inference, i.e., discovering latent diffusion networks from observed cascades, has been studied extensively in recent years, leading to a series of excellent work. However, it has been observed that the accuracy of existing methods deteriorates significantly when the number of cascades are limited (compared with the large number of nodes), which is the norm in real world applications. Meanwhile, we are able to collect cascades on many different topics or over a long time period: the associated influence networks (either topic-specific or time-specific) are highly correlated while the number of cascade observations associated with each network is very limited. In this work, we propose a generative model, referred to as the *MultiCascades* model (MCM), to address the challenge of data scarcity by exploring the commonality between multiple related diffusion networks. MCM builds a hierarchical graphical model, where all the diffusion networks share the same network prior, e.g., the popular Stochastic Blockmodels or the latent space models. The parameters of the network priors can be effectively learned by gleaning evidence from a large number of inferred networks. In return, each individual network can be inferred more accurately thanks to the prior information. Furthermore, we develop efficient inference and learning algorithms so that MCM is scalable for practical applications. The results on both synthetic datasets and real-world datasets demonstrate that MCM infers both topic-specific and time-varying diffusion networks more accurately.

1. INTRODUCTION

Understanding the processes and dynamics of *information diffusion* through networks plays a fundamental role in a variety of domains, such as evaluating the effects of networks in marketing [5, 17, 19], monitoring the spread of news, opinions, and scientific ideas via citation networks [1, 13, 22], and detecting the spread of erroneous information [6]. In practical applications, the underlying diffusion network (e.g. networks on who influenced whom) is often hidden. Therefore,

many interesting models have been developed to automatically infer diffusion networks from observed cascades, i.e., time stamps when users post a blog on certain topic or purchase products [10, 9, 29, 30, 26, 4, 8, 7]. Popular models include cascade models, for example, *NetInf* [10] and *NetRate* [9], and point process models such as *MMHP* [29] and *LowRankSparse* [30].

Given the complexity of the network inference problem, most existing algorithms need a large number of cascades to achieve a good performance. For example, for a network with a hundred nodes, at least a thousand cascades are required for reasonable inference performance [10]. In practical applications, the amount of data is usually insufficient (relative to the large number of nodes in the diffusion networks). Therefore, it is not surprising that the inference accuracy is significantly lower in real-world applications (compared with synthetic dataset results) [10, 9]. Several recent works have been developed to address the data scarcity issue through sparsity regularization [4, 9, 30] and low-rank structure constraint [30]. However, most of these models rely on strong assumptions of the network structures, which may be easily violated in real applications. In addition, the performance improvement by these models could be limited [30, 4] since one structure property may not be enough to compensate for the limited number of cascades, and it is extremely difficult to incorporate several network properties into these models.

In order to address this problem, we notice that we do have access to cascade observations from a large number of diffusion networks over different topics or a long time period: the associated influence networks (either topic-specific or time-specific) are highly correlated even though the averaged number of observations associated with each network is limited. Therefore, it would be interesting to investigate how this large-scale heterogeneous data collection can help effectively infer the latent diffusion networks. In this paper, we propose a novel generative model referred to as the *MultiCascades* model (MCM). Instead of assuming independence between the cascades from different networks and carrying out network inference separately for each network, MCM jointly infers the networks by exploiting the commonalities between them. Specifically, it builds a hierarchical graphical model, where all the diffusion networks share one network prior. The cascades are then generated from the corresponding diffusion network.

MCM provides a systematic and flexible framework to take advantage of many key findings in network analysis, such as heavy-tail degree distribution, community structure,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM 2017, February 06-10, 2017, Cambridge, United Kingdom

© 2017 ACM. ISBN 978-1-4503-4675-7/17/02...\$15.00

DOI: <http://dx.doi.org/10.1145/3018661.3018675>

local clustering property and small world phenomenon [16, 28], for effective network inference. In this paper, we will use Stochastic Blockmodels (SBM) [27] and latent space models (LSM) [15, 18] as examples to illustrate the power of our approach. MCM captures the commonalities among different networks through the network priors by jointly inferring the unknown but shared parameters in the network generation models from all the cascades. Conversely, the shared prior with many important properties of networks can effectively mitigate the problem of data sparsity. To the best of our knowledge, this is one of the first models that can effectively utilize both observation data and network generation models for network inference.

For effective inference and learning, we apply the EM algorithm to estimate the parameters of the graph generation model (i.e., the network prior) and the diffusion networks. Since it is intractable to carry out exact posterior inference in the E-step, we use the MAP estimation instead of marginalization [24]. The MAP formulation allows us to reuse existing network inference algorithms as building blocks. In M-step, we update the parameters of the graph generation model based on the inferred diffusion networks in the E-step. We carry out empirical evaluation on both synthetic datasets and two real-world datasets, Twitter and MemeTracker. Experiment results show that our MCM model achieves significant improvement in accuracy compared to several state-of-the-art baseline algorithms by jointly inferring multiple diffusion networks.

2. RELATED WORK

Our work is built on top of many excellent prior work, which we summarize into three categories:

Network inference: There has been a significant amount of work on inferring the information diffusion network where either cascade models [10, 9, 4] or point processes [29, 30, 14] are used. The network inference algorithms either aim at inferring the structure of the diffusion network, such as NetInf [10] and MultiTree [12] algorithm or simultaneously inferring the strength of influence between nodes, such as NetRate [9]. However, all algorithms above infer a *single* diffusion network by treating all the cascades associated with one graph while in practice, it is necessary and beneficial to infer multiple diffusion networks.

Network inference with prior knowledge: Motivated by the distinct properties exhibited in social networks, many algorithms have incorporated prior knowledge to improve the accuracy of diffusion network inference [11, 4, 9, 30]. For example, [11, 4, 9] introduce L_1 regularization in the maximal likelihood estimation motivated by the sparsity of the social networks. Recently, Zhou et al. propose the *LowRankSparse* algorithm [30] utilizing the fact that the adjacent matrix has low rank structure that captures the community structure in the diffusion network. However, many other useful properties, such as heavy-tail degree distributions, community structure, local clustering property and small world phenomenon [16, 28], remain unexplored in the previous line of work. On the contrary, our MCM provides a general framework to incorporate the prior knowledge with a wide choices of possible graph generation models.

Heterogeneous influence: Recently, a line of work has considered the heterogeneity of influence in social networks [8, 7, 26]. For example, the *TopicCascades* algorithm [8] focuses on inferring diffusion network from the text-based cascades

where the diffusion rate depends on the similarity of the contents. *KernelCascade* algorithm [8] assumes the delay distributions are different for different edges in the diffusion network. Moreover, Wang et al. propose the innovative *MMRate* algorithm to carry out multi-aspect multi-pattern network inference [26]. In MMRate, the transmission rate depends on both the cascade patterns and the aspect of the diffusion networks. Then a mixture model is proposed, where the hidden variable determines under which aspect the cascade is propagated. The focus of MMRate is to automatically infer the aspect of each cascade and then infer the aspect-specific diffusion networks independently. In contrast, we answer the question on how to jointly infer multiple diffusion networks with limited number of observations. Moreover, the performance of MMRate is evaluated by combining all aspects and it is noted that MMRate performs worse than baselines for diffusion networks with smaller number of cascades [26]. In contrast we show that our MCM can infer multiple diffusion networks accurately even with limited number of observations, suggesting the necessity of adopting powerful network priors.

3. PROBLEM FORMULATION

Considering a network with $|V|$ nodes, traditional network inference algorithms focus on inferring a *single* hidden diffusion network G from observed cascades $C = \{c_1, \dots, c_{|C|}\}$. Each cascade c_i provides the trace of the propagation among the nodes in the network, i.e., $c_i = \{(v_j, t_j)\}$ as node v_j is activated at time t_j .

In this work, we assume that there are multiple diffusion networks $\mathcal{G} = \{G_1, \dots, G_M\}$. All diffusion networks share the same set of individuals V but differ in their edges. We use $G_i = (V, E_i)$ to denote diffusion network i . The multiple diffusion networks can for example capture topic-specific influence strength or be multiple snapshots of one dynamic diffusion network. Let $C_i = \{c_{i,1}, \dots, c_{i,|C_i|}\}$ be the set of cascades generated from network i . We focus on simultaneously inferring *multiple* diffusion networks $\mathcal{G} = \{G_1, \dots, G_M\}$, i.e., the edge set E_i for each network G_i , based on the collections of cascades generated from all the networks $\mathcal{C} = \{C_1, \dots, C_M\}$. Without loss of generality, we assume that the information on which network a cascade is generated from is given.

For ease of presentation, we will use the widely adopted continuous-time Independent Cascade (CIC) model as the diffusion model [9, 10, 23]. In the CIC model, each edge (u, v) is associated with a delay distribution with parameter $\alpha_{u,v}$. When a node u becomes newly activated at time t_u , for every neighbor v that is still inactive, a delay time Δt is drawn from the delay distribution as the duration taken to activate its neighbor v . Namely, node v is guaranteed to be activated at time $t_u + \Delta t$. If multiple nodes succeed in activating the same inactive node, the activation of the node is taken as the earliest one. As we aim at inferring the edges of the diffusion networks, we assume the delay parameters for all edges are the same. This assumption is also acknowledged in [9, 12]. In this paper, we use the CIC model as a concrete example to demonstrate our framework. Our method can be easily extended to other network inference models, such as discrete-time Independent Cascade (DIC) model [25] and also the recent proposed Hawkes Process models [29, 30].

4. MULTICASCADES MODEL

Our MCM is mostly motivated by the following two observations. First, the associated diffusion networks are highly correlated with each other and share a lot of commonalities. As a illustration, we visualize four Twitter retweet networks corresponding to the propagation of different topics in Figure 1. (The detailed description of the dataset is provided in Section 5.2.) The community in the center of the plot is shared by Iran, Haiti and Climate networks, while the US politics network shares the left community with both the Iran and Haiti networks. In our MCM model, we capture the commonality by assuming all associated networks are generated from the same network model with the same set of parameters. Moreover, we infer the parameters jointly from the ensemble of all cascades to mitigate the data scarcity issue. Second, the diffusion networks indicate distinct properties such as the community structure and local clustering properties [16, 28]. MCM can easily incorporate the prior knowledge on the structure of the diffusion networks by choosing which type of graph generation models to use as the prior. For example, we can use the Stochastic Blockmodels to capture the community structure of the diffusion networks [27] in Figure 1.

Based on the intuition above, we construct MCM with the following generative process: first, we generate the diffusion networks from a parametric network generation model with the parameter Θ , and then generate the cascades $C_i = \{C_{i,j}\}$ from the diffusion network G_i according to the CIC model. The graphical representation of MCM is shown in Figure 2.

In order to carry out multiple network inference under our MCM, we notice that the diffusion networks are not observed and therefore EM algorithms are applied. That is, we estimate the parameters of the network generation model from the ensemble of all cascades as follows:

$$\Theta^* = \arg \max_{\Theta} P(\mathcal{C}|\Theta) = \arg \max_{\Theta} \sum_{\mathcal{G}} P(\mathcal{C}, \mathcal{G}|\Theta).$$

Imposing the network generation model as prior, we then infer each diffusion network G_i as an MAP estimation problem:

$$G_i^* = \arg \max_{G_i} P(G_i|C_i, \Theta).$$

The joint inference of network prior from the ensemble of all cascades allows us to transfer information between cascades in different groups to alleviate the data scarcity. The network prior allows us to incorporate our prior knowledge on the structure of the diffusion networks. Moreover, it serves as a regularizer to prevent overfitting when the number of observations is small.

However, exact posterior inference in the E-step is intractable for most choices of network priors and diffusion models. Since we need to marginalize over exponentially many possible graphs. Thus, we propose to use MAP estimation instead of marginalization:

$$G_i^{(t)} = \arg \max_{G_i} P(G_i|C_i, \Theta^{(t-1)}),$$

where $\Theta^{(t-1)}$ is the estimated parameters from the previous M-step. Though sampling methods, such as MCMC can be used to approximate the posterior distribution, they are usually not scalable due to the slow convergence in sampling over super-exponential space of all possible graphs.

Algorithm 1 EM Algorithm for MCM

```

1: Inputs: Observed cascades of different types  $\mathcal{C}$ 
2: Outputs: Inferred networks  $\mathcal{G} = \{G_1, \dots, G_M\}$ , estimated prior parameters  $\Theta$ 
3: Initialized  $\Theta^{(0)}$ 
4: for fixed rounds or convergence do
5:   // E Step
6:   for  $i = 1, \dots, M$  do
7:      $G_i^{(t+1)} \leftarrow \arg \max_{G_i} P(G_i|C_i, \Theta^{(t)})$ 
8:   end for
9:   // M step
10:   $\Theta^{(t+1)} \leftarrow \arg \max_{\Theta} P(\mathcal{C}, G_1^{(t+1)}, \dots, G_M^{(t+1)}|\Theta)$ 
11: end for

```

Moreover, the MAP formulation allows for reusing existing network inference algorithms with only small adaption. As summarized in Algorithm 1, the EM algorithm using MAP estimation as E-step is equivalent to maximizing the joint likelihood $P(\mathcal{C}, \mathcal{G}|\Theta) = \prod_{i=1}^M P(C_i|G_i)P(G_i|\Theta)$ with coordinate ascend method where we alternatively optimize over the diffusion networks $\mathcal{G}$ and the prior parameter Θ .

The EM algorithm above provides a general framework to infer multiple diffusion networks with a wide choices of possible network priors. The two requirements of the graph generation model to be used as network prior in our model are as follows: (1) There exists efficient algorithm for model parameter estimation; (2) The model provides a probabilistic adjacency matrix which determines the existence of each edge independently. Let P_{Θ} be the probabilistic adjacency matrix where $p_{u,v}$ is the probability edge (u, v) exists in the graph. This requirement suggests that the probability $P(G_i|\Theta)$ takes the following forms:

$$P(G_i|\Theta) = \prod_{(u,v) \in E_i} p_{u,v} \prod_{(u,v) \notin E_i} (1 - p_{u,v}).$$

Many widely used graph generation models naturally satisfy the requirements. Examples include Stochastic Blockmodels [27], latent space models [15] and Kronecker graph models [21]. Moreover, our framework is flexible enough to use mixture of graph generation models as prior. However in this paper, we only use single graph generation model as prior.

Next, we show how to adapt the *NetInf* algorithm and the *MultiTree* algorithm for MAP inference in the E-step. We then use Stochastic Blockmodels and latent space models as two concrete examples of network priors and illustrate how we implement parameter updates in the M-step.

4.1 MAP Inference in E-step

In the E-step, we carry out the MAP inference of the diffusion network G_i given the cascade observations C_i in group i and the network prior with parameters Θ . We first briefly review the *NetInf* algorithm [10] and the *MultiTree* algorithm [12] and then demonstrate that they can be easily adapted for the MAP inference with minimum changes. We omit the subscript i in both diffusion networks G_i and cascades C_i below to simplify the notation.

Both *NetInf* and *MultiTree* algorithms infer the diffusion networks by maximizing the likelihood function, leading to

Figure 1: Visualization of retweet networks of different topics

Figure 2: Graphical representation of our MultiCascades model. Θ is the parameter for the network generation model; G_i is the hidden diffusion network for cascades in graph i ; $C_{i,j}$ represents the j -th observed cascades in group i .

the following combinatorial optimization problem:

$$\begin{aligned} & \text{Maximize} && P(C|G) = \prod_{i=1}^{|C|} P(c_i|G) \\ & \text{Subject to} && G = (V, E), |E| \leq k. \end{aligned}$$

As the problem is intractable due to its super-exponential search space, an approximation to the likelihood function is proposed by assuming all cascades follow a tree structure:

$$\begin{aligned} \prod_{c \in C} P(c|G) &\approx \prod_{c \in C} \left[\sum_{T \in T(G)} P(c|T) P(T|G) \right] \\ &\propto \sum_{T \in T(G)} \prod_{(i,j) \in T} P_c(i,j), \end{aligned}$$

where $T(G)$ is the set of all directed spanning trees in G . The approximate likelihood can be further approximated via the maximum spanning tree (e.g. NetInf) or calculated exactly with the matrix tree theorem (e.g., MultiTree). Then, both algorithms maximize the approximated likelihood $F_C(G)$ by a greedy algorithm. To be more specific, the algorithms start from an empty network and iteratively add edge that maximizes the marginal gain. As the objective function $F_C(G)$ is monotone and submodular, this procedure results in a $1 - 1/e$ approximation guarantee [10, 12].

In order to adapt these algorithms for the MAP inference in the E-step of our MCM model, we combine the objective function together with the likelihood corresponding to our network prior. The new objective function $F'_C(G)$ takes the following form:

$$\begin{aligned} F'_C(G) &= F_C(G) + \log P(G|\Theta) \\ &= F_C(G) + \sum_{(u,v) \in E_i} \log p_{u,v} + \sum_{(u,v) \notin E_i} \log(1 - p_{u,v}). \end{aligned}$$

Since $\log P(G|\Theta)$ is a modular function in the selected edges, the new objective function $F'_C(G)$ is still submodular as it is the summation of a submodular function and a modular function. However, the objective function $F'_C(G)$ is no

longer guaranteed to be monotone. As a result, the greedy algorithm fails to provide the approximation guarantee.

To address this issue, we instead use the Random Greedy algorithm proposed in [3] to solve this non-monotone submodular optimization problem. Instead of greedily adding the edge with the maximum marginal gain in each iteration, the Random Greedy algorithm first finds all candidates with large marginal gains and then randomly pick one to include. It has been shown in [3] that the Random Greedy algorithm achieves $1/e$ approximation guarantee.

4.2 Parameter Update in M-step

In this section, we use the Stochastic Blockmodels [27] and the latent space models [15, 18] as two concrete examples of the network priors and demonstrate how to update the model parameter Θ in the M-step.

4.2.1 Latent space model

The first example model we consider is the latent space models [15, 18]. Under this model, the network is assumed to be embedded in a D dimensional latent space. The closer two nodes are in the latent space, the more likely there is an edge between the two nodes in the diffusion networks. Under this model, the parameter Θ is simply a V -by- D node location matrix. Each row x_u of the matrix provides the position of node u in the D dimensional latent space. In this work, we assume each node position is drawn from a multivariate Gaussian distribution, i.e. $x_u \sim N(0, \sigma^2 I)$. Moreover, we use a Gaussian kernel to generate the probabilistic adjacency matrix from the node positions:

$$P_{u,v} = \exp\left(-\frac{\gamma}{2} \|x_u - x_v\|_2^2\right).$$

In the M-Step, we update parameter Θ by maximizing the log likelihood of the inferred diffusion networks $\mathcal{G} = \{G_1, \dots, G_M\}$:

$$\begin{aligned} L &= \sum_{i=1}^M \left[\sum_{(u,v) \in E_i} \log P_{u,v} + \sum_{(u,v) \notin E_i} \log(1 - P_{u,v}) \right] \\ &\quad + \sum_{v \in V} \frac{1}{2\sigma^2} \|x_u\|_2^2 \end{aligned}$$

The optimization is carried out via the gradient ascend method. Let $\ell_{u,v} = \sum_{i=1}^M (\mathbb{I}_{G_i}[u \rightarrow v] + \mathbb{I}_{G_i}[v \rightarrow u])$ and $\bar{\ell}_{u,v} = 2M - \ell_{u,v}$, where $\mathbb{I}_{G_i}[u \rightarrow v] = 1$ if there is an edge from u to v in diffusion network G_i . The gradient for each node position x_i is as follows:

$$\frac{\partial L}{\partial x_u} = \sum_{v \in V} \left[\gamma (x_u - x_v) \left(\frac{\ell_{u,v} P_{u,v}}{1 - P_{u,v}} - \bar{\ell}_{u,v} \right) \right] - \frac{1}{\sigma^2} x_u$$

4.2.2 Stochastic Blockmodels

We use Stochastic Blockmodels [27] as another example of the network prior. Stochastic Blockmodels assume that each node v in the network belongs to one of the K communities. The probability of an edge from node u to v depends on the community identity of the nodes and the K -by- K Bernoulli rates matrix B , where $B_{i,j}$ corresponds to the probability of having an edge from a node in community i to a node in community j . We organize the community identity for all nodes into a $|V|$ -by- K community indicator matrix $Z = [z_1^T, \dots, z_{|V|}^T]^T$, where z_u is the community identity of node u in the 1-of- K representation. As a result, the model parameters consist of both the community indicator matrix and the Bernoulli rates matrix. Moreover, the relationship between the probabilistic adjacency matrix P_Θ and the parameters $\Theta = (B, Z)$ can be represented concisely as:

$$P = ZBZ^T.$$

The optimization of the parameters $\Theta = (B, Z)$ is carried out by the stochastic EM algorithm. The update of node u 's community identity via Gibbs sampling in the E-step is as follows:

$$\begin{aligned} \mathbb{P}[z_{u,k} = 1 | Z_{-u}, \mathcal{G}] \propto \\ \sum_{i=1}^M \sum_{v \in V} (\mathbb{I}_{G_i}[u \rightarrow v] \log(B_{j, z_{v,k}}) + \mathbb{I}[u \nrightarrow v] \log(1 - B_{j, z_{v,k}})) \\ + \sum_{i=1}^M \sum_{v \in V} (\mathbb{I}_{G_i}[v \rightarrow u] \log(B_{z_{v,k}, j}) + \mathbb{I}[v \nrightarrow u] \log(1 - B_{z_{v,k}, j})) \end{aligned}$$

Here, we use $z_{v,k}$ to denote the community that node v_k is in.

The analytic solution for optimizing matrix B in M-Step is as follows:

$$B_{i,j} = \frac{\sum_{i=1}^M \sum_{u \in V} \sum_{v \in V} \mathbb{I}_{G_i}[u \rightarrow v] \mathbb{I}[z_{u,i} = 1, z_{u,j} = 1]}{\sum_{i=1}^M \sum_{u \in V} \sum_{v \in V} \mathbb{I}[z_{u,i} = 1, z_{u,j} = 1]}.$$

In our implementation, we use a slightly different approach by calculating the posterior distribution $P(Z|\mathcal{G}, B)$ by the Gibbs sampling. Then, the probabilistic adjacency matrix P_Θ is approximated as

$$\hat{P} = \mathbb{E}[Z]B\mathbb{E}[Z]^T.$$

We use this $\hat{P}$ in the MAP inference as it is more stable and robust.

4.3 Discussion

In this work, we focus only on inferring the structure of the diffusion networks, i.e., the edge set E_i for each network G_i , similar to the NetInf algorithm and the MultiTree algorithm. As a result, we assume the graph generation model induces a probabilistic adjacency matrix which we incorporate as a prior in the MAP inference in the E-step. In contrast, there are algorithms that infer both the edges and the strength of influence such as the *NetRate* algorithm [9], the *MMHP* algorithm [29] and the *LowRankSparse* algorithm [30]. These algorithms aim at inferring a real-value influence matrix A , where $A_{u,v}$ corresponds to the strength of influence from node u to node v .

Our MultiCascades model can be easily generalized to this case. Instead of providing the probability for the existence of edges, we require that the model imposes independent prior

Table 1: Comparison between Greedy algorithm and Random Greedy Algorithm

	Greedy	Random-Greedy
SBM-NetInf	0.461	0.459
SBM-MultiTree	0.461	0.458
Latent-NetInf	0.435	0.428
Laten-MultiTree	0.436	0.428

distribution on the influence strength of each edge. For example in the Stochastic Blockmodels, we can change the matrix B to provide the mean influence strength between nodes in different communities and generate the influence matrix $A_{u,v}$ from truncated normal distribution with mean B_{z_u, z_v} or Laplace distribution to induce sparsity. Accordingly, we can adapt the NetRate algorithm by combining the network prior and infer the diffusion networks via convex optimization as the MAP inference in the E-step.

Running time: The running time of our MCM algorithm depends on the time complexity of the components used in the EM algorithm. Empirically, we observed that our MCM converges in five to ten iterations. As a result, the running time of our MCM is only a constant factor slower than separately inferring all the networks. Since the E-step of network inference is the dominant term in the time complexity while the prior parameters learning in M-step only takes marginal time in our experiments. Moreover, the network inference of multiple networks in E-step can be trivially parallelizable leading to even faster implementation.

5. EMPIRICAL EVALUATION

For empirical evaluation, the main question we seek to answer is whether MCM can effectively infer multiple diffusion networks. First, we test our model on controlled synthetic datasets where the true diffusion networks are known. This set of experiments serve as proof of concept for our inference algorithm and show that incorporating network priors significantly improves the inference accuracy. Second, we provide the experiment results of MCM on two real-world datasets, the Twitter dataset and the MemeTracker dataset. We show that jointly inferring multiple networks leads to more accurate network discovery and better understanding of the diffusion dynamics.

Algorithms: We include the following algorithms for comparisons in our empirical evaluation¹:

- **MCM-[NetInf|Multi]-[SBM|LSM]:** Our MCM algorithm with NetInf/MultiTree for the MAP inference in the E-step using either SBM or Latent space model as the network prior.
- **Sep-[NetInf|Multi]:** Baseline algorithms that infer multiple diffusion networks independently using Netinf or MultiTree algorithm.
- **Single-[NetInf|Multi]:** Baseline algorithms that infer a single diffusion network by combining all cascades with NetInf or MultiTree algorithm.

¹We have also experimented with NetRate algorithm with L_1 regularization [4], which can be considered as incorporating independent Laplace prior. However, its performance is worse or similar to NetInf and MultiTree and omitted due to space limit.

Table 2: Network inference accuracy of synthetic datasets with 200 cascades.

		NetInf			MultiTree		
		Sep	Single	MCM	Sep	Single	MCM
SBM	maxF1	0.449	0.393	0.475(5.8%)	0.429	0.387	0.477(11.2%)
	AUC	0.300	0.254	0.377(25.7%)	0.300	0.244	0.373(25.7%)
Latent	maxF1	0.487	0.135	0.510(4.7%)	0.483	0.136	0.506(4.8%)
	AUC	0.338	0.060	0.359(6.2%)	0.330	0.061	0.368(11.5%)

Evaluation metrics: As all the algorithms greedily add edges one by one, we generate the precision and recall curve by evaluating at different number of edges. We then compute both the max F1 score and the area under the precision-recall curve (AUC) as metrics to evaluate the accuracy of the inferred networks, where the $F1$ score is defined as

$$F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

It should be noticed that here AUC corresponds to the area under the precision-recall curve instead of the receiver operating curve.

As Sep-[NetInf/Multi] and our MCM algorithms infer multiple diffusion networks, we evaluate the inferred network against the corresponding ground truth diffusion network and report the averaged max F1 score and the area under the precision-recall curve over all networks. In contrast, for Single-[NetInf/Multi] algorithms, we evaluate the inferred single network against all ground truth diffusion networks and report the averaged results.

5.1 Synthetic data

We first evaluate our method on synthetic data with a focus on the effectiveness of the inference algorithm and network priors in MCM.

Figure 3: Visualization for SBM prior. The darker the background color, the higher the value in the B matrix. The blue dots corresponds to the edges of one inferred diffusion network.

Data generation: We generate synthetic datasets using both Stochastic Blockmodels and the latent space model as graph generation priors (the two types of datasets are later referred as SBM and Latent). For Stochastic Blockmodels, we set the number of communities to 3 and the Bernoulli rates matrix B is set to 0.05 on the diagonal and 0.01 elsewhere. For the latent space model, we set the dimension

Table 3: Relative improvement in AUC with different $|C|$.

#Cascades	SBM	Latent
100	17.56±0.01%	8.98±0.01%
200	25.71±0.01%	6.21±0.01%
300	7.62±0.01%	7.51±0.71%
400	8.33±0.01%	0.30±0.01%
500	13.32±0.38%	4.18±0.01%

of latent space to 3 with $\gamma = 2.0$ and $\sigma = 1.0$. For both models, we generate ten different diffusion networks corresponding to ten sets of cascades. Each diffusion network has 256 nodes. We use exponential delay distribution with $\lambda = 0.1$ in the CIC model to generate the cascades with a observation window of length 10. We vary the number of cascades in each network from 100 to 500. For each setting, we randomly generate three datasets and report the mean for each evaluation metric.

Results: We first show the comparison between the simple Greedy algorithm and the Random Greedy method in the E-step in our MCM as in Table 1. The performance of the two methods is compared under both NetInf and MultiTree algorithms for both type of synthetic networks with the corresponding network prior. Under all cases, the simple Greedy method performs slightly better than Random Greedy, though the Random Greedy algorithm provides approximation guarantee. This can be explained by the observation that the objective function $F'_C(G)$ is approximately monotone, as the value of prior is usually smaller than the likelihood of cascades. Given the superior performance of the simple Greedy method, we only present the results of the simple Greedy method in following experiments.

Table 2 shows the experiment results of all algorithms on the synthetic datasets with 200 cascades. The first observation is that Single-NetInf and Single-Multi perform significantly worse than joint network inference algorithms due to the large variation in the structures of the diffusion networks in the synthetic datasets. Second, MCM achieves consistent improvement on the inference accuracy on both datasets across various number of cascades by incorporating the correct prior, i.e., the same model used in the network generation. In particular, it achieves more than 15% accuracy on average in the area under the PR curve on datasets compared to the running inference separately without prior. Moreover, we vary the number of cascades, i.e., the degree of data scarcity, from 100 to 500. The relative improvement of MCM-NetInf compared to Sep-NetInf in terms of AUC is shown in Table 3 average over three independent runs. The results of MultiTree is similar and therefore omitted. From the results, we can see that incorporating the network prior improves the inference accuracy significantly especially when there is only limit number of cascades for each network.

We further visualize the inferred network to show that they capture the structure of the diffusion networks. We first plot the inferred SBM prior overlaying together with one of the inferred diffusion networks by rearranging the nodes according to their community identity. The result is shown in Figure 3. Our MCM correctly infers the community identity for each node in the graph leading to the block structure along the diagonal.

Next, we visualize the inferred node positions in the latent space prior together with the growth process of one cas-

Table 4: Network inference accuracy of Twitter dataset.

		NetInf				MultiTree			
		Sep	Single	MCM-SBM	MCM-LSM	Sep	Single	MCM-SBM	MCM-LSM
Twitter100	maxF1	0.152	0.143	0.206	0.150	0.164	0.148	0.180	0.169
	AUC	0.074	0.062	0.112	0.088	0.097	0.072	0.107	0.100
Twitter250	maxF1	0.089	0.095	0.172	0.094	0.106	0.105	0.135	0.130
	AUC	0.027	0.037	0.066	0.044	0.0232	0.028	0.060	0.059

Table 5: Network inference accuracy of MemeTracker dataset.

		NetInf				MultiTree			
		Sep	Single	MCM-SBM	MCM-LSM	Sep	Single	MCM-SBM	MCM-LSM
MemeTracker128	maxF1	0.303	0.299	0.308	0.315	0.309	0.298	0.309	0.315
	AUC	0.182	0.178	0.193	0.194	0.177	0.185	0.193	0.194
MemeTracker256	maxF1	0.228	0.229	0.232	0.239	0.236	0.229	0.232	0.239
	AUC	0.122	0.123	0.129	0.131	0.127	0.123	0.129	0.130
MemeTrackerTopic	maxF1	0.287	0.136	0.233	0.393	0.300	0.135	0.349	0.339
	AUC	0.406	0.190	0.470	0.667	0.293	0.190	0.612	0.643

cade as is suggested in [18]. Figure 4 (a)-(d) show the active nodes (marked as red triangles) within 0%, 33%, 66%, 100% of the observation window. The results demonstrate that the node locations inferred by MCM satisfy the *influence preservation principle* [18], i.e., each node attempts to place its influential nodes relatively closer than non-influential ones in the visualization space.

5.2 Real data

Finally, we evaluate our approach on two real-world datasets, Twitter and MemeTracker. For the Twitter dataset, we aim at inferring multiple topic-specific diffusion networks. On the MemeTracker dataset, we aim at inferring both time-specific and topic-specific diffusion networks.

5.2.1 Twitter dataset

The Twitter networks are extracted from a complete collection of tweets between Jan. 2010 and Feb. 2010. We treat each hashtag as a separate cascade, and extract the top 100/250 users with the most tweets containing these hashtags into two datasets (Twitter100 and Twitter250). The pruning of the data sets is necessary because most users have showed very limited activity. The hashtags are manually grouped into five categories corresponding to major events/topics during the data collection period, where each group of hashtags is treated as one topic. The five groups are: Haiti earthquake (Haiti), Iran election (Iran), Technology, US politics, and the Copenhagen climate change summit (Climate). Examples of hashtags of each topic are shown in Table 6. It should be noticed that the inference task is very challenging as on average we only have about 30 observations to infer each topic-specific network. Given the true diffusion

Table 6: Examples of hastags in each category

Category	Hashtags
Iran	#iranelection, #iran, #16azar, #tehran
Haiti	#haiti, #haitiquake, #supphaiti, #cchaiti
Technology	#iphone, #mac, #microsoft, #tech
US politics	#obama, #conservative, #teaparty, #liberty
Climate	#copenhagen, #cop15, #climatechange

networks are not available, we use the retweet networks for each type of cascades as an alternative to the ground truth

(as adopted in many related works such as [2]). We apply all algorithms mentioned in the previous section and evaluate the inferred diffusion networks using the averaged max F1 and the area under the PR curve as in the synthetic experiments. We set the delay distribution in CIC to exponential distribution with $\lambda = 0.1$. The length of the observation window is set to 100 where each time unit corresponds to a day.

Results: The results on the Twitter dataset is summarized in Table 4. First, the result validates our claim that inferring multiple diffusion networks according to different topics leads to better accuracy. MCM using both SBM prior and LSM prior achieves improved accuracy for both evaluation metrics. It suggests that our MCM framework is not sensitive to the choice of network prior when applying to real-world data. The second observation is that incorporating network prior is necessary for accurate inference of multiple networks. Inferring each diffusion network independently (Sep-NetInf/Sep-MultiTree) performs slightly better than just inferring a single network on the smaller dataset with 100 node. However, they perform worse on the larger dataset with 200 nodes due to insufficient number of cascades. In contrast, MCM achieves consistent and significant improvement on both datasets. In particular, MCM-NetInf with the SBM prior achieves near 100% improvement on both Twitter100 and Twitter250 compared to Sep (i.e., inferring separate diffusion networks independently).

In order to provide further insight into the performance of our MCM model, we plot the precision-recall curve averaged over five diffusion networks in Figure 5. From the results, we can see that all the algorithms perform relatively well in choosing the first few edges with strong signals in the observed cascades. In contrast, MCM performs significantly better than the baselines when including more edges, as our algorithm is able to transfer the information from the cascades of other topics (or time stamps) by incorporating the network prior.

Moreover, we visualize the diffusion networks among top 10 users for all five topics in Figure 6. The plots further validate the necessity of inferring multiple diffusion networks as the influence structure varies significantly across different topic-specific networks. For example, both **persia_max_news** and **tehranweekly** are most influential in “Iran election” as

Figure 4: Visualization of cascade growth under the latent space model, where T is the length of the observation window. The blue circles corresponds to inactive nodes while the red triangles are the activated nodes.

Iran news agencies while `persia_max_news` has little influence in the propagation of other topics, such as “Haiti earthquake”.

5.2.2 MemeTracker dataset

The MemeTracker collection [20] contains memes extracted from Aug 2009 to Feb 2010 in Blogosphere and main-stream media sites. We preprocess the data to different datasets to test the performance of our approach on inferring both snapshots of dynamic diffusion networks and topic-specific networks.

Joint inference of dynamic networks: As the first experiment, we aim at inferring multiple snapshots of the dynamic diffusion network between blog sites and main-stream media. We extract the 128/256 sites with the most posting activity across the time period we study (MemeTracker128 and MemeTracker256). We group the cascades by their publishing month, leading to six groups of cascades. We focus on inferring the diffusion networks spanning over different months. Again, the true diffusion network is not available for this dataset, we use the citation networks (constructed from the hyperlinks between the blog posts during the same period) as an alternative to the ground truth network. That is, if site A publishes an article in August with links to site B, we add an edge in citation network of August. We apply all the algorithms mentioned in previous section and evaluate the inferred diffusion networks using the averaged max F1 and the area under the PR curve. We set the delay dis-

Figure 5: Precision-Recall curve of network inferences in the Twitter100 dataset. The solid squares correspond to points with maximum F1 score on the PR curve.

tribution in the CIC model to exponential distribution with $\lambda = 1$. The length of the observation window is set to 30 where each time unit corresponds to a day. The choice of larger λ compared to the Twitter dataset is due to the different time-scale normalization.

The results on inferring dynamic networks are summarized in Table 5. From the results, we can see that MCM with both network priors consistently outperform the baselines. MCM with the latent space prior achieves the best performance with more than 5% improvement on all the metrics of dynamic network inference.

Joint inference of topic-specific networks: We also construct topic-specific cascades on the MemeTracker dataset. We manually pick five major events/topics in year 2008 over a wide range of domains. The selected events are BARACK OBAMA, OLYMPIC GAME, FINANCIAL CRISIS, IRAQ WAR and MICROSOFT NEWS. We then extract all related memes as cascades of the five topics. Similar to the Twitter dataset, we extract the topic-specific diffusion networks between 250 most active sites using hyperlinks from articles that contains the corresponding topics. Due to the noise in the hyperlinks between the blog posts, for example, links to advertisement page or hyperlinks in irrelevant frames, we further exclude the hyperlinks that never appear in the collection of cascades. That is, we only include an edge (u, v) if u is at least once activated before v in a cascade. This dataset is referred to as MemeTrackerTopic.

The results on inferring topic-specific networks on MemeTrackerTopic dataset is also shown in Table 5. Compared to inferring dynamic networks, our method with LSM prior achieves much more significant improvement in the topic-specific networks inference task (with 50% improvement) in both metrics we considered. As the variations across topic-specific networks are larger than that in the dynamic networks, our MCM benefits more from incorporating the prior to mitigate the data scarcity issue. This behavior is also observed in the Twitter dataset.

6. CONCLUSION AND FUTURE WORK

In this paper, we study the problem of inferring *multiple* diffusion networks with limited cascade observations. Our

Figure 6: Inferred diffusion networks among top 10 users from the Twitter dataset. The bold black line segments at the end of edges correspond to the direction of the influence.

proposed model, MCM, naturally incorporates network priors in the generative process and provides a feasible framework to capture the shared information among different networks. Moreover, MCM can easily take advantage of recent advances in network generation models. An EM algorithm is developed to learn the parameters of the network priors, and the network inference is carried out by a MAP inference using the learned network prior. We evaluate MCM on both synthetic datasets and two real-world datasets, Twitter and Memetracker, with Stochastic Blockmodels and latent space models as examples of network priors. Experiment results show that our approach can achieve significant improvement in inference accuracy.

For future work, we are interested in extending our framework to inferring the influence strength as discussed in Section 4.3. Furthermore, the current MCM model assumes that the forms of the network priors are given as input. A natural next step is to develop a unified model that automatically decides which network priors is most appropriate for the inference problem as suggested in [26].

Acknowledgments

We would like to thank Manuel Gomez Rodriguez and Le Song for useful discussions and anonymous reviewers for useful feedback. The research was sponsored in part by NSF research grant IIS-1254206 and by the U.S. Defense Advanced Research Projects Agency (DARPA) under Social Media in Strategic Communication (SMISC) program, Agreement Number W911NF-12-1-0034. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

7. REFERENCES

- [1] E. Adar, L. Zhang, L. Adamic, and R. Lukose. Implicit structure and the dynamics of blogspace. In *Workshop on the Weblogging Ecosystem*, volume 13, 2004.
- [2] M. T. Bahadori, Y. Liu, and E. P. Xing. Fast structure learning in generalized stochastic processes with latent factors. In *Proc. 19th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 284–292, 2013.
- [3] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz. Submodular maximization with cardinality constraints. In *Proc. 25th ACM-SIAM Symp. on Discrete Algorithms*, pages 1433–1452, 2014.
- [4] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schölkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *Proc. 31st Intl. Conf. on Machine Learning*, 2014.
- [5] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. 7th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 57–66, 2001.
- [6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *Proc. VLDB Endow.*, 2(1), Aug. 2009.
- [7] N. Du, L. Song, H. Woo, and H. Zha. Uncover topic-sensitive information diffusion networks. In *Proc. 16th Intl. Conf. on Artificial Intelligence and Statistics*, 2013.
- [8] N. Du, L. Song, S. Yuan, and A. J. Smola. Learning networks of heterogeneous influence. In *Proc. 24th Advances in Neural Information Processing Systems*, pages 2780–2788, 2012.

- [9] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proc. 28th Intl. Conf. on Machine Learning*, pages 561–568, 2011.
- [10] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data*, 5(4), 2012.
- [11] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *Proc. 30th Intl. Conf. on Machine Learning*, 2013.
- [12] M. Gomez-Rodriguez and B. Schölkopf. Submodular inference of diffusion networks from multiple trees. In *Proc. 29th Intl. Conf. on Machine Learning*, 2012.
- [13] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *13th Intl. World Wide Web Conference*, pages 491–501, 2004.
- [14] X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In *Proc. 32nd Intl. Conf. on Machine Learning*, 2015.
- [15] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.
- [16] M. O. Jackson et al. *Social and economic networks*, volume 3. Princeton University Press Princeton, 2008.
- [17] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proc. 9th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 137–146, 2003.
- [18] T. Kurashima, T. Iwata, N. Takaya, and H. Sawada. Probabilistic latent network visualization: Inferring and embedding diffusion networks. In *Proc. 20th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 1236–1245, 2014.
- [19] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1), May 2007.
- [20] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proc. 15th Intl. Conf. on Knowledge Discovery and Data Mining*, 2009.
- [21] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *J. Mach. Learn. Res.*, 11:985–1042, 2010.
- [22] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *Proc. 11th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 177–187, 2005.
- [23] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *Proc. 22nd Advances in Neural Information Processing Systems*, pages 1741–1749, 2010.
- [24] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. 1998.
- [25] N. Praneeth and S. Sujay. Learning the graph of epidemic cascades. In *Proc. 12th ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems*, pages 211–222, 2012.
- [26] S. Wang, X. Hu, P. S. Yu, and Z. Li. Mmrate: Inferring multi-aspect diffusion networks with multi-pattern cascades. In *Proc. 20th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 1246–1255, 2014.
- [27] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82:8–19, 1987.
- [28] S. Wasserman. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.
- [29] S.-H. Yang and H. Zha. Mixture of mutually exciting processes for viral diffusion. In *Proc. 30th Intl. Conf. on Machine Learning*, 2013.
- [30] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank network using multi-dimensional hawkes processes. In *Proc. 30th Intl. Conf. on Machine Learning*, 2013.