Uncover Topic-Sensitive Information Diffusion Networks

Nan Du Le Song Hyenkyun Woo Hongyuan Zha
dunan@gatech.edu, {lsong, hyenkyun.woo, zha}@cc.gatech.edu
College of Computing, Georgia Institute of Technology

Abstract

Analyzing the spreading patterns of memes with respect to their topic distributions and the underlying diffusion network structures is an important task in social network analysis. This task in many cases becomes very challenging since the underlying diffusion networks are often hidden, and the topic specific transmission rates are unknown either. In this paper, we propose a continuous time model, TopicCascade, for topic-sensitive information diffusion networks, and infer the hidden diffusion networks and the topic dependent transmission rates from the observed time stamps and contents of cascades. One attractive property of the model is that its parameters can be estimated via a convex optimization which we solve with an efficient proximal gradient based block coordinate descent (BCD) algorithm. In both synthetic and real-world data, we show that our method significantly improves over the previous state-of-the-art models in terms of both recovering the hidden diffusion networks and predicting the transmission times of memes.

1 INTRODUCTION

Ideas, tweets, styles, and online advertisements spread from person to person within social networks consisting of millions of entities and edges. Analyzing the diffusion behaviors of memes with respect to network structures and their topic distributions is an important task in social network analysis. A better understanding of the temporal dynamics of diffusion networks can potentially provide us a better prediction of the occurrence time of a future event. For instance, in e-commerce, Ad-providers would like to know how to make their advertisements reach a large number of target consumers in a very short term. To achieve this goal, they have to figure out quantitatively the spreading speed of different advertisements among different communities. Outdoor equipment promotions spread faster within a community of mountaineering club than within a political group. In contrast, messages from political campaign tend to diffuse faster in the latter. Understanding the topic specific temporal dynamics will help marketers expand the influence of their ads and ideas, or reduce the negative impact of rumors and gossips.

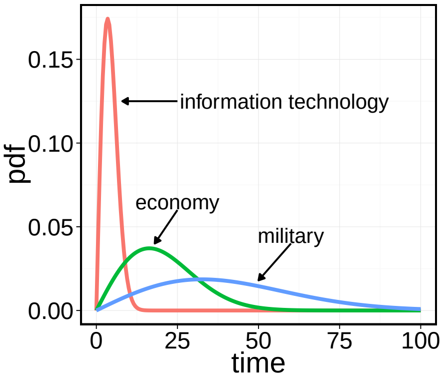


Figure 1: The fitted Rayleigh distributions of the interval between the time when a post appeared in one site and the time when a new post in another site links to it (transmission time). Posts of different topics have a different spreading rate, reflected in the different shapes of the fitted Rayleigh distribution.

However, the problem of analyzing topic dependent information diffusion can be complicated by the fact that the diffusion network is often hidden, and the topic specific transmission rates are also unknown. For instance, when consumers rush to buy some particular products, marketers can know when purchases are made, but they cannot track where the recommendations originated. When an online post mentions a piece of news, the blogger may carelessly or intentionally miss the links to the sources. In all such cases, we observe only the temporal information together with the possible content when a piece of information has been received by a particular entity, but the exact path of transmission is not observed. The complexity of topic specific transmission patterns is well illustrated by our earlier

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31. Copyright 2013 by the authors.

examples of outdoor equipment promotions and political campaign messages. Another more quantitative example can be found in Figure 1, where we examine a pair of media sites from the MemeTracker dataset [4, 9] and plot the fitted Rayleigh models to the transmission time of three types of posts corresponding to the topic of information technology, economy, and military, respectively. It clearly shows the existence of three different modes, indicating that posts on information technology spread much faster than those of economy and military. As a consequence, this phenomenon leads to a multimodal distribution of times, which cannot be easily captured by existing models, not to mention the incorporation of contents into a diffusion model.

In this paper, we propose a probabilistic model, referred to as TopicCascade, to capture the diffusion of memes with different topics through an underlying network. The key idea is to explicitly model the transmission times as continuous random variables and modulate the transmission likelihood by the topic distribution of each meme. The sparsity pattern of the model parameters provides us the structure of the diffusion network, while the parameters capture the degree of modulation of topics on the information transmission rate. Moreover, based on a set of sample cascades, we have designed an efficient proximal gradient based block coordinate decent algorithm to recover the diffusion network and model parameters. We applied TopicCascade to both synthetic and real world data, and it better recovers the underlying diffusion networks and significantly improves the accuracy of predicting the transmission time between networked entities.

2 RELATED WORK

A number of studies in the literature attempted to learn causal structures from multivariate time series (*e.g.*, [3, 6, 12]). However, these models treated time as discrete steps rather than a continuous random variable.

Recently, researchers started modeling information diffusion using continuous time models. For instance, Meyers and Leskovec [13] proposed a model called CONNIE which inferred the diffusion network by learning the pairwise infection probability using convex programming. The optimal diffusion network topology is then inferred from the infection probability matrix. Gomez-Rodriguez et al. [5] proposed a model called NETINF which used submodular optimization to find the optimal network connectivity by constructing candidate subgraphs. However, both CONNIE and NETINF assumed that the transmission rate is fixed across the network as a predefined constant.

Subsequently, Gomez-Rodriguez et al. [4] proposed an elegant model called NETRATE using continuous time model which allows variable diffusion rates across different edges. NETRATE achieved better modelings in various aspects compared to previous two approaches. However, NETRATE ignored the contents of memes and assumed that all memes are transmitted with the same rate, which can deviate far from reality as illustrated in Figure 1. Wang et al. [17] proposed a model called MoNET which considered additional features of nodes in addition to time stamps. The major difference of our model from MoNET is that we explicitly model and learn the influence of meme contents on information diffusion, while MoNET used a predefined similarity measure between meme contents.

3 PRELIMINARY

In this section, we present basic concepts from survival analysis [7, 8], which are essential for our later modeling. We first define a nonnegative random variable T to be the time when an event happens. Let $f(t)$ be the probability density function of T, and $F(t) = \Pr(T \leq t) = \int_0^t f(x)dx$ is thus its cumulative distribution function. The survival function $S(t)$ gives the probability that an event does not happen up to time t,

$$S(t) = \Pr(T \geq t) = 1 - F(t) = \int_t^\infty f(x)\,dx. \quad (1)$$

Thus, $S(t)$ is a continuous and monotonically decreasing function with $S(0) = 1$ and $S(\infty) = \lim_{t\to\infty} S(t) = 0$.

Given $f(t)$ and $S(t)$, the hazard function $h(t)$ is the instantaneous rate that an event will happen within a small interval just after time t given it has not happened yet up to time t,

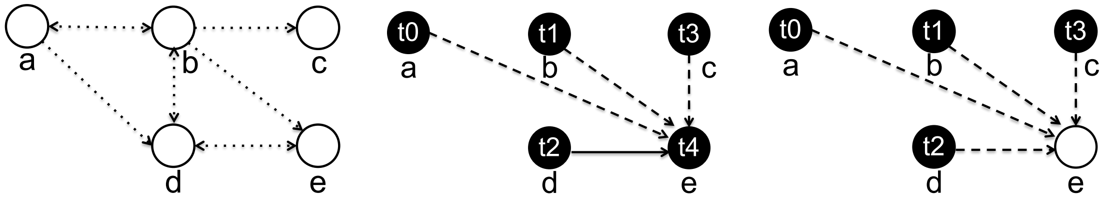
$$h(t) = \lim_{\Delta t \to 0} \frac{\Pr(t \leq T \leq t + \Delta t | T \geq t)}{\Delta t} = \frac{f(t)}{S(t)}. \quad (2)$$

The hazard function $h(t)$ is related to the survival function $S(t)$ via the differential equation $h(t) = -\frac{d}{dt}\log S(t)$, where we have used $f(t) = -S'(t)$. Solving the differential equation with boundary condition $S(0) = 1$, we can represent $S(t)$ and $f(t)$ solely from the hazard function $h(t)$, *i.e.*,

$$S(t) = \exp\left(-\int_0^t h(x)\,dx\right), \text{ and } f(t) = h(t)S(t). \quad (3)$$

4 MODELING CASCADES BY SURVIVAL ANALYSIS

We apply the survival analysis methods to model the information diffusion processes by following the work of Gomez-Rodriguez et al. [4]. We assume that within a directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes, memes transfer directly between neighboring nodes and will reach the far away nodes only through a diffusion process. Because the true underlying network is unknown, our observations include only the temporal information when events occur and the content information of memes. The temporal information is then organized as cascades, each of which corresponds to a diffusion of a particular event. For instance, an



(a) Hidden network (b) Node e gets infected at time t_4 (c) Node e survives

Figure 2: Cascades over a hidden network. Dot lines in panel (a) are connections in a hidden network. In panel (b) and (c), events occur to a, b, c and d at $t_0 < t_1 < t_2 < t_3$, respectively. In panel (b), node e was influenced by node d and survived from a, b, and c. In panel (c), node e survived from the influence of all its parents.

event could be a tweet saying that "Curiosity Rover successfully lands on the Mars". This tweet spreads over the social network by triggering a sequence of tweets and retweets that later refer to it. During the diffusion process, each tweet will have a time stamp recording when it is created. Therefore, a **cascade** can be denoted as an N-dimensional vector $\boldsymbol{t}^c := (t_1^c, \ldots, t_N^c)^\top$ with i-th dimension recording the time stamp when event c occurs to node i, and $t_i^c \in [0, T^c] \cup \{\infty\}$. The symbol ∞ labels nodes that have not been reached in a cascade during an observation window $[0, T^c]$. The 'clock' is set to 0 at the start of each cascade. A dataset usually consists of a collection, $\mathcal{C}$, of cascades $\{\boldsymbol{t}^1, \ldots, \boldsymbol{t}^{|\mathcal{C}|}\}$. Within a cascade c, if $t_j^c < t_i^c$, we say node j is the parent of node i. Note that this *parents and children within a cascade* [4] relation is different from the parent-child structural relation on the true underlying diffusion network. In Figure 2(a), only b and d are the parents of e in topology. Yet, in Figure 2(b), all the nodes from a to d are the parental nodes of e in the given cascade.

Moreover, each directed edge, $j \to i$, is associated with a **transmission function** $f_{ji}(t_i|t_j)$, which is the conditional likelihood of an event happening to node i at time t_i given that the same event has already happened to node j at time t_j. It captures the temporal pattern between two successive events from node j to i. Here, we focus on shift-invariant transmission functions whose value only depends on the time difference, *i.e.*, $f_{ji}(t_i|t_j) = f_{ji}(t_i - t_j) = f_{ji}(\Delta_{ji})$ where $\Delta_{ji} := t_i - t_j$. If there is no edge from j to i, we will have $f_{ji}(\Delta_{ji}) = 0$ and $h_{ji}(\Delta_{ji}) = 0$. Therefore, the structure of the diffusion network is reflected in the non-zero patterns of a collection of transmission functions (or hazard functions). The likelihood $\ell(\boldsymbol{t}^c)$ of a cascade induced by event c is then a product of all individual likelihood $\ell_i(\boldsymbol{t}^c)$ whether event c occurs to each node i or not.

If **event** c **occurs to node** i, we assume that it happens only under the influence of the first parent and will not happen again. In Figure 2(b), node e is susceptible to its parent a, b, c and d. Yet, only node d is the first parent who actually influences node e, in other words, node e first gets the meme from node d and survives from the influence of all the other nodes. Because each parent is equally likely to be the first parent, the likelihood is derived as

$$\begin{aligned}\ell_i^+(\boldsymbol{t}^c) &= \sum_{j:t_j^c<t_i^c} f_{ji}(\Delta_{ji}^c) \prod_{k:k\neq j, t_k^c<t_i^c} S_{ki}(\Delta_{ki}^c) \\ &= \sum_{j:t_j^c<t_i^c} h_{ji}(\Delta_{ji}^c) \prod_{k:t_k^c<t_i^c} S_{ki}(\Delta_{ki}^c), \qquad (4)\end{aligned}$$

where we have used relation $f_{ji}(\cdot) = h_{ji}(\cdot) S_{ji}(\cdot)$.

If **event** c **does not occur to node** i, node i survives from the influence of all parents (see Figure 2(c) for illustration). The likelihood is a product of survival functions, *i.e.*,

$$\ell_i^-(\boldsymbol{t}^c) = \prod_{t_j \leq T^c} S_{ji}(T^c - t_j). \qquad (5)$$

Combining the above two scenarios together, we can obtain the overall likelihood of a cascade $\boldsymbol{t}^c$, *i.e.*,

$$\ell(\boldsymbol{t}^c) = \prod_{t_i^c > T^c} \ell_i^-(\boldsymbol{t}^c) \times \prod_{t_i^c \leq T^c} \ell_i^+(\boldsymbol{t}^c). \qquad (6)$$

Thus, **the likelihood of all cascades** is a product of these individual cascade likelihoods, *i.e.* $\ell(\{\boldsymbol{t}^1, \ldots, \boldsymbol{t}^{|\mathcal{C}|}\}) = \prod_{c=1,\ldots,|\mathcal{C}|} \ell(\boldsymbol{t}^c)$. In the end, we take the negative log of this likelihood function and regroup all terms associated with edges pointing to node i together to derive

$$\begin{aligned}\mathcal{L}(\{\boldsymbol{t}^1, \ldots, \boldsymbol{t}^{|\mathcal{C}|}\}) = &-\sum_i \sum_j \sum_{\{c|t_j^c<t_i^c\}} \log S_{ji}(\Delta_{ji}^c) \\ &-\sum_i \sum_{\{c|t_i^c<T^c\}} \log \sum_{\{t_j^c<t_i^c\}} h_{ji}(\Delta_{ji}^c) \quad (7)\end{aligned}$$

There are two important implications from this negative log likelihood function. First, the function can be expressed using only the hazard and the survival function. Second, both the hazard function and the survival function are fixed between the same pair of nodes $j \to i$ regardless of the content features of the memes sent from node j.

5 TOPIC-SENSITIVE DIFFUSION MODEL

In this section, we will present our diffusion model which takes into account the topic dependent transmission rates of memes. The basic idea is to modulate the hazard function of a Rayleigh distribution by the topic distributions of memes. Then we will infer the parameters by maximizing the likelihood of observed cascades with grouped lasso type of regularization. Furthermore, we will design an efficient optimization algorithm using a proximal gradient based block coordinate descent method.

5.1 Topic Modulated Rayleigh Distribution

The Rayleigh distribution is often used in epidemiology and survival analysis (*e.g.*, [7, 8, 16]). The corresponding probability density function first increases rapidly and then decays to zero. As a consequence, it is well-suited to capture the phenomenon that there is a typical response time for posts of a particular topic, and it is less likely the response time is very different from the mode. Given a pair of nodes j and i, the density, hazard and survival functions of the Rayleigh distribution are

$$f_{ji}(\Delta_{ji}) = \alpha_{ji} \cdot \Delta_{ji} \cdot \exp\left(-\frac{1}{2} \cdot \alpha_{ji} \cdot \Delta_{ji}^2\right), \quad (8)$$

$$h_{ji}(\Delta_{ji}) = \alpha_{ji} \cdot \Delta_{ji}, \quad (9)$$

$$S_{ji}(\Delta_{ji}) = \exp\left(-\frac{1}{2} \cdot \alpha_{ji} \cdot \Delta_{ji}^2\right), \quad (10)$$

where $\Delta_{ji} = t_j - t_i > 0$, and $\alpha_{ji} \in \mathbb{R}^+$ is the transmission rate from node j to i. We note that this standard Rayleigh distribution does not take into account the potential influence of topics on the transmission rate. We want to take that into account and allow α_{ji} to be modulated by the topics of a meme.

More specifically, suppose that node j just published a meme $\mathbf{m}_j^{t_j}$ at time t_j. We represent each meme as a topic vector in the canonical K-dimensional simplex, in which each component is the weight of a topic. That is $\mathbf{m}_j^{t_j} := (m_1, \ldots, m_K)^\top$ and $\sum_i m_i = 1$ and $m_i \in [0, 1]$. Such a representation can be readily obtained from the text of a meme using standard topic model tools such as Latent Dirichlet Allocation [2]. To incorporate the topic information of the topic, $\mathbf{m}_j^{t_j}$, we assume that α_{ji} is a nonnegative combination of the entries in $\mathbf{m}_j^{t_j}$, *i.e.*

$$\alpha_{ji} = \sum_{l=1}^{K} \alpha_{ji}^l m_l, \quad (11)$$

where $\alpha_{ji}^l \geqslant 0$ ensures that the hazard function $h_{ji}^m(\Delta_{ji}) = \Delta_{ji} \sum_{l=1}^K \alpha_{ji}^l m_l$ is nonnegative. For notation simplicity, let the vector $\boldsymbol{\alpha}_{ji} := (\alpha_{ji}^1, \ldots, \alpha_{ji}^K)^\top$, and we define the topic modulated hazard function as

$$h_{ji}^m(\Delta_{ji}) = \Delta_{ji} \boldsymbol{\alpha}_{ji}^\top \mathbf{m}_j^{t_j} \quad (12)$$

For the modulated hazard function $h_{ji}^m(\Delta_{ji})$, the coefficients vector $\boldsymbol{\alpha}_{ji}$ can be interpreted as the topic preference of the diffusion channel from node j to i. For each meme $\mathbf{m}_j^{t_j}$ published by node j, $h_{ji}^m(\Delta_{ji})$ uniquely determines its transmission pattern. If $\mathbf{m}_j^{t_j}$ is compatible with the topic preference encoded in $\boldsymbol{\alpha}_{ji}$, then their inner product will be large, and this meme will have a high transmission rate; otherwise, the product will be small, and the transmission will be slow and even impossible.

The flexibility of the topic modulated hazard function in (12) captures our previous example of the outdoor equipment promotions. Within a mountaineering club, the general preference vector $\boldsymbol{\alpha}_{ji}$ mainly concentrates on the topics related to sport activities. The $\mathbf{m}_j^{t_j}$ of the outdoor equipment promotions will have a larger inner product with $\boldsymbol{\alpha}_{ji}$ than that of a political campaign message, so the related sport ad will spread faster along this edge. Based on their respective relations with the hazard function in (3), the topic modulated survival and density function become

$$S_{ji}^m(\Delta_{ji}) = \exp\left(-\frac{1}{2} \boldsymbol{\alpha}_{ji}^\top \mathbf{m}_j^{t_j} \Delta_{ji}^2\right), \quad (13)$$

$$f_{ji}^m(\Delta_{ji}) = \boldsymbol{\alpha}_{ji}^\top \mathbf{m}_j^{t_j} \Delta_{ji} \exp\left(-\frac{1}{2} \boldsymbol{\alpha}_{ji}^\top \mathbf{m}_j^{t_j} \Delta_{ji}^2\right). \quad (14)$$

5.2 Parameter Estimation

Next we plug in the modulated hazard function (12) and survival function (13) back into the negative log-likelihood (7). Because the negative log likelihood is separable for each node i, we can optimize the set of variables $\{\boldsymbol{\alpha}_{ji}\}_{j=1}^N$ separately. As a result, the negative log likelihood for the data associated with node i can be estimated as

$$\mathcal{L}_i\left(\{\boldsymbol{\alpha}_{ji}\}_{j=1}^N\right) = \sum_j \sum_{\{c|t_j^c < t_i^c\}} \frac{(\Delta_{ji}^c)^2}{2} \cdot \boldsymbol{\alpha}_{ji}^\top \cdot \mathbf{m}_j^{t_j^c} - \sum_{\{c|t_i^c < T^c\}} \log \sum_{\{t_k^c < t_i^c\}} \Delta_{ki}^c \cdot \boldsymbol{\alpha}_{ki}^\top \cdot \mathbf{m}_k^{t_k^c}. \quad (15)$$

A desirable feature of this negative log-likelihood function is that it is convex in the arguments, $\{\boldsymbol{\alpha}_{ji}\}_{j=1}^N$. The convexity will allow us to find the global minimum solution efficiently using various convex optimization tools. We also assume that the terms with $\log(\cdot)$ is larger than $\log(\epsilon)$ with small constant $\epsilon > 0$. It means that each node has at least one parent in the diffusion network.[1]

Moreover, we want to induce a sparse network structure from the data and avoid overfitting. If the coefficients $\boldsymbol{\alpha}_{ji} = \mathbf{0}$, then there is no edge (or direct transmission) from node j to i. For this purpose, we will impose grouped lasso type of regularization on the coefficients $\boldsymbol{\alpha}_{ji}$, *i.e.*, $(\sum_j \|\boldsymbol{\alpha}_{ji}\|_2)$ [14, 18, 19]. Grouped lasso type of regularization has the tendency to select a small number of salient groups of non-zero coefficients and push other groups of potentially noisy coefficients to zero. Then we

[1] To handle the case of isolated nodes, we can introduce a base hazard rate b into equation (12).

have the following optimization problem

$$\min_{\{\boldsymbol{\alpha}_{ji}\}_{j=1}^{N}} \; \mathcal{L}_i\left(\{\boldsymbol{\alpha}_{ji}\}_{j=1}^{N}\right) + \lambda\Big(\sum\nolimits_j \|\boldsymbol{\alpha}_{ji}\|_2\Big) \qquad (16)$$
$$\text{s.t. } \boldsymbol{\alpha}_{ji} \geq \mathbf{0}, \quad \forall j \in \{1, \ldots, N\}$$

where λ is a regularization parameter which trades off between the data likelihood and the group sparsity of the coefficients. After we obtain a sparse solution from the above optimization, we obtain the model parameters $\boldsymbol{\alpha}_{ji}$ and partial network structures, each of which centers around a particular node i. We can then join all the partial structures together and obtain the overall diffusion network.

5.3 Optimization

We note that the optimization problem in (16) is a nonsmooth but separable minimization problem, which can be difficult to optimize using standard methods. We will use the following block coordinate descent framework [15] to efficiently find a solution. At each iteration, we solve (16) with respect to one $\boldsymbol{\alpha}_{ji}$ variable while fixing other variables, *i.e.*,

$$\left\{\begin{array}{lcl} \multicolumn{3}{l}{\text{For } k = 1, \ldots, k_{\max}} \\ \quad \boldsymbol{\alpha}_{1i}^{k+1} & = & \underset{\boldsymbol{v}_1 \geq \mathbf{0}}{\operatorname{argmin}} \; \mathcal{L}_{1i}(\boldsymbol{v}_1) + \lambda \|\boldsymbol{v}_1\|_2 \\ \quad \boldsymbol{\alpha}_{2i}^{k+1} & = & \underset{\boldsymbol{v}_2 \geq \mathbf{0}}{\operatorname{argmin}} \; \mathcal{L}_{2i}(\boldsymbol{v}_2) + \lambda \|\boldsymbol{v}_2\|_2 \\ \quad \cdots & & \\ \quad \boldsymbol{\alpha}_{Ni}^{k+1} & = & \underset{\boldsymbol{v}_N \geq \mathbf{0}}{\operatorname{argmin}} \; \mathcal{L}_{Ni}(\boldsymbol{v}_N) + \lambda \|\boldsymbol{v}_N\|_2 \end{array}\right. \qquad (17)$$

where the function $\mathcal{L}_{ji}(\boldsymbol{v}_j)$ is defined by fixing all other coordinates to their current values in the iteration, *i.e.*,

$$\mathcal{L}_{ji}(\boldsymbol{v}_j) := \mathcal{L}_i\left(\boldsymbol{\alpha}_{1i}^{k+1}, ..., \boldsymbol{\alpha}_{(j-1)i}^{k+1}, \boldsymbol{v}_j, \boldsymbol{\alpha}_{(j+1)i}^{k}, ..., \boldsymbol{\alpha}_{Ni}^{k}\right).$$

Essentially, the optimization is carried out by solving a sequence of subproblems each involving only one $\boldsymbol{\alpha}_{ji}$. Since $\mathcal{L}_i\left(\{\boldsymbol{\alpha}_{ji}\}_{j=1}^{N}\right)$ is sufficiently smooth convex fitting terms and $\sum_j \|\boldsymbol{\alpha}_{ji}\|_2$ is non-differentiable but separable regularization terms, the global convergence of the above algorithm is guaranteed by results from [15].

However, it can still be difficult to find a solution for each subproblem directly due to the non-smooth regularizer. Thus we will use a proximal gradient method for these subproblems (*e.g.*, [1]). First, the gradient of $\mathcal{L}_{ji}$ with respect to $\boldsymbol{v}_j$ can be readily calculated as

$$\nabla\mathcal{L}_{ji}(\boldsymbol{v}_j) := \frac{\partial\mathcal{L}_{ji}(\boldsymbol{v}_j)}{\partial\boldsymbol{v}_j} = \sum_{\{c|t_j^c<t_i^c\}} \frac{(\Delta_{ji}^c)^2}{2} \cdot \mathbf{m}_j^{t_j^c} - \sum_{\{c|t_i^c<T^c\}} \frac{\Delta_{ji}^c \cdot \mathbf{m}_j^{t_j^c}}{A_j^c + \Delta_{ji}^c \cdot \boldsymbol{v}_j^\top \; \boldsymbol{m}_j^{t_j^c}} \qquad (18)$$

where the term A_j^c is defined as

$$A_j^c = \sum_{l<j, t_l^c<t_i^c} \Delta_{li}^c \cdot (\boldsymbol{\alpha}_{li}^{k+1})^\top \boldsymbol{m}_l^{t_l^c} + \sum_{l>j, t_l^c<t_i^c} \Delta_{li}^c \cdot (\boldsymbol{\alpha}_{li}^{k})^\top \boldsymbol{m}_l^{t_l^c}.$$

Second, to use the proximal gradient method, the gradient with respective to the component $\boldsymbol{v}_j$ has to be Lipschitz continuous satisfying the condition

$$\|\nabla\mathcal{L}_{ji}(\boldsymbol{v}_j) - \nabla\mathcal{L}_{ji}(\boldsymbol{v}_j')\|_2 \leqslant L\|\boldsymbol{v}_j - \boldsymbol{v}_j'\|_2 \qquad (19)$$

for some constant $L < \infty$. This condition can be readily satisfied by our model, since the Hessian of $\mathcal{L}_{ji}(\boldsymbol{v}_j)$ is bounded, *i.e.*,

$$\frac{\partial^2\mathcal{L}_{ji}(\boldsymbol{v}_j)}{\partial\boldsymbol{v}_j^2} = \sum_{\{c|t_i^c<T^c\}} \frac{\boldsymbol{m}_j^{t_j^c} \cdot (\boldsymbol{m}_j^{t_j^c})^\top \cdot (\Delta_{ji}^c)^2}{\left(A_j^c + \Delta_{ji}^c \cdot \boldsymbol{v}_j^\top \cdot \boldsymbol{m}_j^{t_j^c}\right)^2} \leq L.$$

We will use iterative procedure to solve each subproblem

$$\boldsymbol{\alpha}_{ji}^{k+1} = \underset{\boldsymbol{v}_j \geq \mathbf{0}}{\operatorname{argmin}} \; \mathcal{L}_{ji}(\boldsymbol{v}_j) + \lambda \|\boldsymbol{v}_j\|_2,$$

and more specifically, we will minimize a sequence of quadratic approximation to $\mathcal{L}_{ji}(\boldsymbol{v}_j)$

$$\mathcal{L}_{ji}^s(\boldsymbol{v}_j, \boldsymbol{v}_j') = \mathcal{L}_{ji}(\boldsymbol{v}_j') + \left\langle \nabla\mathcal{L}_{ji}(\boldsymbol{v}_j'), \boldsymbol{v}_j - \boldsymbol{v}_j' \right\rangle + \frac{1}{2}\left\langle \boldsymbol{v}_j - \boldsymbol{v}_j', \boldsymbol{D}_{\boldsymbol{v}_j'}(\boldsymbol{v}_j - \boldsymbol{v}_j') \right\rangle,$$

where $\boldsymbol{D}_{\boldsymbol{v}_j'}$ is a positive definite matrix that dominates the Hessian $\frac{\partial^2\mathcal{L}_{ji}(\boldsymbol{v}_j')}{\partial(\boldsymbol{v}_j')^2}$. This surrogate function has the following useful properties: $\mathcal{L}_{ji}^s(\boldsymbol{v}_j, \boldsymbol{v}_j) = \mathcal{L}_{ji}(\boldsymbol{v}_j)$ and $\mathcal{L}_{ji}^s(\boldsymbol{v}_j, \boldsymbol{v}_j') \geqslant \mathcal{L}_{ji}(\boldsymbol{v}_j)$, $\forall \boldsymbol{v}, \boldsymbol{v}_j' \geq \mathbf{0}$. As a consequence, we find $\boldsymbol{\alpha}_{ji}^{k+1}$ by the following iterative procedure

$$\left\{\begin{array}{lcl} \multicolumn{3}{l}{\boldsymbol{v}_j' \leftarrow \boldsymbol{\alpha}_{ji}^{k}} \\ \multicolumn{3}{l}{\text{For } l = 1, \ldots, l_{\max}} \\ \quad \boldsymbol{v}_j' & \leftarrow & \underset{\boldsymbol{v}_j \geq 0}{\operatorname{argmin}} \; \mathcal{L}_{ji}^s(\boldsymbol{v}_j, \boldsymbol{v}_j') + \lambda\|\boldsymbol{v}_j\|_2 \\ \multicolumn{3}{l}{\boldsymbol{\alpha}_{ji}^{k+1} \leftarrow \boldsymbol{v}_j'} \end{array}\right. \qquad (20)$$

If $\boldsymbol{D}_{\boldsymbol{\alpha}_{ji}^k} = L \cdot \boldsymbol{I}$, then we only need to solve the following proximal mapping

$$\boldsymbol{v}_j' \;\leftarrow\; \underset{\boldsymbol{v}_j \geq 0}{\operatorname{argmin}} \; \frac{L}{2}\left\|\boldsymbol{v}_j - \left(\boldsymbol{v}_j' - \frac{\nabla\mathcal{L}_{ji}(\boldsymbol{v}_j')}{L}\right)\right\|_2^2 + \lambda\|\boldsymbol{v}_j\|_2,$$

which has a closed form solution

$$\boldsymbol{v}_j' \;\leftarrow\; \frac{(\boldsymbol{v})_+}{\|\boldsymbol{v}\|_2}\left(\|\boldsymbol{v}\|_2 - \frac{\lambda}{L}\right)_+, \qquad (21)$$

where $\boldsymbol{v} = \boldsymbol{v}_j' - \frac{1}{L}\nabla\mathcal{L}_{ji}(\boldsymbol{v}_j')$ and $(\cdot)_+$ simply sets the negative coordinates of its argument to 'zero'. (see [20] and reference therein for more details). If $\boldsymbol{v}_j' = \mathbf{0}$ at some point in the iteration, we just stop updating it and directly assign zeros to $\boldsymbol{\alpha}_{ji}^{k+1}$.

The overall pseudo codes are given in Algorithm 1. Note that to guarantee convergence, we need to repeat (21) several times. But empirically we did not observe significant difference if we just set $l_{\max}$ to 1. Furthermore, because the optimization is independent for each node i, the outer loop process can be easily parallelized into N separate subproblems. Meanwhile, we can prune the possible nodes that never appeared before node i in any cascade indicating that j is not a possible parent of i, which is at least shown from the data. In the end, if we further assume that all the edges from the same node have similar topic preference, especially when the sample size is small, the N edges could share a common set of K parameters, and thus we can only estimate $N \times K$ parameters in total.

Algorithm 1: TOPICCASCADE

```
Input: cascades $\{\boldsymbol{t}^1, ..., \boldsymbol{t}^{|\mathcal{C}|}\}$, memes $\{\boldsymbol{m}_j^{t_j^c}\}_{j=1...N}^{c=1...|\mathcal{C}|}$
Output: $\{\boldsymbol{\alpha}_{ji}\}_{j,i\in\{1,...,N\}}$
for $i = 1, \dots, N$ do
    Initialize $\{\boldsymbol{\alpha}_{ji}^1\}_{j=1}^N$ as uniform vectors;
    for $k = 1, \dots, k_{\max}$ do
        for $j = 1, \dots, N$ do
            $\boldsymbol{v} = \boldsymbol{\alpha}_{ji}^k - \frac{1}{L}\nabla\mathcal{L}_{ji}(\boldsymbol{\alpha}_{ji}^k)$;
            $\boldsymbol{\alpha}_{ji}^{k+1} = \frac{(\boldsymbol{v})_+}{\|\boldsymbol{v}\|_2}\left(\|\boldsymbol{v}\|_2 - \frac{\lambda}{L}\right)_+$;
```

6 EXPERIMENTAL RESULTS

We will evaluate TOPICCASCADE on both realistic synthetic networks and real world networks. We compare it to the state-of-the-art method NETRATE [4] and MoNET [17], and then we show that TOPICCASCADE can perform significantly better in terms of both recovering the network structures and predicting the transmission time.

6.1 Synthetic Networks

We first evaluate our method in synthetic datasets where we know the true parameters to study accuracy of the estimated parameter by TOPICCASCADE.

Network Generation. We generate synthetic networks that mimic the structural properties of real networks. These synthetic networks can then be used for simulation of information diffusion. Since the latent networks for generating cascades are known in advance, we can perform detailed comparisons between various methods. We use Kronecker generator [10] to examine three types of networks with directed edges: (i) the core-periphery structure [11] with parameters [0.9 0.5; 0.5 0.3], which mimics the information diffusion traces in real world networks, (ii) the Erdős-Rényi random networks with parameters [0.5 0.5; 0.5 0.5], being typical in physics and graph theory, and (iii) the hierarchy networks with parameters [0.9 0.1; 0.1 0.9].

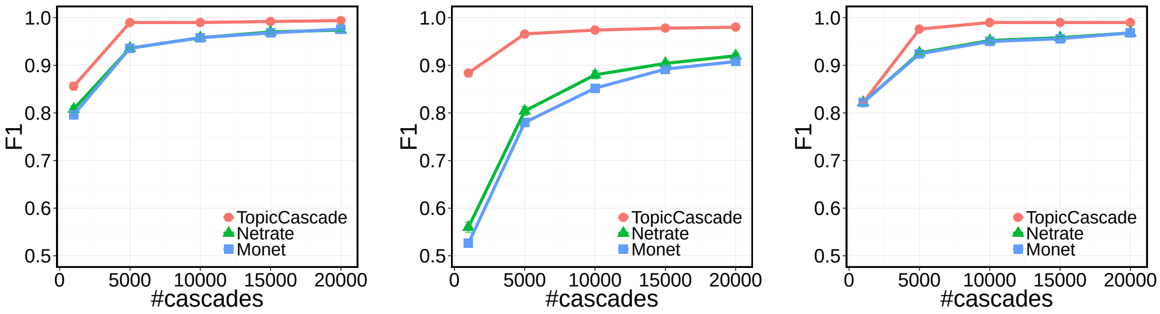
Topic Generation. Each node j is assigned a uniformly distributed random variable $\boldsymbol{\theta}_j \in (0,1]^K$ which is the parameter for a K-dimensional symmetric Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\theta}_j)$. Then we can sample a K-dimensional topic distribution $\boldsymbol{m}_j$ from $\mathrm{Dir}(\boldsymbol{\theta}_j)$ for each node j. Since the entries of $\boldsymbol{\theta}$ is less than one, the generated memes are more focused on a small subsets of topics.

Cascade Generation. For each pair of nodes j and i, the edge $j \to i$ is randomly assigned a K-dimensional topic preference vector $\boldsymbol{\alpha}_{ji}$ where each component $\alpha_{ji}^l \in [0,1]$. Given a network $\mathcal{G}$, we generate a cascade from $\mathcal{G}$ by randomly choosing a node j as the root of the cascade. The root node j is then assigned to time stamp $t_j = 0$. We sample a meme $\boldsymbol{m}_j^{t_j}$ from $\mathrm{Dir}(\theta_j)$. Then, for each neighbor node i pointed by j, its event time t_i is sampled from the formula (14). The child node i will copy the received meme $\boldsymbol{m}_j^{t_j}$ and forward it to its own children. In general, node i can add a small disturbance to the vector $\boldsymbol{m}_j^{t_j}$ to generate a slightly different meme. However, for simplicity, in our later experiments, we assume that node i directly copies $\boldsymbol{m}_j^{t_j}$ as its own meme. The diffusion process will continue by further infecting the neighbors pointed by node i in a breadth-first fashion until either the overall time exceed the predefined observation time window T^c, or there is no new node being infected. If a node is infected more than once by multiple parents, only the first infection time stamp and the meme will be recorded.

Experimental Setting. For each type of synthetic networks, we randomly instantiate the network topologies and all the required parameters ($\boldsymbol{\alpha}_{ji}$ for each edge and the set of memes $\boldsymbol{m}_j^{t_j}$) for five times. The number of cascades varies from 1000, 5000, 10000, 15000 to 20000. For each experiment setting, the regularization parameter is chosen based on two-fold cross validation, and the experimental results are reported on a separate hold-out test set consisting of the same number of cascades. For NETRATE and MoNET, we fit them with a Rayleigh transmission model.

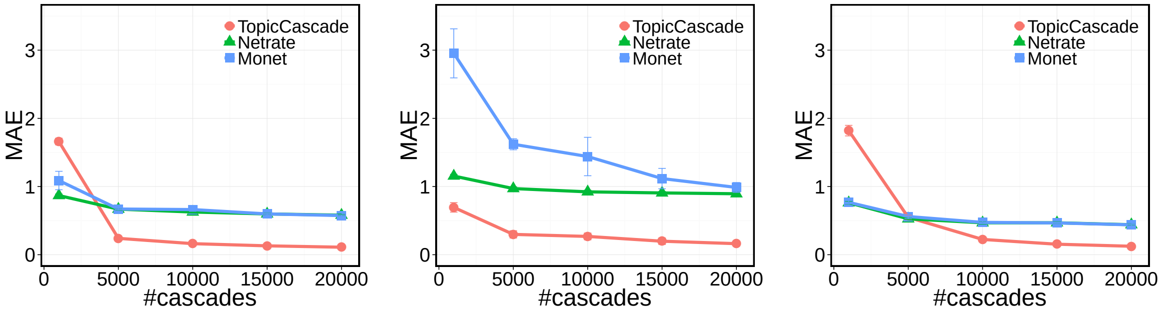
Evaluation Metrics. We have considered three different metrics: (1) we first compare the $F1$ score for the network recovery. $F1 := \frac{2 \cdot precision \cdot recall}{precision + recall}$, where $precision$ is the fraction of edges in the inferred network that also present in the true network and $recall$ is the fraction of edges in the true network that also present in the inferred network; (2) we compare the modes of the fitted Rayleigh distributions given by TOPICCASCADE, NETRATE, and MoNET with the true mode for each meme $\boldsymbol{m}_j$ from $j \to i$, and report the Mean Absolute Error (MAE). (3) and finally, we report the distance $\|\hat{\alpha_{ji}} - \boldsymbol{\alpha}_{ji}\|_2$ between the estimated parameters and the true ones as number of cascades varies.

F1 score for network recovery. From Figure 3, we can see that in all cases, TOPICCASCADE performs consistently and significantly better than NETRATE and MoNET. Furthermore, its performance also steadily increases as we increase the number of cascades, and finally TOPICCASCADE almost recovers the entire network with around 10000 cascades. In contrast, the competitor method seldom



(a) Random (b) Core-Periphery (c) Hierarchy

Figure 3: F1 Scores for network recovery. Each network has 512 nodes and 1024 edges.



(a) Random (b) Core-Periphery (c) Hierarchy

Figure 4: Absolute mean errors between the estimated modes and the true modes.

fully recovers the entire network given the same number of cascades, although it has less parameters to fit.

MAE for mode prediction. Figure 4 presents the MAE between the estimated mode and the true mode. When there are only 1000 cascades, NETRATE and MoNET can perform better than TOPICCASCADE, since TOPICCASCADE has more parameters to estimate. However, as the number of cascades grows, the MAE for TOPICCASCADE quickly decreases. In contrast, NETRATE and MoNET keep a steady error regardless of the increased number of cascades since it is not sensitive to topics of the meme.

To further illustrate the extent to which the estimated mode is close to the true value, we plot the relative error of the estimated median mode with respect to the true value in Figure 5. In all cases, the estimation given by TOPICCASCADE goes to the true mode quickly as the number of cascades increases, while NETRATE and MoNET keep an almost steady error rate.

Distance between the estimation and the true value. Because in our simulations we have known in advance the true topic preference parameter on each edge, we would like to check how close the estimated $\hat{\boldsymbol{\alpha}}_{ji}$ is close to the true $\boldsymbol{\alpha}_{ji}$. In Figure 6, we plot the average distance between $\hat{\boldsymbol{\alpha}}_{ji}$ and $\boldsymbol{\alpha}_{ji}$. Again, it shows that $\hat{\boldsymbol{\alpha}}_{ji}$ approaches to $\boldsymbol{\alpha}_{ji}$ quickly as the number of cascades increases.

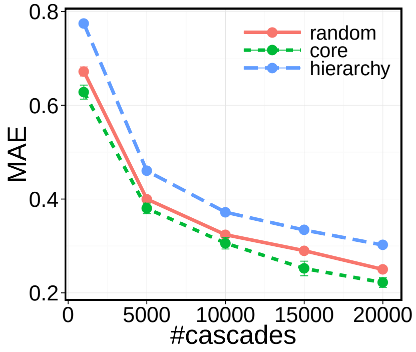
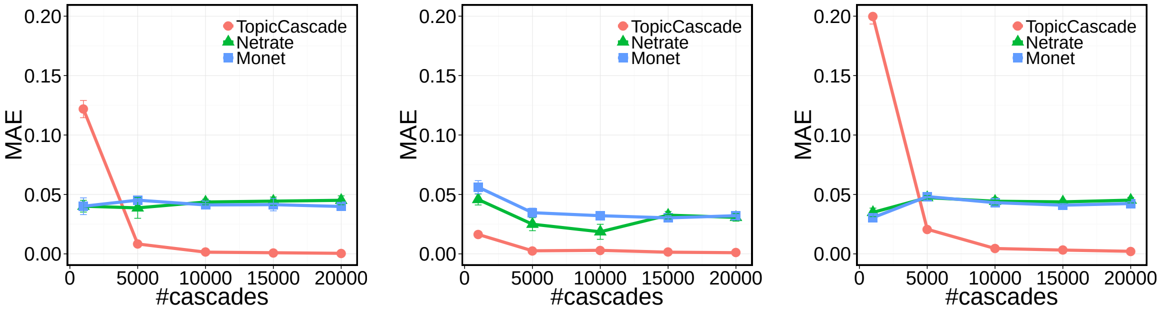


Figure 6: Average distance between the estimated parameters and the true values.

6.2 Real World Dataset

Finally, we use the MemeTracker dataset [4] to evaluate our model. In this dataset, the hyperlinks between articles and posts can be used to represent the flow of information from one site to another site. When a site publishes a new post, it will put hyperlinks to related posts in some other sites published earlier as its sources. Later as it also becomes "older", it will be cited by other newer posts as well. As a consequence, all the time-stamped hyperlinks form a cascade for particular piece of information (or event) flowing among different sites. The networks formed by these hyperlinks are used to be the ground truth. In addition, each post also has some texts to describe its content. There are



(a) Random (b) Core-Periphery (c) Hierarchy

Figure 5: Relative errors between the estimated median mode and the true median mode.

Table 1: Ten topics learnt from the posts.

Topic 0 : information	social	computer	google
Topic 1 : people	women	life	god
Topic 2 : time	love	people	life
Topic 3 : market	year	money	economy
Topic 4 : war	georgia	military	country
Topic 5 : obama	mccain	president	campaign
Topic 6 : energy	oil	gas	plan
Topic 7 : people	laws	issues	free
Topic 8 : loss	men	till	las
Topic 9 : windows	web	technology	games

totally 3771 posts with 85671 words. We have extracted a network consisting of top 500 sites with about 1100 edges and 2000 cascades. The maximum time is 614 hours. We first fit an LDA [2] model to the corpus of posts and extract ten typical topics shown in Table 1 where each topic includes the top four descriptive words.

Table 2: Estimations in the MemeTracker dataset with Rayleigh transmission functions on the correctly predicted edges.

Models	train			test	
	F1	MAE	Median	MAE	Median
TopicCascade	0.60	15.8	7.4	28.0	2.4
NETRATE	0.23	31.8	21.2	33.7	24.9
MoNET	0.14	32.3	20.8	35.2	25.8

Table 3: Estimations in the MemeTracker dataset with exponential transmission functions on the correctly predicted edges.

Models	train			test	
	F1	MAE	Median	MAE	Median
TopicCascade	0.72	174	14.9	241	9.3
NETRATE	0.81	1307	532.8	538.5	27.1
MoNET	0.72	1317	527	585	27

For each pair of nodes i and j, given a post topic $\boldsymbol{m}_j$, we use the modes of the fitted Rayleigh distributions as the estimated transmission time from j to i for the post $\boldsymbol{m}_j$. We then report the MAE and the median error between the estimated time and the real time on the correctly estimated edges in Table 2. We start to compare different models first by using the whole dataset to train and report the different metrics in the *train* column of Table 2. Then, we uniformly select 10-percent of the edges as the test data and use the other 90-percent data to train. The experiments have been repeated for 10 times. We report the average value of all the metrics in the *test* column of Table 2. Moreover, we also repeat all the above experiments by fitting a topic-modulated exponential transmission function for each edge and use the expectation as the estimated time on each correctly predicted edge in Table 3. In all cases, TopicCascade have lower MAE, Median error as well as a relatively better F1 score. For both NETRATE and MoNET, they are also fitted by the exponential distribution to the MemeTracker dataset in order to recover the network structure. Because the exponential model is relatively easier to fit than the Rayleigh distribution, it gives a better F1 score. However, when we use the expectation of the exponential to predict the time, Table 3 shows that it has large MAE and Median error in terms of predicting the actual time of the events.

7 CONCLUSIONS

We have developed a topic-sensitive diffusion model, referred to as TopicCascade to model the variable diffusion rates of memes with different topics. In contrast to the previous state-of-the-art methods, such as NETRATE and MoNET, our model adapts to the topics of each meme to capture their differential diffusion patterns. We designed an efficient algorithm to find a sparse solution using a proximal gradient based block coordinate descent method with group-lasso type of regularization. Experimental results on both synthetic and real data show that TopicCascade have better performance in terms of both the network structure recovery and predicting the transmission time.

Acknowledgment Our work is supported by NSF IIS-1218749, IIS-1116886 and the Darpa Xdata grant.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, Mar. 2009.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[3] N. Eagle, A. S. Pentland, and D. Lazer. From the cover: Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, Sept. 2009.

[4] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, pages 561–568, 2011.

[5] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD*, pages 1019–1028, 2010.

[6] M. Kolar, L. Song, A. Ahmed, and E. Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123, 2010.

[7] J. F. Lawless. *Statistical Models and Methods for Lifetime Data*. Wiley-Interscience, 2002.

[8] E. T. Lee and J. Wang. *Statistical Methods for Survival Data Analysis*. Wiley-Interscience, Apr. 2003.

[9] J. Leskovec, L. Backstrom, and J. M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, pages 497–506, 2009.

[10] J. Leskovec, D. Chakrabarti, J. M. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11:985–1042, 2010.

[11] J. Leskovec, K. J. Lang, and M. W. Mahoney. Empirical comparison of algorithms for network community detection. In *WWW*, pages 631–640, 2010.

[12] A. C. Lozano and V. Sindhwani. Block variable selection in multivariate regression and high-dimensional causal inference. In *NIPS*, pages 1486–1494, 2010.

[13] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *NIPS*, pages 1741–1749, 2010.

[14] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet, et al. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[15] P. Tseng and C. O. L. Mangasarian. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim Theory Appl*, pages 475–494, 2001.

[16] J. Wallinga and P. Teunis. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, 160(6):509–516, 2004.

[17] L. Wang, S. Ermon, and J. E. Hopcroft. Feature-enhanced probabilistic models for diffusion network inference. In *ECML/PKDD (2)*, pages 499–514, 2012.

[18] Z. Xu, R. Jin, H. Yang, I. King, and M. R. Lyu. Simple and efficient multiple kernel learning by group lasso. In *ICML*, pages 1175–1182, 2010.

[19] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, Feb. 2006.

[20] S. Yun and H. Woo. Linearized proximal alternating minimization algorithm for motion deblurring by nonlocal regularization. *Pattern Recognition*, 44:1312–1326, 2011.