Adding Structure: Social Network Inference with Graph Priors

Han Liu
ECE, UC-Davis
bhgliu@ucdavis.edu

Stratis Ioannidis
ECE, Northeastern
ioannidis@ece.neu.edu

Smriti Bhagat
Facebook
smr@fb.com

Chen-Nee Chuah
ECE, UC-Davis
chuah@ucdavis.edu

ABSTRACT

We study the problem of social network graph inference, whereby the topology of user interaction networks, as well as the strength of pairwise influences, are inferred from traces of information cascades. We propose a framework introducing graph structural priors into the above inference process. This framework allows us to capture different priors on the graph's degree distribution, including, e.g., stretched exponential, power-law, and an approximation of log-normal, which are important due to their natural prevalence in real social networks and many other complex graphs. We show that network inference under our model is amenable to the so-called majorize-minimize method, and that its implementation is tractable, as each step amounts to solving a convex optimization problem. We evaluate our method over synthetic datasets as well as real-world datasets from Twitter and a Facebook gifting application. We observe that network inference incorporating our structural priors significantly outperforms state-of-the-art inference.

1. INTRODUCTION

Online Social Networks (OSNs) have long been perceived as platforms over which information spreads and, as such, they have been extensively used by marketing companies, grass-roots movements, and political strategists for the implementation of "word-of-mouth" campaigns [8]. Following the seminal work by Kempe et al. [16], there has been an extensive research effort around the optimal design of such campaigns [17, 5, 13, 2]; typically, such efforts rely on parametrized models, such as the independent cascade [9] and the linear threshold model [12, 26]. Motivated by this state of affairs, learning the parameters of these models—or variants thereof—from datasets of information cascades has recently received considerable attention [11, 23, 7, 21, 4].

In short, the above works attempt to solve a problem commonly referred to as the *social network inference* problem. This amounts to observing a sequence of cascades—e.g., adoptions of a product across a population, the spread of tweets, hashtags or URLs over Twitter, etc.—and inferring the underlying user-to-user interaction network structure over which propagations take place and, *a fortiori*, the relative influence users exert on each other. The latter is model dependent; for example, in the independent cascade model [4, 21, 23], which we also adopt, the influence of user i to user j is captured by a probability that, given that i is "recruited", it successfully recruits user j. Learning pairwise user influence thus amounts to training these probabilities from cascade traces, while learning the underlying graph amounts to classifying the edges between users as existing or non-existing based on the inferred probability.

In this work, we incorporate *graph structural priors* to the network inference problem. This allows us to capture inherent information that may be known a priori about the underlying network. For example, social networks have been reported to follow power law [29, 1, 18, 20], stretched exponential [24, 30, 14], or log-normal [25, 10] degree distributions. Our approach incorporates such information in the inference process, leading to better estimation.

Clearly, as the number of graphs grows exponentially with the graph size, introducing prior distributions may lead to a combinatorial explosion. As such, a careful selection of the prior structure is necessary. Moreover, the parameter estimation problem that results from the introduction of priors should be tractable. In existing prior-free models [4, 21, 23], parameter estimation (i.e., learning the influence probabilities), reduces to solving a convex optimization problem. Standard regularization approaches can easily break this convex structure [21, 19]. It is unclear how to introduce, e.g., a power-law or a stretched exponential prior while maintaining tractability. In this paper, we tackle these challenges and make the following contributions:

- We present a generic scheme for introducing degree-dependent priors to social network inference problems. Even though the resulting inference problem is not convex, we show that our priors are amenable to analysis through the *majorize-minimize* (MM) method [15]. It is also highly scalable: multiple MM iterations, executed in parallel, lead to an increase in the likelihood of the computed estimates. Each iteration solves a convex optimization problem, thus ensuring tractability.
- We evaluate our method over both synthetic and real datasets. Despite the lack of global convexity and potential convergence to local minima, our method outperforms convex methods in inferring both the underlying topology and the influence strengths.

We describe prior work in Sec. 2. Sections 3 and 4 present our problem formulation and proposed solution, respectively. We evaluate our approach in Sec. 5, and conclude in Sec. 6.

2. RELATED WORK

There are several recent approaches inferring the underlying unobserved social network from cascade traces [21, 23,

11, 4]. The most relevant to our study is the methodology by Myers and Leskovec [21], who show that maximum likelihood estimation (MLE) under a version of the independent cascade model reduces to a convex optimization problem [3] (see also Appendix A). A similar reduction to a convex optimization is used by Netrapalli and Sanghavi [23]. Both works also observe that the above optimization problems are separable, and thus amenable to large-scale parallelization. Netrapalli and Sanghavi [23] also study the sample complexity of this inference. Abrahao et al. [4] show that a simple "first-edge" algorithm infers the graph using a number of samples within a logarithmic factor from the optimal, if all nodes appear as seeds sufficiently often—an assumption that may not hold in practise.

We apply a framework similar to the work of Myers and Leskovec [21], and Netrapalli and Sanghavi [23], also reducing maximum likelihood estimation of our model to a parallelizable convex optimization problem. Nevertheless, we depart from this earlier work by incorporating generic priors on the graph over which cascades take place. Myers and Leskovec propose incorporating a simple Laplace-based prior; our work covers a much wider class of priors, a special case of which is the one provided by Myers and Leskovec. Additional technical difficulties arise from this generalization: in contrast to [21, 23], we go beyond convex optimization by using the majorize-minimize (MM) method.

We also showcase how to use our model on power-law networks, stretched exponential networks and log-normal networks. These three degree distributions are most commonly reported in empirical social network data analyses. Power-law node degree distribution is usually seen in the user friendship graphs of large OSN platforms, such as MySpace, Orkut [1], Youtube, Flicker [20], and Sina Microblogging (the Chinese version of Twitter) [29]. The node degrees of most Twitter users follow power-law distribution as well, except for extremely popular users (with more than 10^5 followers) who are very rare (less than predicted by power-law) in Twitter [18]. A log-normal degree distribution has been observed in Slashdot [10] and the news spreading subnet on Digg and Twitter [25], while a stretched exponential distribution has been observed in, e.g., collaboration networks [30], email communication graphs [24], and in-campus social networks among music listeners [14].

The MM method has many applications [15], the most well-known being the Expectation-Maximization (EM) algorithm [28]. Closer to the setting we consider here, Liu and Ihler propose its use for maximum likelihood estimation of power-law (i.e., scale-free) graphical models [19]. However, there are two key differences between the work of Liu and Ihler and our work. First, we consider a different likelihood-estimation procedure–namely, graph inference from traces. This introduces an additional technical difficulty as, in contrast to the work of Liu and Ihler, the prior-free case of graph inference may *not* be convex. Second, unlike the work of Liu and Ihler, we do not focus only on power-law models. In fact, our analysis spans a generic class of priors, which can be used for graphs with different degree distributions.

3. PROBLEM DESCRIPTION

We consider the following setup. We are given a set of n users $V = \{1, \ldots, n\}$, and observe a series of cascades over these users. In particular, each cascade amounts to the propagation of, e.g., a piece of information (a hashtag, a rumor), or the adoption of a product. We represent a cascade c through n time-stamps $T^c = \{t_i^c\}_{i \in V}$, each indicating the time at which user i was *recruited* (i.e., adopted the product, obtains the piece of information, etc.). If user i did not get recruited into cascade c, we assume by definition that $t_i^c = +\infty$. We denote by $\mathcal{C}$ the set of all cascades and by $T = \{T^c : c \in \mathcal{C}\}$ the *input* data traces, i.e., the information available about who was recruited, and when.

The observed recruitment records are evidence of a cascade over a social network. In particular, there exists a directed graph $G(V, E)$ whose nodes are the users V and its edges E connect users that interact with, and hence can recruit, each other. For example, the existence of an edge $(i, j) \in E$ implies that user i has an influence over user j: whenever i gets recruited, it may contact user j (e.g., by posting the new information on their blog or Twitter feed) and trigger j's recruitment. Moreover, not all influence relationships are equal: some users may be more influential than others, and be more likely to recruit their neighbors.

Our goal is to infer both (a) the topology of the underlying interaction network $G(V, E)$ as well as (b) the strength of influence of each edge, simply by observing the trace of cascades T. To stress the challenge behind this task, we only observe *when* someone was recruited without explicitly observing *who* caused this recruitment. In what follows, we formalize the influence model we use in our analysis.

3.1 Probabilistic Influence Model

We follow the model by Myers and Leskovec [21] and Netrapalli and Sanghavi [23], which itself is an adaptation of the classic independent cascades model [16]. Whenever user i is recruited, it attempts to recruit all its neighbors in G. Attempts are independent Bernoulli random variables, and for $(i, j) \in E$ the probability that i succeeds in recruiting j is $b_{ij} \in (0, 1]$. If j's recruiting succeeds, the infection/adoption manifests after a time t from the time node i was recruited, where t is sampled from a well-known probability distribution (e.g., Poisson, exponential, etc.). We denote by $w(t)$, $t \geq 0$, the density function of this distribution.

The above formulation gives a principled means for attempting to discover the graph $G(V, E)$ as well as the influence strength of each individual through MLE. Let $B = \{b_{ij}\}_{i,j \in V}$ be the matrix of influence probabilities, with $b_{ij} = 0$ if $(i, j) \notin E$. Graph G can be obtained from the support of B; hence, the estimation of the graph and the strength of each pairwise influence amounts to estimating B.

Let $L(T; B)$ be the probability (likelihood) that trace T occurs, given the influence probabilities B, computed as:

$$L(T;B) = \prod_{i \in V} \Big[\prod_{c \in C : t_i^c = \infty} \Big(\prod_{j : t_j^c < \infty} (1 - b_{ji}) \Big) \prod_{c \in C : t_i^c < \infty} \Big(1 - \prod_{j : t_j^c \leq t_i^c} \big(1 - w(t_i^c - t_j^c) b_{ji}\big) \Big) \Big]$$

Using this notation, MLE of B from the trace T amounts to solving the following optimization problem:

$$\begin{aligned} \text{Minimize :} \quad & -\log L(T;B) \\ \text{subject to :} \quad & b_{ij} \in [0,1], \text{ for all } i, j \in V. \end{aligned} \tag{1}$$

where

$$-\log L(T;B) = -\sum_{i \in V} \Big[\sum_{c \in C : t_i^c = \infty} \sum_{j : t_j^c < \infty} \log(1 - b_{ji}) + \sum_{c \in C : t_i^c < \infty} \log \Big(1 - \prod_{j : t_j^c \leq t_i^c} \big(1 - w(t_i^c - t_j^c) b_{ji}\big) \Big) \Big]$$

The MLE (1) is separable, and thus is amenable to parallelization: it can be reduced to solving n simpler optimization problems, one for each $i \in V$, each of which can be solved by a different processor [21, 23]. Crucially, although (1) is *not* convex, there is a way of transforming each of these n problems to *convex optimization problems* which can thus be solved using standard techniques [3]. For completeness, we review in Appendix A both the separation of (1) into n constituent problems, as well as the reduction to convex optimization, as proposed in [21, 23].

3.2 Introducing Graph Priors

We have seen that the problem of estimating B through MLE (1) reduces to solving n convex optimization problems. In this work, we wish to incorporate *prior information on G's structure* to this estimation task. In particular, we wish to embed the prior information such as node degree distributions in G, e.g., a power-law or some other well-known distribution. In particular, let $P(B)$ be a given prior distribution over the model parameters B; MLE (1) becomes the following maximum a posteriori estimation in this case:

$$\begin{aligned} \text{Minimize}: \quad & -\log L(T;B) - \beta \log P(B) \\ \text{subject to}: \quad & b_{ij} \in [0,1], \text{ for all } i,j \in V, \end{aligned} \tag{2}$$

where the regularization term $-\beta \log P(B)$ penalizes solutions B with small prior probability. The *regularization parameter* $\beta > 0$ moderates the significance of this penalty.

After introducing the prior term, we face the following challenge: *contrary to* (1)*, the estimation problem* (2) *may not be readily reducible to a convex problem*! On the other hand, for many real-world applications, the structure of underlying user interaction network is already known (e.g., power-law, stretched exponential or log-normal), and incorporating this structure can yield a significant improvement in the estimation of both the graph G as well as influence probabilities B. We propose a general class of priors that are of interest because they can approximate many interesting well-known cases of graph structures, including the power-law and stretched exponential distribution. The MLE problems resulting from incorporating these priors are not necessarily convex, nor can they be reduced to convex problems: nonetheless, we show that they are amenable to a solution through the majorization-minimization method.

4. WEIGHT-INVERSE GRAPH PRIORS

We consider a general class of priors of the form:

$$P(B) = \textstyle\prod_{i \in V} f(\sum_{j \in V \setminus \{i\}} \frac{1}{1-b_{ji}}), \tag{3}$$

where f satisfies the following assumption:

ASSUMPTION 1. *The density function f is strictly positive, differentiable, log-convex and non-increasing in $\mathbb{R}^+$.*

Notice that this is not a limiting assumption, as several common priors such as Laplace and exponential priors satisfy Assumption 1. More precisely, the Laplace priors:

$$f(x) = Ce^{-\lambda x}, \tag{4}$$

the power-law prior:

$$f(x) = Cx^{-\alpha}, \tag{5}$$

and the stretched exponential prior:

$$f(x) = Ce^{-x^{\alpha}}, \tag{6}$$

for some $\alpha > 0$, $C > 0$, all satisfy Assumption 1. In all these cases, the constants C are positive numbers such that the integral of f is 1 over $[0,1]^{n-1}$, the feasible domain of $b_{\cdot i}$.

In general, the "weight-inverse" graph priors (3) have the following properties: First, increasing b_{ji} decreases the probability P. Therefore, the estimation problem (2) penalizes solutions with large values of B. Second, they *heavily penalize* influence probabilities approaching 1. This induces sparsity, which is a highly desirable property in problems with few traces; indeed, the term P in (2) acts as a regularization factor.

Setting f to be the Laplace prior (4) allows us to recover the prior used by Myers and Leskovec as a special case of (3). In contrast to general f, in this setting (2) can then be reduced to a single convex optimization problem [21]. Though this reduction does not apply to (2) in the general case, we show that (2) can still be solved through the majorization-minimization (MM) method [15]. For completeness, we give a brief description of MM below.

4.1 The MM Method

Consider an optimization of the form

$$\begin{aligned} \text{Minimize}: \quad & L(x) = G(x) + F(x) \\ \text{subject to}: \quad & x \in D \end{aligned} \tag{7}$$

where D is some subset of $\mathbb{R}^d$ (not necessarily convex) and F is concave and differentiable in D. MM amounts to:

$$x_{k+1} = \arg\min_{x \in D} \Big(G(x) + \nabla F(x_k)^T (x - x_k) \Big) \tag{8}$$

Note that implementing MM presumes that the minimization (8) can be computed efficiently, which is the case if, e.g., G is a convex function and D is a convex domain. In any case however, the following theorem implies that the procedure (8) finds a "local minimum" of (7).

THEOREM 1. *Procedure* (8) *satisfies $L(x_{k+1}) \leq L(x_k)$, for all $k \geq 0$, i.e., the objective decreases with each step.*

We provide a proof in Appendix B.

4.2 Application of MM to Graph Inference

The product form of (3) implies that the problem (2) is separable, and can be solved by solving n optimization problems. For each $i \in V$ it suffices to solve:

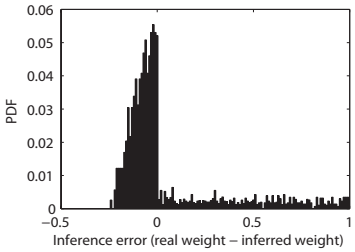
$$\begin{aligned} \text{Minimize}: \quad & \mathcal{L}_i(T; b_{\cdot i}) - \beta \log f\Big(\sum_{j \in V \setminus \{i\}} \frac{1}{1-b_{ji}} \Big) \\ \text{subject to}: \quad & b_{ij} \in [0,1], \text{ for all } j \in V \setminus \{i\}, \end{aligned}$$

where $\mathcal{L}_i$ is given by

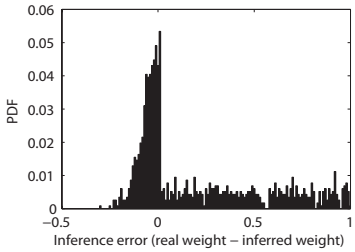
$$\begin{aligned} \mathcal{L}_i(T; b_{\cdot i}) = & -\textstyle\sum_{c \in C: t_i^c = \infty} \sum_{j: t_j^c < \infty} \log(1 - b_{ji}) \\ & - \textstyle\sum_{c \in C: t_i^c < \infty} \log\Big(1 - \prod_{j: t_j^c \leq t_i^c} \big(1 - w(t_i^c - t_j^c) b_{ji}\big)\Big). \end{aligned} \tag{9}$$

Using the transformation $y_j = \frac{1}{1-b_{ji}}$, and letting $y = \{y_j\}_{j \in V \setminus \{i\}}$ we can rewrite this as:

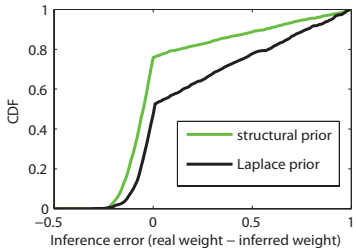
$$\begin{aligned} \text{Minimize}: \quad & \mathcal{L}_i(T; y) + \beta F(y) \\ \text{subject to}: \quad & y \in \mathbb{R}_+^{n-1} \end{aligned} \tag{10}$$



(a) PDF of the inference error on edge weight, using structural prior. The average absolute error is 0.1776.



(b) PDF of the inference error on edge weight, using Laplace prior. The average absolute error is 0.2635.



(c) CDF of the inference error on edge weight.

Figure 1: Performance comparison between structural prior and previously proposed Laplace prior that does not include the graph structure information. Experiments are done upon synthetic power-law network, when best edge weight inference accuracy is obtained.

where $F(y) = -\log f(\sum_{j\in V\setminus\{i\}} y_j)$. Under Assumption 1, $F(y)$ is differentiable and concave. Hence, we can apply MM to (10), yielding the following iterative procedure:

$$y^k = \arg\min_{y\in\mathbb{R}_+^{n-1}} \left(\mathcal{L}_i(T;y) + \beta\nabla F(y^{k-1})^T(y-y^{k-1})\right) \quad (11)$$

Though Theorem 1 ensures that (11) yields an improvement with each step, the minimization involved *may still not be a convex optimization problem.* Nevertheless, we show that, under Assumption 1, it can be reduced to one through an appropriate transformation of variables:

THEOREM 2. *Under Assumption 1, the estimation method of* (11) *decreases the objective in* (10) *with each step. Moreover, the minimization in* (11) *can be converted to a convex optimization problem.*

PROOF. The first statement follows from Theorem 1 and Assumption 1. To prove the second statement, define $d_{ji} \equiv \log(1-b_{ji}) = -\log y_j$, and $\gamma_c \equiv 1 - \prod_{j:t_j^c\le t_i^c}\left(1-w(t_i^c-t_j^c)b_{ji}\right)$. Let $d = \{d_{ji}\}_{j\in V\setminus\{i\}}$, and $\gamma = \{\gamma_c\}_{c\in\mathcal{C}}$, and

$$G(d,\gamma) = -\sum_{c\in C:t_i^c=\infty}\sum_{j:t_j^c<\infty} d_{ji} - \sum_{c\in C:t_i^c<\infty}\gamma_c, \text{ and}$$

$$F(d) = -\log f\left(\textstyle\sum_{j\in V\setminus\{i\}}(1-e^{d_{ji}})\right)$$

Then, we can rewrite (10) as:

$$\begin{aligned} &\text{Minimize:} \quad G(d,\gamma) + \beta\sum_{j\in V\setminus\{i\}}\frac{\partial F(y^k)}{\partial y_j}(e^{-d_{ji}} - y_j^k) \\ &\text{subject to:} \quad (d,\gamma)\in D. \end{aligned} \quad (12)$$

where D is the convex set of constraints in the prior-free problem (14), in Appendix A. Under Assumption 1, F is non-decreasing, so $\frac{\partial F(y^k)}{\partial y_j} \ge 0$ for all $j \in V\setminus\{i\}$. As a result, $\sum_{j\in V\setminus\{i\}}\frac{\partial F(y^k)}{\partial y_j}(e^{-d_{ji}} - y_j^k)$ is convex. □

We note that, inder the Laplace-based prior in [21], the MM method (11) becomes degenerate: it reaches a fixed point in a single iteration.

5. EVALUATION

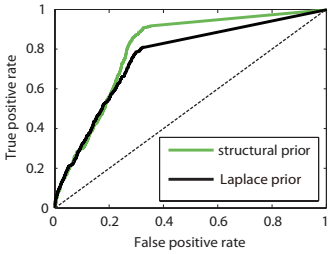
Our evaluation includes both data from synthetically generated networks as well as real OSN data from Twitter and a Facebook app, *iHeart*. We investigate the performance of network inference from two perspectives: the accuracy of inferring the existence of edges and the accuracy of inferring the edge weight (recruitment probability).

We study three degree distributions that are frequently observed in OSNs and many other complex real world graphs, namely power-law [1, 20, 29, 18], stretched exponential [30, 14, 24], and log-normal [10, 25]. We compare these to the Laplace prior (4) used by Myers and Leskovec [21]; recall that under the latter, MM (2) terminates in one iteration; equivalently, MLE reduces to solving a single convex optimization problem. Despite the lack of convexity, MM significantly outperforms this simple prior.

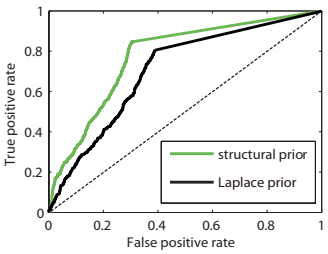
5.1 Evaluation on Synthetic Data

Our synthetic data experiments follow the setup in the experiments of Myers and Leskovec [21] and begin with the construction of underlying network. We first distribute 1000 nodes according to a homogeneous Poisson point process in a (2-dimensional) space of unit area. Then we randomly assign the in-degree for each node according to the power-law, stretched exponential, or log-normal distribution. For each node with in-degree l, we add l directed edges into the network using this node as the ending point. The starting point of an added edge is chosen randomly according to the WPR model [27], in which the connection probability between two nodes is a function of each node's distance to the ending point. After this network is constructed, each edge is assigned a uniformly random weight (recruitment probability) between 0 and 1.

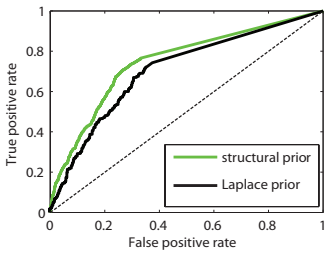
Over this randomly constructed network, we generate cascades by first selecting 100 random starting nodes. The propagation of the cascade starts from these nodes using the independent cascade model. The time t, for which every newly recruited node waits before manifesting the infection, follows an exponential distribution (to mimic the exponential waiting time distribution observed in all the real datasets we have). This generation process is repeated until a predetermined number of cascades are generated. For each cascade, we only record the time when every node gets recruited, which constitutes the input our inference algorithm.



(a) Power-law network (AUC of structural prior is 0.819, AUC of Laplace prior is 0.751).



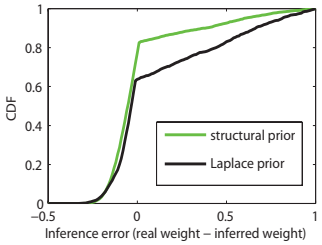
(b) Stretched Exponential network (AUC of structural prior is 0.7774, AUC of Laplace prior is 0.7013).



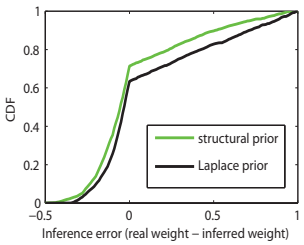
(c) Log-normal network (AUC of structural prior is 0.7322, AUC of Laplace prior is 0.7147).

Figure 2: Performance comparison when best edge existence inference accuracy is obtained, upon synthetic networks. AUC of structural prior and previously proposed Laplace prior which does not include the graph structure information are calculated for each study.

In our experiments, we quantify the performance of inference using two sets of metrics. First, we calculate the error between the inferred edge weight (recruitment probability) and the real weight to evaluate the inference accuracy. Second, our algorithm can also be used to infer the existence of edges in the underlying user interaction network. For this kind of inference, we can apply a threshold based binary classifier upon the estimated edge weights. With a given threshold, the classifier identifies those inferred edges, the estimated weights of which are lager than the threshold, as present in the real network and rest edges as not in the network. We then compute the receiver operating characteristic (ROC) curve [6], which illustrates the performance the classifier system as its discrimination threshold varies, to analyze how good the inference accuracy could be. The area under curve (AUC) of ROC is used as a quantitative measurement for this evaluation. For each experiment, we refer to the inference using corresponding prior degree distribution as "with structural prior", and compare it to inference with Laplace prior, which includes no structural information. Our experiments are conducted under optimal regularization parameter β found through exaustive search.



(a) Stretched exponential network (average error of structural prior and Laplace prior are 0.1461 and 0.2249).



(b) Log-normal network (average error of structural prior and Laplace prior are 0.2047 and 0.2433, respectively).

Figure 3: Performance comparison when best edge weight inference accuracy is obtained, upon synthetic networks. The average relative absolute inference error of structural prior and previously proposed Laplace prior which does not include the graph structure information are calculated.

5.1.1 Power-Law Synthetic Graph

The exponent parameter α of our power-law synthetic graph (5) is set to be 1.5 (as observed in our empirical datasets, this value is generally between 1.5 to 3).

Our experiment results are based on 30 synthetic cascades, which can provide similar average number of node recruitment as we observed in empirical data. We find that different β is required for obtaining the lowest error in edge weight (recruitment probability) inference and the highest AUC in edge existence inference. We get the best edge weight inference when setting the regularization parameter to be 150 for our method and 10 for the existing method with a Laplace regularization term [21].

The experiment results are shown in Figure 1. We can see from Figure 1 (a) and (b) that the distribution of inference error centers at the peak where error 0. In addition, the experiment results also show that the average absolute error of our algorithm incorporating prior knowledge on the underlying network structure is as low as 67% of the error incurred by [21]. On the other hand, for the best inference of edge existence, the regularization parameter should be set to 15 for our method and 0.025 for Laplace regularization. The resulting performance is illustrated in Figure 2 (a), which also shows that our method outperforms [21].

5.1.2 Stretched Exponential Synthetic Graph

In the stretched exponential degree distribution (6) graph generation, α is set as 0.5, which is similar to the value of several real networks reported in previous studies [14, 24].

The experiment results when regularization parameters are set for best edge weight inference ($\beta = 150$ for our structural prior, $\beta = 20$ for Laplace prior) are shown in Figure 3 (a). Our algorithm incorporating prior knowledge on the underlying network structure achieves an average absolute error only 62.9% of the error incurred by [21]. On the other hand, when the best edge existence inference is obtained ($\beta = 15$ for structural prior, $\beta = 0.01$ for Laplace regularization), Figure 2 (b) shows that the AUC of our method is 0.7774 while the AUC of exiting method applying Laplace regularization is 0.7013.

5.1.3 Log-Normal Synthetic Graph (Approximation)

The probability density function (PDF) of log-normal distribution, $f(x)$, takes the following form

$$\log f(x) = -\log(x) - \frac{1}{2}\left(\frac{\log(x)-\mu}{\sigma}\right)^2 + const$$

In the synthetical graph generation, μ is set as 1 and σ is set as 2, which are within the value ranges that have been observed in real networks [10]. Unfortunately, $f(x)$ does not follow the Assumption 1, so our approach cannot be directly used in this case. As suggested by Liu et al. [19], we use only the first term $-\log(x)$ as an approximation of $\log f(x)$.

The experiment results when regularization parameters are set for best edge weight inference ($\beta = 350$ for our structural prior, $\beta = 25$ for Laplace prior) are shown in Figure 3 (b). Our algorithm incorporating prior knowledge on the underlying network structure achieves an average absolute error around 84.1% of the error incurred by [21]. On the other hand, Figure 2 (c) shows the comparison results when the best edge existence inference is obtained ($\beta = 20$ for structural prior, $\beta = 0.05$ for Laplace regularization). Although our algorithm does not perform as good as in the power-law and stretched exponential experiments, it is still outperforms the Laplace prior used in [21].

5.2 Evaluation on Real Data

Two real datasets are used in our evaluation experiments. One is a Twitter trace (from Aug 1, 2009 to Aug 31, 2009), made publicly available by the Stanford Network Analysis Project (SNAP). The other is based on a popular Facebook gifting application operated by Manakki LLC at the time of data collection (from the 25th week of 2009 to the 28th week of 2009). An empirical study of this application can be found in the work of Nazir et al. [22]. Both datasets are summarized in Table 1. Due to the large number of user activities recorded in the data, we sample a small subset of users for our experiment. The node sampling is done through a breadth first sampling (BFS) through the network, starting from nodes that have participated in the largest number of cascades. BFS is used for sampling because we are interested in preserving the complete structure of the subgraph, among which influence expands through a selected subset of nodes in the original graph.

Since in real datasets the ground truth of edge weight (recruitment probability) is unavailable, in this section we evaluate the accuracy of weight inference by simulating the user recruitment process based on estimated recruitment probabilities, and compare the resulting cascades with real records. We conduct a group of experiments for each dataset. In every experiment, the propagation of one cascade is simulated. We call this cascade as the "target cascade". For a target cascade, all involved nodes can be divided into two groups: (1) seeders and (2) invited nodes. Seeders are users who are, as observed in the empirical records, involved in the target cascade without any neighbor being recruited earlier, and the users other than seeders are categorized as invited nodes. In an experiment, we set the recruitment time for seeders to be exactly the same as in empirical records, and simulate how the invited users behave according to the the independent cascade model.

To ensure the reliability of our results, we perform N-fold cross validation (N is the number of cascades), i.e. we remove the data of one cascade when inferring recruitment probabilities, and then evaluate the inference results by simulating on the cascade that was taken out. Because the area covered by each cascade distributes unevenly among real social networks, the number of users each node recruited in a data trace varies dramatically. As a result, in our empirical traces some nodes are recruited for very few times. Certain portion of nodes even get recruited only once in the whole data record. For those nodes, if the cascade in which they are recruited is taken out for cross validation, the rest data can provide no information for our algorithm to make any useful inference on them. Under such a condition, any comparison between our structural prior based MLE and [21] would be meaningless. To deal with this issue of too few recruitments, we select only part of cascades from each dataset for the cross validation experiment. The selection rule is: for nodes participating each cascade, we calculate the average number that they have been recruited in the whole dataset, and then pick the top 10 cascades.

Table 1: Dataset Overview

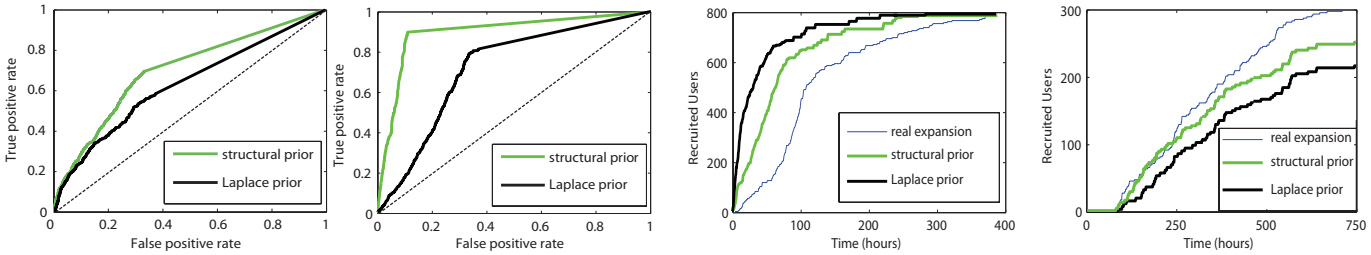
Data	Duration	Total Users	Sampled Users	Selected Cascade Num
iHeart	28 days	1.1M	3642	40
Twitter	31 days	484K	3352	100

5.2.1 Evaluation Based on iHeart Data

iHeart is a facebook based gifting application launched in June 2009, and was ranked as one of the top three Facebook applications by monthly active users in December 2009. In *iHeart*, users send different types of virtual "hearts" (a particular type of virtual gifts) to each other, and new types of hearts are launched by the developer over time. For every new heart (or gift) type, some users adopt it on their own and send it to other users. When received, these new types of hearts constitute a form of invitation, exposing receivers to new hearts and effectively recruiting them as new adopters. A receiver in the data trace is considered "successfully recruited" when he/she starts sending this same type of hearts to other users. *iHeart* represents an important kind of OSN cascades, where users can explicitly choose which neighbors they want to influence. We use the time when every invited node gets recruited (the time would be infinite if the recruitment does not happen) in each cascade as input for the inference algorithms, and evaluate the results.

Since in the dataset of *iHeart*, we have the exact record of sender ID and receiver ID for each virtual gift sent, it is possible to recover the ground truth of which edge exists in the underlying user-to-user interaction network. The node degree distribution of this underlying network can be fitted to power-law distribution with exponent parameter $\alpha = 2.59$ using least square fitting. We first set regularization parameter for best edge existence inference ($\beta = 10$ for our structural prior, $\beta = 0.05$ for Laplace prior), and the corresponding ROCs are shown in Figure 4 (a).

Then the regularization parameter is set for best edge weight (pairwise influence) inference ($\beta = 150$ for our structural prior, $\beta = 15$ for Laplace prior). For the simulations on the expansion of cascades, the statistics on the average relative absolute errors of all simulation results are shown in Table 2. Since, after a cascade starts, it takes certain time for the growth of user number to become stable, we calculate the error 48 hours after the cascade starts. We also randomly select 1 simulation result and illustrate it in Figure 4 (c).



(a) data "heart type" 0 (AUC of structural prior is 0.6996. AUC of Laplace prior is 0.6422).

(b) Twitter data "indexed URL" 20 (AUC of structural prior is 0.8981. AUC of Laplace prior is 0.7401).

(c) *iHeart* data "heart type" 0 (Average relative absolute inference errors of structural prior and Laplace prior are 0.2440 and 0.4798)

(d) Twitter data "indexed URL" 20 (Average relative absolute inference errors of structural prior and Laplace prior are 0.122 and 0.304)

Figure 4: Real data based performance comparison, between structural prior and previously proposed Laplace prior that excludes graph structure information. (a) and (b) are the results when best edge existence inference accuracy is obtained, while (c) and (d) are the results when best edge weight inference accuracy is obtained.

Table 2: Statistics on the Average Relative Absolute Error of Simulation Results for Different Set of Experiments, Using *iHeart* Data

Experiment	Mean	Max	Min
structural prior, best weight inference	0.3354	0.6201	0.1563
structural prior, best existence inference	0.4089	0.8029	0.1536
Laplace prior, best weight inference	0.4125	0.7213	0.1528
Laplace prior, best existence inference	0.5924	1.280	0.3655

Table 3: Statistics on the Average Relative Absolute Error of Simulation Results for Different Set of Experiments, Using Twitter Data

Experiment	Mean	Max	Min
structural prior, best weight inference	0.4633	0.6252	0.1221
structural prior, best existence inference	0.792	1.1376	0.5312
Laplace prior, best weight inference	0.6289	0.8907	0.3043
Laplace prior, best existence inference	1.152	1.8621	0.7101

In general, the observations from the experiments on *iHeart* agree with our findings in the synthetic experiments. Our method can achieve a smaller error in the simulation experiment than existing approach when β is properly set. Moreover, when used for inferring the topology of underlying user interaction network, our method can also get a AUC higher than inference applying Laplace prior, which does not include graph structural information.

5.2.2 *Evaluation Based on Twitter Data*

We use the dataset crawled from Twitter by Yang and Leskovec (2011) and made publicly available by SNAP. It was collected between June and December 2009, and includes about 20-30% of all tweets posted during this period. We focus on a single month of the trace (August 2009). We consider information cascades formed by spreading posted URLs through the follower/followee graph. When a user first posts a URL, we consider her/him as recruited into the corresponding cascade, and record the recruitment time as input for the inference algorithm. We take out spamming URLs which are only tweeted or intensively re-tweeted by their seeders, and randomly select 100 URLs from the rest for our experiments. The selected URLs are anonymized and indexed with integer numbers from 1 to 100. Different from *iHeart*, Twitter represents the other kind of cascades where the spreading of influence is broadcasting in nature, i.e. users cannot choose specific neighbors to "recruit".

From the Twitter data, we cannot extract the ground truth of the exact underlying user-to-user interaction network. As a result, we can only use the follower-followee network available in our dataset as an approximation. According to our empirical analysis on twitter, for most users, certain portion of their followers never react to any information they post. This means that the approximation we use is actually a super set of the underlying user-to-user interaction network. The node degree distribution of the follower-followee network can be fitted to power-law distribution with exponent parameter $\alpha = 2.12$ using least square fitting. We first set the regularization parameter for best edge existence inference ($\beta = 10$ for our structural prior, $\beta = 0.025$ for Laplace prior), and the corresponding ROCs are illustrated Figure 4 (b). The result shows that the AUC is higher for our method than the existing estimation algorithm applying Laplace prior, which excludes graph structural information.

We then set the regularization parameter for best edge weight (pairwise influence) inference ($\beta = 150$ for our structural prior, $\beta = 10$ for Laplace prior). The experiment results are shown in Table 3 and Figure 4 (d) (a randomly selected cascade), indicating that our method based on structural priors outperforms the Laplace prior approach in cascade expansion simulations as well.

6. CONCLUSIONS

In this paper we propose a framework to incorporate graph structural priors, which capture the structural characteristics of a wide array of graph degree distributions, into the problem of inferring the underlying topology of user-

to-user interactions and influence. We demonstrate how to iteratively solve our inference problem using the so-called majorize-minimize method, which is tractable, as each step amounts to solving a convex optimization problem. The performance of our method is demonstrated over synthetic datasets as well as real-world datasets from Twitter and a Facebook gifting application.

7. ACKNOWLEDGEMENT

This work is partly supported by NSF CNS-1302691 grant and Technicolor.

8. REFERENCES

[1] Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings ACM WWW*, 2007.

[2] S. Bhagat, A. Goyal, and L.V. Lakshmanan. Maximizing product adoption in social networks. 2012.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] A. Bruno, C. Flavio, K. Robert, and P. Alessandro. Trace complexity of network inference. In *Proceedings of KDD*, 2013.

[5] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *ACM SIGKDD*, 2009.

[6] T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[7] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer. Outtweeting the twitterers-predicting information cascades in microblogs. In *Proceedings of the OSN*, 2010.

[8] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 2001.

[9] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Sci Review*, 2001.

[10] V. Gómez, A. Kaltenbrunner, and V. López. Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of ACM WWW*, 2008.

[11] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM TKDD*, 2012.

[12] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, pages 1420–1443, 1978.

[13] J. Hartline, V. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. 2008.

[14] H. Hu and D. Han. Empirical analysis of individual popularity and activity on an online music service system. *Physica A: Statistical Mechanics and its Applications*, 2008.

[15] D. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.

[16] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD*, 2003.

[17] M. Kitsak, L. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. Stanley, and H. Makse. Identification of influential spreaders in complex networks. *Nature Physics*, pages 888–893, 2010.

[18] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *ACM WWW*, 2010.

[19] Q. Liu and A. Ihler. Learning scale free networks by reweighted l1 regularization. In *Proceedings of the AISTATS*, 2011.

[20] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of ACM IMC*, 2007.

[21] S. Myers and J. Leskovec. On the convexity of latent social network inference. In *NIPS*, 2010.

[22] A. Nazir, A. Waagen, V. Vijayaraghavan, C-N. Chuah, R. D'Souza, and B. Krishnamurthy. Beyond friendship: modeling user activity graphs on social network-based gifting applications. In *Proceedings ACM IMC*, 2012.

[23] P. Netrapalli and S. Sanghavi. Finding the graph of epidemic cascades. In *Sigmetrics*, 2012.

[24] M. Newman, S. Forrest, and J. Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 2002.

[25] K. LermanÂąÂąand R.Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. *ICWSM*, 2010.

[26] D. Watts. A simple model of global cascades on random networks. *Proceedings of the National Academy of Sciences*, 99(9):5766–5771, 2002.

[27] L. Wong, P. Pattison, and G. Robins. A spatial model for social networks. *Physica A: Statistical Mechanics and its Applications*, 360(1):99–120, 2006.

[28] C. Wu. On the convergence properties of the em algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

[29] D. Zhang and G. Guo. A comparison of online social networks and real-life social networks: A study of sina microblogging. *Mathematical Problems in Engineering*, 2014.

[30] P. Zhang, K. Chen, Y. He, T. Zhou, B. Su, Y. Jin, H. Chang, Y. Zhou, L. Sun, B. Wang, et al. Model and empirical study on some collaboration networks. *Physica A: Statistical Mechanics and its applications*, 2006.

APPENDIX

A. PRIOR-FREE CASE

The MLE (1) [23, 21] is separable, i.e., can be reduced to solving n simpler optimization problems, each of which can be solved by a different processor [21, 23]: for each $i \in V$, one can solve

$$\begin{aligned} \text{Minimize}: \quad & \mathcal{L}_i(T; b_{\cdot i}) \\ \text{subject to}: \quad & b_{ij} \in [0,1], \text{ for all } j \in V \setminus i, \end{aligned} \tag{13}$$

where $b_{\cdot i} = \{b_{ji}\}_{j \neq i}$ be the vector of influence probabilities of users influencing i and $\mathcal{L}_i$ is given by (9).

The above problem, though not convex, can be transformed to a convex optimization problem [21, 23] and, as such, can be solved using well known techniques [3]. In particular, it can be shown that it reduces to solving, for each $i \in V$:

$$\text{Min.:} \quad -\sum_{c \in C: t_i^c = \infty} \sum_{j: t_j^c < \infty} d_{ji} - \sum_{c \in C: t_i^c < \infty} \gamma_c \tag{14a}$$

$$\text{s.t.:} \quad d_{ji} \leq 0, \text{ for all } j \in V \setminus \{i\} \tag{14b}$$

$$\gamma_c \leq 0, \text{ for all } c \in C \tag{14c}$$

$$\log\left(e^{\gamma_c} + \prod_{j: t_j^c \leq t_i^c} \left(1 - w(t_i^c - t_j^c)(1 - e^{d_{ji}})\right)\right) \leq 0 \tag{14d}$$

which is indeed a convex optimization problem. The solution B results by taking $b_{ji} = 1 - \exp d_{ji}$, where $\{d_{ji}\}_{i,j \in V}$ are obtained by solving the above n optimization problems.

B. PROOF OF THEOREM 1

PROOF. Since F is concave and differentiable,

$$F(x) - F(x') \leq \nabla F(x')^T (x - x')$$

which implies

$$L(x) - L(x') \leq L(x) + \nabla F(x')^T (x - x') - L(x')$$

Hence,

$$\begin{aligned} L(x_k) - L(x_{k-1}) &\leq L(x_k) + \nabla F(x_{k-1})^T (x_k - x_{k-1}) - L(x_{k-1}) \\ &= \min_{x \in D} \Big(L(x) + \nabla F(x_{k-1})^T (x - x_{k-1}) - L(x_{k-1}) \Big) \leq 0 \end{aligned}$$

where the latter inequality follows from the fact that the function minimized is zero for $x = x_{k-1}$. □