Inferring Diffusion Networks with Sparse Cascades by Structure Transfer

Senzhang Wang[1](✉), Honghui Zhang[2], Jiawei Zhang[3], Xiaoming Zhang[1], Philip S. Yu[3,4], and Zhoujun Li[1]

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
{szwang,yolixs,zjli}@buaa.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China
zhh11@mails.tsinghua.edu.cn
[3] Department of Computer Science, University of Illinois at Chicago, Chicago, USA
[4] Institute for Data Science, Tsinghua University, Beijing, China
{jzhan9,psyu}@uic.edu

Abstract. Inferring diffusion networks from traces of cascades has been intensively studied to gain a better understanding of information diffusion. Traditional methods normally formulate a generative model to find the network that can generate the cascades with the maximum likelihood. The performance of such methods largely depends on sufficient cascades spreading in the network. In many real-world scenarios, however, the cascades may be rare. The very sparse data make accurately inferring the diffusion network extremely challenging. To address this issue, in this paper we study the problem of transferring structure knowledge from an external diffusion network with sufficient cascade data to help infer the hidden diffusion network with sparse cascades. To this end, we first consider the network inference problem from a new angle: link prediction. This transformation enables us to apply transfer learning techniques to predict the hidden links with the help of a large volume of cascades and observed links in the external network. Meanwhile, to integrate the structure and cascade knowledge of the two networks, we propose a unified optimization framework TrNetInf. We conduct extensive experiments on two real-world datasets: MemeTracker and Aminer. The results demonstrate the effectiveness of the proposed TrNetInf in addressing the network inference problem with insufficient cascades.

Keywords: Information diffusion · Network inference · Transfer learning

1 Introduction

Utilizing cascades to infer the diffusion network is an important research issue and has attracted a great deal of research attentions recently [17,20,22,23]. In many scenarios, we only have the traces of information spreading in a network

DOI: 10.1007/978-3-319-18120-2_24

without explicitly observing the network structure. For example, in virus propagation we only observe which people get sick at what time, but without knowing who infected them [18]; in viral marketing, viral marketers can track when customers buy products or subscribe to services, but it is hard to exactly know who influence the customers' decisions [25]. Inferring the underlying connectivity of diffusion networks is of outstanding interest in many applications, such as technological innovations spreading [16], word-of-mouth effect in viral marketing [26], and personalized recommendation in E-commerce websites [24].

Traditional approaches normally formulate a generative probability model to find the network which can generate all the cascades with the maximum likelihood, such as ConNIe [20], NETINF [23], NETRATE [21], and InfoPath [22]. Although these models can work well on synthetic datasets, their performance on real-world datasets is usually undesirable [3,21]. This is firstly due to the fact that information diffusion on real-world networks is too complex for existing information propagation models to handle. Secondly, the performance of generative models largely relies on a large volume of cascades, while in real-world scenarios the cascades may be rare or at least not sufficient [19].

To address above mentioned problems, in this paper we will study *how to borrow the structure knowledge from an external diffusion network whose links are known to help us infer a diffusion network whose links are hidden by transfer learning.* In many cases, although the cascades in the hidden diffusion network are sparse, a network related to the hidden diffusion network is known and may be helpful for our task [6]. For example, we want to infer the network of who influencing whom to buy some products based on the transaction logs of users' purchase history, such as iPhone 5S. The result might be quite inaccurate if we do not have enough such logs. However, if we know their following relationships and tweets about iPhone 5S in Twitter, the diffusion process of the tweets among them may potentially help us infer who influenced whom to buy an iPhone 5S.

Transfer learning has achieved significant success in many machine learning tasks including classification [14,15], regression [13], and clustering [12] to address the problem of lacking enough training data in the target domain. However, it is challenging to directly exploit transfer learning to our task. Traditional generative models formulate this task as an optimization problem, hence it is naturally hard for such models to extract and map feature spaces from one domain to another for knowledge transfer. Meanwhile, transfer learning normally can only capture and transfer knowledge from the source domain. In our task, we need to consider not only the structure knowledge transferred from an external diffusion network, but also the cascade information in the hidden network. How to integrate the knowledge from two different networks in a unified scheme to obtain a better network inference model also makes the problem challenging.

In this paper, we first formulate the network inference problem as a link prediction task by extracting various cascade related features. The advantages of the formulation are two-fold: 1) it paves the way of applying transfer learning techniques for structure transfer; and, 2) link prediction does not rely much on the particular information propagation model. As the links of the external diffusion network are known, we can use these labeled links to train a prediction model for predicting the links in the hidden network by transfer learning. To incorporate the transferred structure knowledge from the external network with the cascades in the hidden network, we next propose a unified optimization framework TrNetInf. TrNetInf jointly maximizes the likelihood of generating the cascades in the hidden diffusion network and minimizes the difference between the links inferred by traditional generative model and those predicted by transfer learning model simultaneously. We evaluate TrNetInf on two real-world datasets: MemeTracker dataset and AMiner citation network dataset. Experimental results on both datasets demonstrate the superior performance of TrNetInf, especially when the cascades are not sufficient. The main contributions of this paper are as follows:

- For the first time, to the best of knowledge, we study the network inference problem with the challenge of lacking enough cascade data (Section 2).
- To transfer structure knowledge from one diffusion network to another, we consider the network inference problem from a new angle: link prediction. Meanwhile, as the links of the hidden network is unknown and structure based features are hence not available, we propose to extract a set of cascade related features for learning (Section 3.2).
- We further propose a unified optimization framework TrNetInf. TrNetInf can efficiently integrate knowledge from source and target diffusion networks, and combine the results from the traditional generative model and the proposed link prediction model (Section 3.3).
- We evaluate the proposed approach on two real-world datasets by comparing it against various baselines. The results verify its effectiveness in addressing the network inference problem with very sparse cascades (Section 4).

The remainder of this paper is organized as follows. Section 2 formally defines the studied problem. Section 3 details the proposed model. Section 4 evaluates the model with two real-world datasets, followed by related work in Section 5. Section 6 concludes this research with directions for future work.

2 Problem Statement

In this section, we will give some terminologies to help us state the problem. Then we will formally define the studied problem. In information diffusion, a diffusion network is usually referred to a network with a set of information spreading in it [21]. Based on the diffusion network, we formally define a hidden diffusion network as follows.

Definition 1 ***Hidden Diffusion Network*** $G_{\mathcal{H}}$***:*** *We define a diffusion network* $G_{\mathcal{H}} = (V, E_{\mathcal{H}})$ *as a hidden diffusion network if only its nodes can be observed but the edges are hidden and need to be inferred. Here* V *denotes the set of node and* $E_{\mathcal{H}}$ *denotes the hidden edges.*

There are usually many traces of information diffusion on a diffusion network. The traces are called cascades and can be formally defined as follows.

Definition 2 ***Cascade:*** *A cascade $\boldsymbol{t}^c$ associated with information c can be denoted as a N-dimensional vector $\boldsymbol{t}^c = (t_1^c, ..., t_N^c)^T$, where N is the number of nodes in the diffusion network. The i_{th} dimension of $\boldsymbol{t}^c$ records the time stamp when information c infects node i, and $t_i^c \in [0, T^c] \cup \{\infty\}$.*

The symbol ∞ labels nodes that are not infected during the observation window $[0, T^c]$. The time stamp is set to 0 at the start of each cascade. A cascade set $\mathbb{C}$ consists of a collection of cascades, i.e. $\mathbb{C} = \{\mathbf{t}^1, ..., \mathbf{t}^M\}$, where M is the number of cascades.

Based on above defined terminologies, the traditional network inference problem can be defined as follows [23].

Problem 1. *Given a hidden diffusion network $G_{\mathcal{H}} = (V, E_{\mathcal{H}})$ and a collection of cascades $\mathbb{C}$ on $G_{\mathcal{H}}$, the network inference problem aims to recover the network structure of $G_{\mathcal{H}}$, namely infer the hidden edges $E_{\mathcal{H}}$ based on the cascades $\mathbb{C}$.*

In our case, besides the hidden diffusion network we also have a related external diffusion network whose structure is known. Here we consider the hidden diffusion network as the target domain network and the related network as the source domain network. In traditional transfer learning setting, a *domain D* consists of two components: a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$ represent the features. Here we define a *domain $\hat{D}$* of information spreading in network G contains a cascade space $\mathcal{C}_{\mathcal{G}}$ and also a marginal probability distribution $P(\mathbb{C}^{\mathbb{G}})$, where $\mathbb{C}^{\mathbb{G}} = \{c_1^G, ..., c_n^G\} \in \mathcal{C}^{\mathcal{G}}$. We will introduce how to compute $P(\mathbb{C}^{\mathbb{G}})$ later. Based on above definitions and terminologies, we formally define the studied problem as follows.

Problem 2. *Given the source domain diffusion network $G^s = (V^s, E^s)$ and the target domain diffusion network $G_{\mathcal{H}}^t = (V^t, E_{\mathcal{H}}^t)$ with corresponding cascades $\mathbb{C}^s \in \mathcal{C}^s$, $\mathbb{C}^t \in \mathcal{C}^t$, where the edges E^s of network G^s is known and the edges $E_{\mathcal{H}}^t$ of network $G_{\mathcal{H}}^t$ is hidden, the problem is how to transfer knowledge from G_s and $\mathbb{C}_s$ and incorporate it with $\mathbb{C}_t$ to better infer the edges $E_{\mathcal{H}}^t$ of $G_{\mathcal{H}}^t$.*

3 Methodology

In this section, we will first revisit some basic concepts and introduce some standard notations. Then we will introduce how to transform the network inference problem to a link prediction task, and how to apply transfer learning techniques to help predict links in the target diffusion network. Next, we will propose a unified scheme to incorporate the generative model on the target diffusion network and the knowledge transferred from the source domain network.

Before introducing the approach, we first give some basic concepts which are essential to model information diffusion. We define a nonnegative random variable T to be the time when an event happens, such as $user_i$ adopting a piece of information. Let $f(t)$ be the probability density function of T, then the cumulative density function can be denoted as $F(t) = P(T \leq t) = \int_0^t f(x)dx$.

Survival Function. The survival function $S(t)$ is the probability that a cascade $\mathbf{t}^c$ does not infect a node by time t:

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

Hazard Function. Given functions $f(t)$ and $S(t)$, we can further define the *hazard function* $H(t)$, which represents the instantaneous rate that a cascade $\mathbf{t}^c$ infects a particular uninfected node within a small interval just after time t.

$$H(t) = \lim_{\Delta t \to 0} \frac{p(t \leq T \leq t + \Delta t | T \geq t)}{\Delta t} = \frac{f(t)}{S(t)}.$$

3.1 Network Inference Based on Generative Model

We define $g(\Delta_{ij}^c; \alpha_{ij})$ as the conditional likelihood of information transmission between node i and node j, where $\Delta_{ij}^c = t_j^c - t_i^c$ is the difference between the infecting time of the two nodes in cascade c and α_{ij} is the transmission rate from node i to j. Here we assume that within a cascade $\mathbf{t}^c$, a node j with a time stamp t_j^c can only be infected by the node i with an earlier time stamp, i.e. $t_j^c < t_i^c$. If $t_j^c > t_i^c$, we can refer node j as one of node i's child node and node i as one of node j's parent node.

Our goal is to infer the pair-wise transmission rate α_{ij}, and we consider that there exists an edge between two nodes if their transmission rate is larger than zero. Three models are used in most previous works to model the diffusion likelihood function $g(\Delta_{ij}^c; \alpha_{ij})$: Exponential model, Power law model, and Rayleigh model [21]. For brevity, we omit the description of the three models.

Likelihood of Node i Infecting j in Cascade $\mathbf{t}^c$. In a cascade, we assume 1) one node gets infected once the first parent infects it, and 2) all the parents infect their child nodes independently. Based on the two assumptions, the likelihood of the parent node i infecting the child node j in cascade $\mathbf{t}^c$ can be computed by

$$g(\Delta t_{ij}^c; \alpha_{ij}) \times \prod_{u \neq i, t_u^c < t_j^c} S(\Delta t_{uj}^c; \alpha_{uj}), \tag{1}$$

where $S(\Delta_{uj}^c; \alpha_{uj})$ is the survival function described before to denote the probability that node j has not been infected by node u before t_j^c under pairwise transmission rate α_{uj}^c between nodes u and j. In the cascade $\mathbf{t}^c$, the node j could be possibly infected by any one of its parent nodes. Hence the likelihood of j getting infected in the cascade $\mathbf{t}^c$ can be calculated by summing over the likelihoods of each potential parent being the first one to infect it:

$$\Gamma_j^+(\mathbf{t}^c) = \sum_{i: t_i^c < t_j^c} g(\Delta t_{ij}^c; \alpha_{ij}) \times \prod_{u \neq i, t_u^c < t_i^c} S(\Delta t_{uj}^c; \alpha_{uj}). \tag{2}$$

Likelihood of a Node j Survives from the Cascade $\mathbf{t}^c$. If node j survives from all the parents by the end time T^c of cascade $\mathbf{t}^c$, we say the node survives from the cascade $\mathbf{t}^c$. The likelihood that node j survives from the cascade $\mathbf{t}^c$ can be represented by the following product of survival function

$$\Gamma_j^-(\mathbf{t}^c) = \prod_{t_i^c < T^c} S(T^c - t_i^c; \alpha_{ij}). \tag{3}$$

Likelihood of the Cascade $\mathbf{t}^c$. Given a cascade $\mathbf{t}^c := (t_1^c, ..., t_N^c)$, its likelihood can be computed by multiplying the likelihoods of all the infected and survived nodes in the cascade. With Eq. (2), Eq. (3), and the hazard function $H(\Delta t_{ij}^c; \alpha_{ij}) = \frac{g(\Delta t_{ij}^c;\alpha_{ij})}{S(\Delta t_{ij}^c;\alpha_{ij})}$, the likelihood of cascade $\mathbf{t}^c$ can be represented as

$$\begin{aligned} g(\mathbf{t}^c; \mathbf{A}) &= \prod_{t_j^c < T^c} \Gamma_j^+(\mathbf{t}^c) \times \prod_{t_j^c > T^c} \Gamma_j^-(\mathbf{t}^c) \\ &= \prod_{t_j^c < T^c} \prod_{t_m^c > T^c} S(T^c - t_j^c; \alpha_{jm}) \times \\ & \prod_{u: t_u^c < t_j^c} S(\Delta t_{uj}^c; \alpha_{uj}) \sum_{i: t_i^c < t_j^c} H(\Delta t_{ij}^c; \alpha_{ij}), \end{aligned} \tag{4}$$

where $\mathbf{A}$ is a $N \times N$ matrix with each element $\mathbf{A}_{ij} = \alpha_{ij}$ denoting the link strength between node i and j.

Assuming the cascades spread independently in the network, the likelihood of a set of cascades $\mathbb{C} = \{\mathbf{t}^1, ..., \mathbf{t}^M\}$ can be represented as the product of the likelihoods of all the individual cascades,

$$\prod_{\mathbf{t}^c \in \mathbb{C}} g(\mathbf{t}^c; \mathbf{A}). \tag{5}$$

Network Inference Problem. The goal is to find the matrix $\mathbf{A}$ such that the network G with edge matrix $\mathbf{A}$ generates cascades $\mathbb{C}$ with the maximum likelihood. This can be achieved by solving the following optimization problem

$$\begin{aligned} & min_{\mathbf{A}} - \sum_{c \in C} \log g(\mathbf{t}^c; \mathbf{A}). \\ & \text{s. t.} \quad \alpha_{ij}^k \geq 0; i, j = 1, ..., N, i \neq j \end{aligned} \tag{6}$$

3.2 Link Prediction in Diffusion Network with Structure Transfer

Fig.1. illustrates the framework of the proposed structure transfer scheme. The left part shows the source domain diffusion network with observed network structure and a large number of cascades. The right part is the target domain diffusion

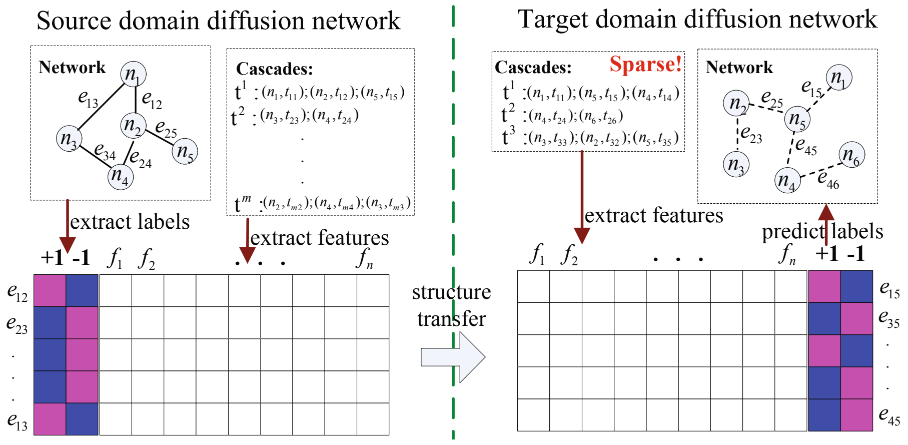


Fig. 1. An illustration of the proposed structure transfer scheme

network with sparse cascades and hidden network structure. In the network, n_i denotes node i, and e_{ij} denotes the edge between node i and j. In a cascade $\mathbf{t}^i$, we use (n_j, t_{ij}) to denote node j is infected at time t_{ij}. Given the two domain diffusion networks, our goal is to borrow the structure and cascade information of the source domain to help infer the network structure in the target domain. To this aim, we formulate it as a link prediction task. Specifically, we first extract features $\{f_1, f_2..., f_n\}$ from the cascades and extract link labels $l_{ij} \in \{+1, -1\}$ in the source domain network. $l_{ij} = +1$ means there exist an edge between node i and j, and otherwise $l_{ij} = -1$. In such a way we extract training samples from the source domain. Then we apply transfer learning technique to select training samples and use them to help predict the link labels in the target domain. Based on the brief description of the framework, next we will elaborate this scheme.

Traditionally, link prediction can be considered as a supervised classification task by constructing a set of features, such as neighborhood based features and path based features [4,10,11]. Motivated by this, we also formulate the network inference problem as a supervised classification problem since the links in the source domain network are known. However, the challenge is that the links in the target domain network are hidden and we cannot construct the features used in traditional link prediction setting. Alternatively, we can extract features from cascades. For example, if node i and j have never appeared in a cascade simultaneously, we can infer that there is probably no link between them; and if node i is the root node of a cascade with node j as the first child node, we can infer that there is definitely a link from i to j. In all we extract 16 cascade related features whose detailed descriptions are given in Table 1.

With the extracted features, we next utilize a popular transfer learning algorithm TrAdaBoost [9] to leverage the links of the source domain network to help us predict the links in the target domain network. TrAdaBoost is a transfer learning framework extended from AdaBoost. Given the limited number of training instances $\mathcal{M}_T^l$ and some test instances $\mathcal{T}$ in the target domain, TrAdaBoost aims to utilize the large volume of available labeled training instances $\mathcal{M}_S$ in the source domain to build a model $f : X \rightarrow Y$ such that the prediction error on $\mathcal{T}$ is minimized. Formally, let X_S be the instances in the source domain network, X_T be the instances in the target domain network, and $Y = \{-1, +1\}$ be the set of labels. Given the source domain network G_S whose edges are known and the target domain network G_T whose edges are hidden, we first assume that their label distribution is the same $\mathcal{Y}_S = \mathcal{Y}_T$, but the feature distribution is different $P_S(y|x) \neq P_T(y|x)$. To utilize TrAdaBoost, we further assume that a small number of labels of the instances X_T^l in the target domain network G_T is given. Therefore, the training data set $\mathcal{M} \subseteq \{X \times Y\}$ includes two parts: $\mathcal{M}_S$, and $\mathcal{M}_T^l$. $\mathcal{M}_S$ represents the source domain network data that $\mathcal{M}_S = \{(x_i^S, y_i^S)\}$, where $x_i^S \in X_S (i = 1, ..., n)$. $\mathcal{M}_T^l$ represents a small number of training data $\mathcal{M}_T^l = \{(x_j^T, y_j^T)\}$ in the target domain network, where $x_j^T \in X_T (j = 1, ..., m)$. n and m are the sizes of $\mathcal{M}_S$ and $\mathcal{M}_T^l$, respectively. By applying TrAdaBoost, we can finally obtain a label matrix $\mathbf{L}$ with $l_{ij} \in \{-1, +1\}$ denoting whether there exists a link from node i to j in the target domain network G_T.

Table 1. Cascade related features for structure transfer

feature	description
f_1	whether node i and j appear in at least one cascade simultaneously, and $t_i < t_j$
f_2	whether there exists a cascade with node i as the root node and node j as its first child node
f_3	the relative frequency of node i appearing before node j in all the cascades
f_4	the minimum time lag $min\Delta t_{ij}^c$ between node i and j in all the cascades
f_5	the average time lag $ave\Delta t_{ij}^c$ between node i and j in all the cascades
f_{6-8}	the maximum probability $maxf(\Delta_{ij}^c; 1)$ of node i infecting node j in all the cascades with three models
f_{9-11}	the average probability $avef(\Delta_{ij}^c; 1)$ of node i infecting node j in all the cascades with three models
f_{12}	for all the cascades that node i is before j, the minimum number of nodes $minN_{ij}^c$ between i and j
f_{13}	for all the cascades that node i is before j, the average number of nodes $aveN_{ij}^c$ between i and j
f_{14}	for all the cascades that node i is before j, the minimum number of nodes $minN_{ri}^c$ between root node r and i
f_{15}	for all the cascades that node i is before j, the minimum number of nodes $minN_{rj}^c$ between root node r and j
f_{16}	for all the cascades that node i is before j, the minimum sum of nodes $min(N_{ri}^c + N_{rj}^c)$ between root node r and i, j

3.3 TrNetInf: Network Inference Incorporating Structure Transfer

By solving the generative model in Eq. (6), we can infer a network matrix $\mathbf{A}$; while by structure transfer with TrAdaBoost, we can obtain a label matrix $\mathbf{L}$. In this section, we will describe how to combine the two parts.

Both methods can infer the connectivity of the target network independently, but the knowledge they used coming from different domains. The generative model only uses the cascades in the target network, and the link prediction based approach mainly relies on the structure knowledge transferred from the source domain network. The results of the two methods may be quite different, and their overlapping part is more likely to be accurate. Thus besides maximizing the probability of generating all the cascades in the target domain, we also want to minimize the difference between the inferred network links by the generative model and the predicted links by diffusion network transfer. We propose to achieve the two goals simultaneously by solving such an optimization problem

$$\begin{gathered} min_{\mathbf{A}} \; -\sum_{c \in \mathbb{C}} \log g(\mathbf{t}^c; \mathbf{A}) + \gamma ||\mathbf{L} - \mathbf{A}||_2, \\ \text{s. t. } \alpha_{ij}^k \geq 0; \; l_{ij} = \{0,1\}; \; i,j = 1,...,N \text{ and } i \neq j \end{gathered} \tag{7}$$

where $\mathbf{A} = \{\alpha_{ij} | i,j = 1,...,N, i \neq j\}$ are the variables and $\mathbf{L} = \{l_{ij} | i,j = 1,...,N, i \neq j\}$ contains the link labels from structure transfer.

Eq. (7) contains two parts. The first part computes the likelihood of the inferred network generating all the cascades, and we want it to be as high as possible. The second part incorporates the structure knowledge transferred from the source domain network. We expect the difference between the two results as small as possible by minimizing the L2 norm distance between $\mathbf{L}$ and $\mathbf{A}$. γ is a parameter used to control the importance of knowledge transferred from the source domain network. Smaller γ implies we trust more on the inferred network by generative model, while larger γ means we rely more on the transferred structure knowledge when available cascades are insufficient.

In addition, most networks are sparse in a sense that one node usually is connected to a small number of other nodes [1,20]. In order to encourage a sparse solution, we add a L2 norm penalty term $||\mathbf{A}||_2$. With the penalty term to control the sparsity of the network, we finally have such an optimization problem

$$\begin{gathered} min_{\mathbf{A}} \; -\sum_{c \in \mathbb{C}} \log g(\mathbf{t}^c; \mathbf{A}) + \gamma_1 ||\mathbf{L} - \mathbf{A}||_2 + \gamma_2 ||\mathbf{A}||_2 \\ \text{s. t. } \alpha_{ij}^k \geq 0; \; l_{ij} = \{0,1\}; \; i,j = 1,...,N \text{ and } i \neq j \end{gathered} \tag{8}$$

We have the following theorem to guarantee that the solution to the optimization problem in Eq. (8) is unique and consistent.

Theorem 1. *Given the optimization problem in Eq. (8), the following results hold:*

1. *Given the log-concave survival functions and concave hazard functions, the problem defined by Eq. (8) is strictly convex in* $\boldsymbol{A}$ *[21].*
2. *The optimization problem defined by Eq. (9) is convex for the proposed TrNetInf model with exponential, Rayleigh, and power law distributions.*
3. *The solution to Eq.(8) gives a unique and consistent maximum likelihood estimator.*

Proof Sketch. 1) Manuel et al. have proved that given the log-concave survival functions and hazard functions in the parameters of the pairwise transmission likelihoods by the exponential, power-law, and Rayleigh models, $\sum_{c\in\mathbb{C}} \log g(\mathbf{t}^c; \mathbf{A})$ is strictly convex in $\mathbf{A}$ [21]. 2) Due to the fact that all the norm functions are convex, we can further infer that both $||\mathbf{L}-\mathbf{A}||_2$ and $||\mathbf{A}||_2$ are convex. As the convex function follows from linearity and composition rules, the liner combination of the three convex functions is also a convex function. 3) For a strictly convex function, its global minimum is unique. Based on the criteria for consistency of identification, continuity and compactness defined by Newey and Mcfadden [27], we can further infer that the solutions to Eq.(8) is consistent. Due to space reason, we omit the proof here, and one can refer [21] for more details.

Solving TrNetInf. Since we have proved Eq. (8) is convex and the solution is unique, we can use a regular convex optimization algorithm to solve Eq. (8). Here we use CVX[1], a popular Matlab-based convex optimization package to solve this problem. We run the algorithm on a Dell PowerEdge T620 server with 32 cores Intel(R) Xeon(R) CPU E5-2670 2.60 GHz, and 64 GB main memory, running the Ubuntu 13.04 operating system.

4 Experimental Results

In this section we conduct a systematic empirical study on real datasets to verify the effectiveness of TrNetInf in inferring diffusion network with sparse cascades. We first introduce the experiment setup, including the used datasets and baselines. Next we give the parameter analysis to show how sensitive the proposed approach is to the parameters γ_1 and γ_2. Then we report the quantitive comparison results with baselines including state-of-the-art methods.

4.1 Experiment Setup

We use two real-world datasets to evaluate TrNetInf: MemeTracker dataset[2] [7] and AMiner citation network dataset[3] [5,8].

MemeTracker Dataset. The MemeTracker dataset contains more than 300 million blog posts and news articles collected from 3.3 million websites. *Memes*

[1] http://cvxr.com/cvx/
[2] http://www.memetracker.org/data.html
[3] http://arnetminer.org/citation

Table 2. Dataset statistics

MemeTracker Datasets			
phrase cluster	# of nodes	# of edges	# of cascades
"good morning America" (Target)	2,754	4,822	425
"put lipstick on a pig" (Source)	2,845	4,621	336
"I'm a mac I'm a pc" (Target)	1,766	2,303	207
"daily show Jon Stewart" (Source)	1,637	2,255	263

AMiner Citation Network Dataset			
research field	# of nodes	# of edges	# of cascades
Computer Theory (Target)	19,073	20,220	832
Graphic (Source)	16,469	21,705	707

are short textual phrases or quotes (like, "good morning America") that spread through the web. Each meme m can be considered as a piece of information, and all the time-stamped webpages which contain meme m forms a diffusion cascade. Memes related to the same topic are considered to be in a same cluster. With the aim of structure transfer, we consider memes in the same cluster coming from the same domain, and memes in different clusters coming from different domains. Given a meme cluster C_m, we first extracted the cascades collection $\mathbb{C}$, and all the websites containing one phrase in C_m as the nodes. For some memes with very long diffusion paths, we split it into several small cascades with length less than 30. The ground truth of the network is constructed by extracting the hyperlinks among all the extracted websites. If a site s_i publishes a phrase and uses a hyperlink to refer to another site s_j that also publishes a similar phrase, we think there exists a link from s_j to s_i.

AMiner Citation Network Dataset. The AMiner citation network dataset contains the citation relationships among papers extracted from DBLP, ACM, and other sources. The citation relationships among papers can be naturally considered as the ground truth of the diffusion network. Similar to MemeTracker dataset, we also consider some term pair phrases (like, "deep learning") extracted from the paper titles and abstracts as the information, and all the papers containing the same phrase can be considered as a cascade. To enable structure transfer, we distinguish the diffusion networks of different domains based on the research fields such as database and computer theory. For example, papers published in the field of database can be considered coming from a domain and those published in computer theory can be consider coming from another domain.

In our experiment, we extract four meme clusters from the MemeTracker dataset forming two groups of datasets for evaluation. For each group of dataset, we use one as the source domain data donated by "Source" and the other as the target domain data denoted by "Target". Similarly, we select the papers published in the venues of two research fields: computer theory and graphic from the AMiner dataset forming another group of dataset. Statistics of the datasets is given in Table 2. We compare TrNetInf with the following baselines.

- **NETRATE**[4] [21]. NETRATE is a representative model to infer both the connectivity of the network and the transmission rates over each edge. As the most relevant work to the proposed model, we choose it as a baseline.
- **NetInf**[5] [23]. Another type of network inference model only aims to infer the network connectivity, such as NetInf. To compare with such kind of methods, we choose the representative approach NetInf as the second baseline.
- **TrNetInf without Sparsity Penalty (TrNetInf-SP).** To study whether and to what extent the sparsity penalty can affect algorithm performance, we use TrNetInf without sparsity penalty as a baseline. For this baseline, we simply set the parameter $\gamma_2 = 0$.
- **TrNetInf without Structure Transfer (TrNetInf-ST)** Similarity, we also use the TrNetInf without structure transfer as a baseline to study how much improvement can be achieved by incorporating structure transfer. In this case, we set the parameter $\gamma_1 = 0$.
- **Link Prediction with Structure Transfer (LPST).** As the proposed TrNetInf combines the information from link prediction model, we use this baseline to study how well the pure link prediction model can perform on the network inference problem and how much achievement can be achieved by TrNetInf. For the LPST baseline, we use TrAdaBoost as the classifier.

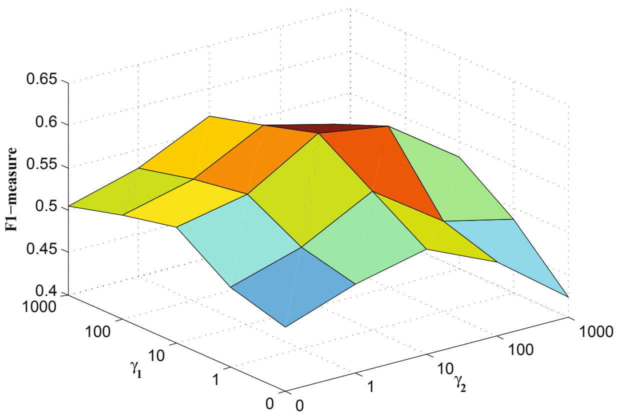


Fig. 2. F1-measure on "good morning America" dataset with various γ_1 and γ_2

4.2 Parameter Analysis

We first study the effect of parameters γ_1 and γ_2 on the performance of TrNetInf. Due to space limitation, we only report the result of the first group of Meme-Tracker dataset. The results of the other datasets are similar.

[4] http://people.tuebingen.mpg.de/manuelgr/netrate/
[5] http://snap.stanford.edu/netinf/

Fig. 2. shows the F1-measure of the "good morning America" dataset with "put lipstick on a pig" as the source domain network over various γ_1 and γ_2. One can see that with the increase of γ_1, the performance first increases, and then decreases, and finally becomes stable. It implies that structure transfer does help our task as the F1-measure are mostly higher than non-transfer with $\gamma_1 = 0$. From $\gamma_1 = 100$ on, the performance tends to be stable, which means the transferred structure knowledge dominates the final results when γ_1 is large. One can also see the F1-measure further increases if we add the sparsity penalty weighted by γ_2, but too large a γ_2 will also hurt the performance. How to choose a proper γ_2 may largely depend on the prior knowledge on the network. A denser network prefers a smaller γ_2, and a larger γ_2 means we may want to infer a less dense network. Fig. 2. suggests that $\gamma_1 = 10$, $\gamma_2 = 10$ seem a good choice of the two parameters for the MemeTracker dataset, and in the following experiments we choose $\gamma_1 = 10$, $\gamma_2 = 10$ as our default parameter settings.

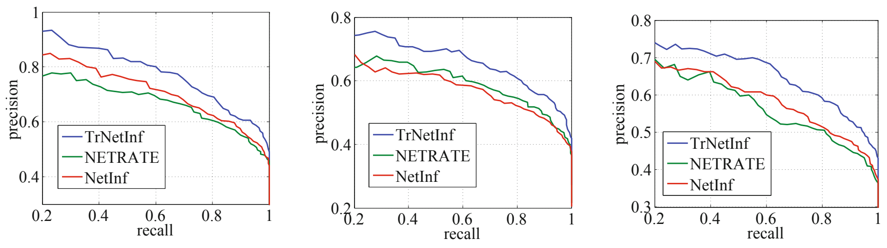
4.3 Quantitive Comparison with Baselines

We quantitively evaluate the performance of TrNetInf via three measures: precision, recall, and F1-measure. We first study the effectiveness of TrNetInf with insufficient cascades by comparing with two state-of-the-art network inference approaches NETRATE and NetInf. To utilize TrAdaBoost for knowledge transfer, some link labels in the target domain network need to be available. In our experiment, we assume 1% links in the target domain network are given.

Comparison Against Network Inference Models. Fig. 3. shows the precision-recall curves of three approaches: NETRATE, NetInf, and TrNetInf over the three datasets. One can see that TrNetInf outperforms NETRATE and NefInf on the three datasets in terms of precision-recall. It implies that the performance can be improved if the structure knowledge is properly transferred. The result also shows that the AMiner dataset seems easier to infer than the two MemeTracker datasets.

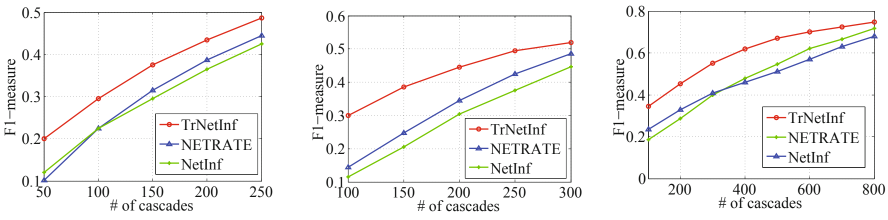
Evaluation with Sparse Cascade Data. To study the effectiveness of TrNetInf with insufficient cascades, we compare the F1-measure achieved by TrNetInf against NETRATE and NetInf by sampling different numbers of cascades in the target domain network. Fig. 4. shows the F1-measures of the three approaches over various numbers of cascades. One can observe that TrNetInf achieves significantly higher F1-measure than NETRATE and NetInf when the number of cascades is relatively small. With the increase of the number of cascades, the performance of the three methods tends to be similar. It implies that structure transfer is especially helpful when the cascade data are very sparse. The performance of the two baselines becomes closer to TrNetInf when more and more cascades are available. It implies that the improvement by structure transfer becomes less significant when a large volume of cascades are available.

Comparison Against Two Variations and Link Prediction Models. Next, we conduct experiment to study whether transfer learning and sparsity



(a) I'm a mac I'm a pc (b) good morning America (c) Computer theory

Fig. 3. The precision-recall curves of the three approaches on three groups of datasets



(a) I'm a mac I'm a pc (b) good morning America (c) Computer theory

Fig. 4. The F1-measure of the three approaches with various numbers of cascades

penalty can both help the network inference task. To this aim, we compare TrNetInf with two variations: TrNetInf without sparsity penalty (TrNetInf-SP) and TrNetInf without structure transfer (TrNetInf-ST). We report precision, recall, and F1-measure for each method on each dataset in Table 3. The figures in bold show the best results. One can see that TrNetInfer is consistently better than TrNetInf-SP and TrNetInf-ST. On average, the F1-measure has improved by about 4% compared with TrNetInf-SP on the three groups of datasets. Compared with TrNetInf-ST, the improvement is more significant, more than 13%. The result leads us to conclude that 1) sparsity penalty do help the studied task, and 2) transfer learning can significantly improve the performance. We also report the performance of link prediction with structure transfer model LPST. One can see that although slightly worth than TrNetInf-ST, LPST model still

Table 3. Experimental result by comparing TrNetInf against two variations and LPST

Method	"I'm a mac I'm a pc"			"good morning America"			Computer theory		
	precision	recall	F1	precision	recall	F1	precision	recall	F1
TrNetInf	0.575	**0.611**	**0.593**	**0.621**	**0.635**	**0.628**	**0.651**	0.700	**0.675**
TrNetInf-SP	0.557	0.598	0.576	0.601	0.598	0.600	0.622	0.657	0.640
TrNetInf-ST	0.534	0.515	0.524	0.546	0.526	0.536	0.540	**0.704**	0.611
LPST	**0.579**	0.379	0.458	0.515	0.534	0.524	0.534	0.598	0.564

gives rather good prediction results. It means that properly structure transfer can provide us useful information for better inferring the diffusion network.

5 Related Work

The problem of inferring the diffusion networks and estimating the diffusion probabilities has been extensively studied in many domains, such as the hyperlink network of on-line new articles [21–23], the coloration network of scientist [20], and the following network in social media [2,17,28]. Previous related works on this topic can be roughly divided into inferring the network structure [23] and inferring both the network structure and the transmission rates between nodes [21]. The representative work on inferring the network structure is NetInf [23]. NetInf formulates this problem as a submodular function maximization problem. NETRATE is a representative approach to infer the diffusion network through estimating the pairwise transmission rates between two nodes. Based on the general inference models, some fine-grained models are proposed. [17] and [19] studied the topic-level diffusion network inference problem.

A related research topic to the network inference problem is link prediction. Link prediction aims to predict the likelihood of a future association between nodes, knowing that there is no association between the nodes in the current state of the graph [4,11]. One of the earliest link prediction models is proposed by Liben-Nowell and Kleinberg [29]. Their proposed approach typically extracts the similarity between a pair of vet ices by various graph-based similarity metrics. Then they use the ranking on the similarity scores to predict the link between two vertices. Besides similarity ranking based approach, another popular approach is to model the link prediction problem as a supervised classification problem [4,10,11]. Such methods normally learn a prediction model by constructing a set of features, such as neighborhood based features [4] and path based features [10]. The main difference between link prediction and network inference is that link prediction aims to predict the future potential connections between nodes based on their current states. In the network inference setting, the network structure is totally hidden and needs to be inferred from traces of information diffusion.

6 Conclusion

To address the problem that traditional inference models may not be effective when lacking enough cascade data, in this paper we proposed a structure transfer scheme to infer the diffusion network with the help of an external diffusion network. We first formulated the network inference problem as a link prediction task by extracting cascade related features. This formulation thus enabled us effectively transfer the cascades and links of the external diffusion network to help predict the hidden links of the target domain network. We also proposed a unified optimization framework to integrate the traditional generative model and the proposed transfer learning model. Evaluations on two real-world datasets demonstrated the effectiveness of the proposed scheme.

In the future, we are particularly interested in further investigating: 1) How to extend one source domain to many source domains. Currently we only consider one source domain diffusion network, but multiple source domains may be more helpful as more information are available [30]. 2) Given multiple source domain diffusion networks, how to select the source domains that are most relevant to the target domain. Currently we only use the domain data which are highly relevant to the target domain. A domain that are irrelevant may also hurt the performance. Source domain diffusion network selection is an interesting and challenging research issue we will focus on in the future.

Acknowledgments. This work is supported in part by NSFC (Nos. 61170189, 61370126, 61202239), the Fund of the State Key Laboratory of Software Development Environment (No. SKLSDE- 2013ZX-19), the Fundamental Research Funds for the Central Universities (YWF-14-JSJXY-16), the Innovation Foundation of BUAA for PhD Graduates (No. YWF-14-YJSY-021), Microsoft Research Asia Fund (No. FY14-RES-OPP-105), NSF through grants CNS-1115234, and OISE-1129076.

References

1. Gomez-Rodriguez, M., Leskovec, J., Scholkopf, B.: Modeling information propagation with survival theory. In: ICML (2013)
2. Wang, S.Z., Yan, Z., Hu, X., Yu, P.S., Li, Z.J.: Burst time prediction in cascades. In: AAAI (2015)
3. Wang, L., Ermon, S., Hopcroft, J.E.: Feature-enhanced probabilistic models for diffusion network inference. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 499–514. Springer, Heidelberg (2012)
4. Lu, L.Y., Zhou, T.: Link Prediction in Complex Networks: A Survey. Physica A: Statistical Mechanics and its Applications **390**(6), 1150–1170 (2011)
5. Tang, J., Zhang, D., Yao, L.M.: Social network extraction of academic researchers. In: ICDM (2007)
6. Zhang, J.W., Yu, P.S., Zhou, Z.H.: Meta-path based multi-network collective link prediction. In: KDD (2014)
7. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: KDD (2009)
8. Tang, J., Zhang, J., Yao, L.M., Li, J.Z., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: KDD (2008)
9. Dai, W.Y., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: ICML (2009)
10. Hasan, M.A., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: SDM (2006)
11. Lichtenwalter, R.N., Lussier, J.T., Chawla, N.V.: New perspective and methods in link prediction. In: KDD (2010)
12. Jiang, W., Chung, F.: Transfer spectral clustering. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 789–803. Springer, Heidelberg (2012)
13. Pardoe, D., Stone, P.: Boosting for regression transfer. In: ICML (2010)
14. Zhu, Y., Chen, Y.Q., Lu, Z.Q., Pan, S.J., Xue, G.R., Yu, Y., Yang, Q.: Heterogeneous transfer learning for image classification. In: AAAI (2011)

15. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Trans. on Knowl. and Data Eng. **22**(10), 1345–1359 (2010)
16. Herlihy, M.: Diffusion in Organizations and Social Movements: From Hybrid Corn to Poison Pills. Annual Review of Sociology **24**, 265–290 (1998)
17. Wang, S.Z., Hu, X., Yu, P.S., Li, Z.J.: MMRate: inferring multi-aspect diffusion networks with multi-pattern cascades. In: KDD (2014)
18. Erdman, D.D.: Propagation and Identification of Viruses. Topley and Wilson's Microblology and Microblal Infections (2010)
19. Du, N., Song L., Woo, H., Zha, H.Y.: Uncover topic-sensitive information diffusion networks. In: AISTATS (2013)
20. Myers, S.A., Leskovec, J.: On the convexity of latent social network inference. In: NIPS (2010)
21. Gomez-Rodriguez, M., Balduzzi, D., Scholkopf, B.: Uncovering the temporal dynamics of diffusion networks. In: ICML (2011)
22. Gomez-Rodriguez, M., Leskovec, J., Scholkopf, B.: Structure and dynamics of information pathways in online media. In: WSDM (2013)
23. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: KDD (2010)
24. Leskovec, J., Singh, A., Kleinberg, J.M.: Patterns of influence in a recommendation network. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 380–389. Springer, Heidelberg (2006)
25. Chen, W., Wang, C., Wang, Y.J.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: KDD (2010)
26. Kempe, D., Kleinberg, J., Tardos, E.:Maximizing the spread of influence through a social network. In: KDD (2003)
27. Newey, W.K., McFadden, D.: Large sample estimation and hypothesis testing. In: Handbook of Econometrics, pp. 2111–2245 (1994)
28. Wang, L., Ermon, S., Hopcroft, J.E.: Feature-enhanced probabilistic models for diffusion network inference. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 499–514. Springer, Heidelberg (2012)
29. Liben-Nowell, D., Kleinberg, J.: The Link Prediction Problem for Social Networks. Journal of the American Society for Information Science and Technology **58**(7), 1019–1031 (2007)
30. Chen, Z.Y., Liu, B.: Topic modeling using topics form many domains, lifelong learning and big data. In: ICML (2014)