NETWORK INFERENCE

R V Pavan Kumar
November 3, 2015
• Challenges to study network diffusion
  • Identify the contagion and trace it
  • Identify the underlying network

• Usually network of propagation is unobserved

• Examples: Information propagation in blogs, virus/disease propagation, viral marketing

• Understanding this diffusion is necessary for stopping infections, predicting meme propagation or maximizing sales of a product.

• Inferring the underlying propagation network from set of observed cascades
PROBLEM STATEMENT
• Given a set of cascades \( C = \{ c_1, c_2, \cdots, c_m \} \) infer the static hidden directed propagation network \( G^* \)
• The trace of each cascade can be represented by \((u_i, t_i, \phi_i)_c\)
where,

\[
\begin{align*}
  u_i & \quad \text{node id/name} \\
  t_i & \quad \text{time at which cascade } c \text{ reached node } u_i \\
  \phi_i & \quad \text{set of features, either from node or contagion}
\end{align*}
\]
• Only \( t_u \)'s \textit{(hit time)} are observed, \( \phi_i \)'s are considered constant
• Hence, \( c = [t_1, t_2, \cdots, t_n] \), where \( n \) is the number of nodes in \( G^* \)
• \( t_u = \infty \) if \( u \) is not reached during the observed time \( T \)
PROBLEM FORMULATION
· **Cascade Transmission Model** ($P_c(u, v)$): probability that cascade $c$ propagated from node $u$ to $v$

· Assume that every node $v$ can be influenced by only one node $u$

· Hence, influence structure of $c$ will be a directed tree $T$

\[
\begin{align*}
P(c|T) & \text{ probability cascade } c \text{ propagated in pattern } T \\
P(c|G) & \text{ probability cascade } c \text{ occurs in } G \\
P(C|G) & \text{ probability cascades } C \text{ occur in } G
\end{align*}
\]

**Maximum Likelihood Estimator**, $\hat{G} = \arg \max_G P(C|G)$
CASCADE TRANSMISSION MODEL

\[ P_c(u, v) = \begin{cases} 
0 & \text{if } t_u \geq t_v \\
1 - \beta & \text{if } t_v = \infty \\
\beta P(\Delta) & \text{otherwise}
\end{cases} \]

- \( \Delta = t_v - t_u \) is time taken to propagate is modelled using
  - **Exponential model** \( P(\Delta) \propto \exp \frac{-\Delta}{\alpha} \)
  - **Power-Law model** \( P(\Delta) \propto \Delta^{-\alpha} \)
- Probability that cascade c propagated in directed tree T
  \[ P(c|T) = \prod_{(i,j) \in T} P_c(i, j) \]
  assuming independence between pairs of nodes
Bruce Logan
The Pennsylvania State University
\[ P(c|G) = \sum_{T \in T(G)} P(c|T)P(T|G) \]
\[ \propto \sum_{T \in T(G)} \prod_{(i,j) \in T} P_c(i,j) \]
where \((G)\) is the set of directed spanning trees on \(G\)
\[ P(T|G) \text{ is a uniform distribution over all } T \]
\[ \text{Probability that set of cascades } C \text{ occur in } G \text{ is given by} \]
\[ P(C|G) = \prod_{c \in C} P(c|G) \]
assume conditional independence between \(c\)’s given \(G\)
\[
\hat{G} = \arg \max_{|G| \leq k} P(C|G)
\]

\[
P(C|G) = \prod_{c \in C} P(c|G) = \prod_{c \in C} \sum_{T \in T(G)} P(c|T)
\]

Computing \( \hat{G} \) is wildly intractable

1. For given \(|G| = n \) (number of nodes) \(|T(G)| \in O(n^l)\), for some constant \( l \), hence computing \( \sum P(c|T) \) over all possible \( T \)’s takes super-exponential time

2. Surprisingly we can mitigate this problem applying Kirchhoff’s matrix tree method. This will improve the execution time of computing \( P(c|G) \) to \( O(n^3) \)

3. Still to find MLE (\( \hat{G} \)) we need to search over all \( G \), which is super-exponential again.
ALTERNATIVE FORMULATION
Consider the most likely propagation tree $T$ per cascade $c$

$$P(C|G) = \prod_{T \in T(G)} \max_{T \in T(G)} P(c|T)$$

Improvement of log-likelihood over empty graph ($\bar{K}$)

$$F_c(G) = \max_{T \in T(G)} \log P(c|T) - \max_{T \in T(\bar{K})} \log P(c|T)$$

External sources influence the propagation in empty graph

Add a new node $m$ to each cascade $c$ such that

$$P_c(m, j) = \epsilon, \; \forall j$$

$$F_c(G) = \max_{T \in T(G)} \sum_{(i,j) \in T} w_c(i, j)$$

where $w_c(i, j) = \log P_c(i, j) - \log \epsilon$
Objective function: \( F_C(G) = \sum_{c \in C} F_c(G) \)

Optimal network: \( G^* = \arg \max_{|G| \leq k} F_C(G) \)

- Most likely propagation tree \( T \) is the maximum weighted directed spanning tree, where each edge \((u, v)\) has weight \( w_c(u, v)\)

**Algorithm:** Select an incoming edge of highest weight for each node, takes \( O(|E|) \) time

- Still solving for \( G^* \) in tractable since there are super-exponential number of possible graphs

- It is **NP-hard**. It has a polynomial time reduction to **MAX-K-Cover** problem
\[ F_C(K) = 0 \]

\[ F_C \text{ is non-negative and monotonic} \]

\[ F_C \text{ satisfies sub-modularity} \]

\[ F_C(G\cup e)F_C(G) \geq F_C(G'\cup e)F_C(G') \]

where \( G \subset G' \) and \( e \notin G' \)

\[ \hat{G} \text{ can be computed using greedy algorithm} \]

\[ \hat{G} \text{ will be within the fraction } (1 - 1/e) \approx 63\% \text{ of } G^* \]

\[ \text{Further more, we can acquire a tight on-line bound as follows} \]

\[ \arg \max_{|G| \leq k} F_C(G) \leq F_C(\hat{G}) + \sum_{i=1}^{k} \delta_{e_i} \]

where \( \delta_{e_i} = F_C(\hat{G} \cup e_i)F_C(\hat{G}) \) and \( e_i \notin \hat{G} \)
PROPOSED ALGORITHM
Algorithm 1 The NETINF Algorithm

Require: $C, k$
$G \leftarrow \overline{K}$;
for all $c \in C$ do
    $T_c \leftarrow \text{dagtree}(c)$;
while $|G| < k$ do
    for all $(j, i) \in C \setminus G$ do
        $\delta_{j, i} = 0$, $M_{j, i} \leftarrow \emptyset$;
        for all $c : (j, i) \in c$ do
            let $w_c(m, n)$ be the weight of $(m, n)$ in $G \cup \{(j, i)\}$;
            if $w_c(j, i) \geq w_c(\text{Par}_{T_c}(i), i)$ then
                $\delta_{j, i} = \delta_{j, i} + w_c(j, i) - w_c(\text{Par}_{T_c}(i), i)$;
                $M_{j, i} \leftarrow M_{j, i} \cup \{c\}$;
                $(j^*, i^*) \leftarrow \text{arg max}_{(j, i) \in C \setminus G} \delta_{j, i}$;
            $G \leftarrow G \cup \{(j^*, i^*)\}$;
        for all $c \in M_{j^*, i^*}$ do
            $\text{Par}_{T_c}(i^*) \leftarrow j^*$;
    return $G$;
EXPERIMENTS
Models

1. Forest Fire model
   - 1000 nodes, 1477 edges, 834 cascades

2. Kronecker graphs model
   - Random graphs \([0.5, 0.5; 0.5, 0.5]\]
   - Hierarchical community structure \([0.962, 0.107; 0.107, 0.962]\]
   - Core-periphery network \([0.962, 0.535; 0.535, 0.107]\]

Cascades

- Selecting a starting node uniformly
- Simulated for both exponential and power-law diffusion models
- 99% of the edges participate in at-least one cascade
• Compute $w(u, v)$ for each possible edge $(u, v)$, where $w(u, v) = \sum_{c \in C} P_c(u, v)$
• Pick $k$ edges which have highest weight $w(u, v)$
EVALUATION
- NETINF find 97% of the optimal network
- Sparse solutions achieve high objective values
Precision-Recall curves

**Precision** what fraction of edges in $\hat{G}_k$ is also present in $G^*$

**Recall** what fraction of edges of $G^*$ appears in $\hat{G}_k$

**Break-even point** point at which recall is equal to precision

Plotted by varying $k$, ($1 \leq k \leq n^2$)

**Keonecker graphs**

- Network with 1024 nodes
- Cascades generated with following transmission models
  - Exponential model with $\alpha = 1$
  - Power-law model with $\alpha = 2$
    - Hierarchical with $\beta = 0.5$
    - Random with $\beta = 0.4$
    - Core-periphery with $\beta = 0.1$
ACCURACY (CONT)

- Baseline method achieved break-even point between 0.4 and 0.5
- NETINF achieved break-even point at 0.99
· Performance of baseline drops dramatically due to high variance of power-law
· NETINF remains stable with break-even point at 0.99
Forest Fire graph

- Network with 1000 nodes
- Cascades generated using power-law transmission model with $\alpha = 1.1$ and $\alpha = 3$

- Performance of baseline is very low where as NETINF achieved break-even point at $0.9$
- NETINF does not depend on the structure of the network
NETINF requires total number of transmission events between 2 to 5 times of number of edges in $G^*$
• Hierarchical Kronecker graph and exponential transmission model ($\alpha = 1$, $\beta = 0.5$)
• NETINF achieved two orders (280x) of magnitude speed up using Localized update (45x) and Lazy evaluation (6x) with of loss in solution quality
Created two datasets from 172 million news articles and blog posts from one million on-line sources over a period of one year.

**Blog hyperlink cascades dataset**

- Hyperlinks between blogs to trace flow of information
- A cascade consists of time-stamps of the hyperlink/post-creation times

**MemeTracker dataset**

- Extracted short textual distinct phrases using Memetracker methodology
- Cascades are the clusters formed by aggregating different textual variants of same phrase
- Largest 5000 cascades by recording when a particular phrase is mentioned by a website.
ACCURACY

Ground Truth

- Network with directed edge \((u,v)\) if site \(u\) is hyperlinked to site \(v\)
- Using top 500 sites in terms of number of hyperlinks as nodes and top 4000 hyperlinks in terms of number of posts as edges

Blog hyperlink cascades dataset

Break-even point: Baseline method - 0.34, NETINF - 0.44
MemeTracker dataset

- Baseline method achieved break-even point at 0.17 while NETINF achieved 0.28
- NETINF can potentially gain additional accuracy with more realistic transmission model
NETINF finds the solution that is least \textbf{84\%} of the optimal

Objective function quickly flattens out which means one needs relatively few edges to capture most of the information
**Figure:** Largest component after choosing 100 edges while inferring a network of top 1000 media sites and blogs with largest number of documents
**Figure:** Small part of network inferred using Memetracker dataset
Most of the information flows from Media-to-Blogs.
Media-to-Media are the strongest and NETINF picks them early.
Blog-to-Media are the rarest and weakest.
Media sites are quickest to infect one another or by blogs.
Blogs tend to be much slower in propagating information.
CONTINUOUS-TIME DIFFUSION MODEL
For $N$ fixed nodes and set of cascades $C = t^1, t^2, \cdots, t^{|C|}$ are observed.

Each cascade $t^c$ is an $N$-dimensional vector given by $(t^c_1, t^c_2, \cdots, t^c_N)$, $t^c_k \in [0, T^c] \cup \infty$.

$t^c_k = \infty$ implies that $k^{th}$ node is not infected during the observation window.

For Simplicity, consider $T^c = T \forall c \in C$.

Cascades induce a directed acyclic graph (DAG) structure on the network i.e. node $i$ is parent of $j$ if $t_i < t_j$. 
Diffusion can occur at different rates $\alpha_{j,i}$ over different edges of a network

$$f(t_i|t_j, \alpha_{j,i})$$ Conditional likelihood of transmission between a node $j$ and node $i$, where $\alpha_{j,i}$ is the transmission rate of pair $(j,i)$

$$F(t_i|t_j; \alpha_{j,i})$$ Cumulative density function computed on $f(t_i|t_j, \alpha_{j,i})$

$$S(t_i|t_j; \alpha_{j,i})$$ Survival function of edge $(j, i)$ i.e. the probability that node $i$ is not infected by node $j$ by time $t_i$

$$S(t_i|t_j; \alpha_{j,i}) = 1 - F(t_i|t_j; \alpha_{j,i})$$

$$H(t_i|t_j; \alpha_{j,i})$$ Hazard function or instantaneous infection rate of edge $(j,i)$

$$H(t_i|t_j; \alpha_{j,i}) = \frac{f(t_i|t_j; \alpha_{j,i})}{S(t_i|t_j; \alpha_{j,i})}$$
Likelihood of a Cascade

Consider a cascade $t = (t_1, t_2, \cdots, t_N)$, the likelihood of observed infections is given by

$$f(t; A) = \prod_{t_i \leq T} f(t_i|t_1, \cdots, t_N \setminus t_i; A)$$

where infections are conditionally independent given the parent.

Given an infected node $i$, the likelihood of a potential parent $j$ to be the first parent is given by

$$f(t_i|t_j; \alpha_{j,i}) \times \prod_{j, t_k < t_i} S(t_i|t_k; \alpha_{k,i})$$
Conditional likelihood of by summing over the likelihoods of each potential parent is the first parent

\[ f(t_i|t_1, \ldots, t_N \setminus t_i; A) = \sum_{j : t_j < t_i} f(t_i|t_j; \alpha_{j,i}) \times \prod_{j \neq k, t_k < t_i} S(t_i|t_k; \alpha_{k,i}) \]

Hence, Likelihood of a cascade is given by

\[ f(t \leq T; A) = \prod_{t_i \leq T} \sum_{j : t_j < t_i} f(t_i|t_j; \alpha_{j,i}) \times \prod_{j \neq k, t_k < t_i} S(t_i|t_k; \alpha_{k,i}) \]

\[ f(t \leq T; A) = \prod_{t_i \leq T} \prod_{k : t_j < t_i} S(t_i|t_j; \alpha_{j,i}) \times \sum_{j : t_j < t_i} \frac{f(t_i|t_j; \alpha_{j,i})}{S(t_i|t_j; \alpha_{j,i})} \]

By adding node that are not infected at all

\[ f(t; A) = \prod_{t_i \leq T} \prod_{t_m > T} S(T|t_i; \alpha_{i,m}) \times \prod_{k : t_j < t_i} S(t_i|t_j; \alpha_{j,i}) \sum_{j : t_j < t_i} H(t_i|t_j; \alpha_{j,i}) \]
NETWORK INFERENCE PROBLEM
Likelihood of set of cascades is given by

\[ f(C; A) = \prod_{t^c \in C} f(t^c; A) \]

The optimal transmission rates \( \alpha_{j,i} \) that maximizes the likelihood of observed cascades \( C \) is given by

\[
\min_A \quad -\frac{1}{n} \sum_{c \in C} \log f(t^c; A) \\
\text{subject to} \quad \alpha_{j,i} \geq 0, \ i, j = 1, \ldots, N, \ i \neq j
\]

where \( A = \{\alpha_{j,i} | i, j = 1, \ldots, n, i \neq j\} \)
This objective function is convex and can be split into independent sub-problems, one per each node $i$, where we solve

$$\min_{\alpha_i} \quad \mathbf{l}_n(\alpha_i)$$

subject to $\alpha_{j,i} \geq 0$, $i, j = 1, \ldots, N$, $i \neq j$ (1)

where $\alpha_i = \{\alpha_{j,i} | j = 1, \ldots, n, i \neq j\}$ and $\mathbf{l}_n(\alpha_i) = -\frac{1}{n} \sum_{c \in C_n} \mathbf{g}_i(t_c; \alpha_i)$

Consider only super-neighborhood $V_i = R_i \cup U_i$ of $i$ because they will never be infected in cascade before $i$
CONSISTENCY
Can we recover the hidden network structures from the observed cascades?

Yes, if Maximum Likelihood estimator $\alpha_i$ provided by equation (1) is consistent we can recover the true network as number of cascades goes to $\infty$.

The Criteria for consistency: continuity, compactness and identification

**Theorem 1**: If the source probability $P(s)$ is strictly positive for all $s \in \mathbb{R}$, then, the MLE $\hat{\alpha}$ given by the solution of equation (1) is consistent.

Under the condition that objective function in equation (1) is convex, using Theorem 1 we can prove that MLE ($\hat{\alpha}$) is consistent.
RECOVERY CONDITIONS
Sufficient conditions on the diffusion model and the cascades sampling process under which we can recover the network structure from finite samples.

**Dependency Condition:** Two connected nodes must co-occur reasonably frequently in the cascades but are not deterministically related.

**Incoherence condition:** For each node $i$ and any of its neighbours should get infected together in a cascade more often than node $i$ and any of its non-neighbours.

**Remark:** As long as $P_0 > 0$ there is always some $\epsilon > 0$ for which the above condition holds, and such $\epsilon$ value depends on the time window and the parameters $\alpha_{0j}$.
Lipschitz Continuity: For any feasible cascade $t^c$, the Hazard vector $X(t^c; \alpha)$ is Lipschitz continuous in the domain $\{\alpha : \alpha_S \geq \alpha_{\text{min}}/2\}$,

$$|X(t^c; \beta) - X(t^c; \alpha)|_2 \leq k_1|\beta - \alpha|_2$$

where $k_1$ is some positive constant

Boundedness: For any feasible cascade $t^c$, the absolute value of each entry in the gradient of its log-likelihood and in the Hazard vector, as evaluated at the true model parameter $\alpha^*$, is bounded,

$$|\Delta g(t^c; \alpha^*)|_{\infty} \leq k_3$$

$$|X(t^c; \alpha^*)|_{\infty} \leq k_4$$

where $k_3$ and $k_4$ are positive constants.

Remark: Well-known pairwise transmission likelihood models such as exponential, Rayleigh or Power-law, used in the previous work satisfy these condition.
How many cascades do we need to recover the network structure?

Given the above conditions we can prove that the number of cascades needs to grow polynomially in the number of true parents of a node, and depends only logarithmically on the size of the network.

**Remark:** If the above recover conditions holds we can recover the network structure using a $l_1$-regularized maximum likelihood estimator and $O(d^3 \log N)$ cascades, such that the probability of the success is approaching 1 at a exponential rate.
EXPERIMENTS
SYNTHETIC DATA

- Two similar to real-world Models: Forest Fire model and Kronecker graph model
- Pair-wise transmission models: Exponential, Power-law, Rayleigh
- Network with 128 nodes and edge transmission rates draw from $U(0.5, 1.5)$
- Accuracy metric
  - $P(\hat{e} = e^*)$, probability of success in inferring neighborhood of a node $i$
  - Compared with state-of-the-art algorithms, NETRATE and First-Edge, by using F1-score

where $F1 = 2PR/(P+R)$, $P$ i Precision and $R$ is Recall
Regularized Parameter

- $\lambda_n = K\sqrt{\log(p)/n}$ for 150 cascades and $T = 5$
- For sufficiently large $\lambda_n$ the success probability flattens out
Number of cascades to successfully infer incoming edges of a node $i$ will increase logarithmically to the node’s super-neighborhood size $p_i$ and increase polynomially to the node’s neighbours size $d_i$.

Very different values of $p$ values lead to curves that line up well.
Comparison with NETRATE and First-Edge

Kronecker hierarchical, POW

Forest Fire, EXP
CONCLUSION
· Investigated problem of tracing paths of diffusion and influence
· Proposed a scalable algorithm, NETINF, to infer propagation networks based on observed cascades
· Evaluated NETINF using both synthetic data and real world networks which lead to interesting insights
· Proposed $l_1$-regularized maximum likelihood inference method for well-known continuous-diffusion model
· An efficient proximal gradient implementation
· Showed that general networks satisfying a natural incoherent condition this method achieves an exponential decreasing error as long as $|C| \in O(d^3 \log N)$
Questions?