

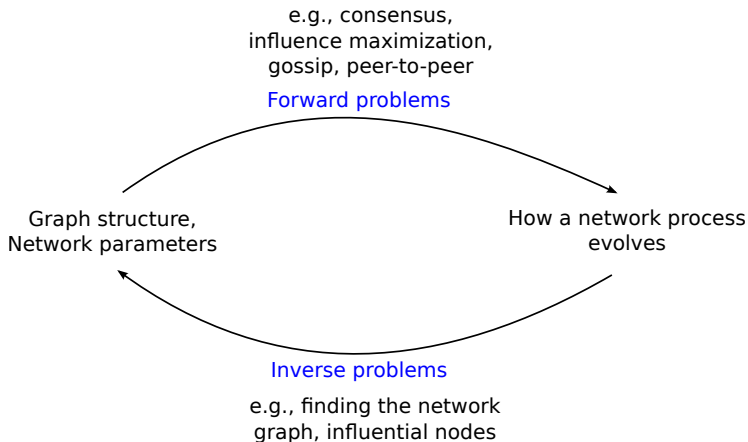
Learning the Graph of Epidemic Cascades

Praneeth Netrapalli
Joint work with Sujay Sanghavi

The University of Texas at Austin

June 13, 2012

Two types of problems



Today: finding the network graph of cascades

Cascades

- Cascades are processes where nodes get infected and further infect other nodes and so on.
- They are used to model the spread of
 - infectious diseases
 - online media such as videos, news stories etc
 - offer uptakes on social buying sites such as Groupon, Living Social etc.

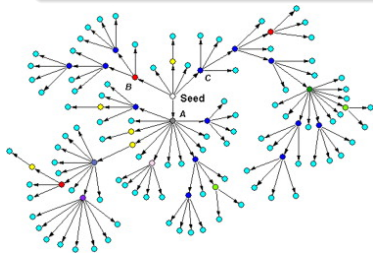
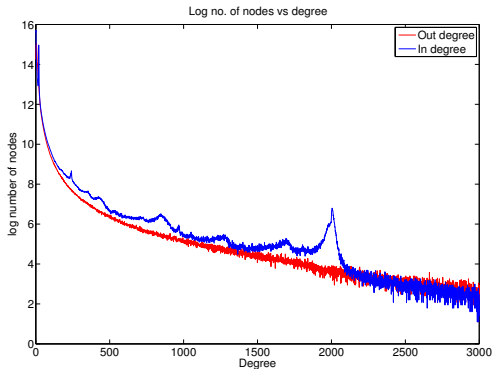


Fig : A tree cascade
Courtesy : Sciencedirect

Motivation for Graph Learning

- Require knowledge of the graph for solving forward problems
- Presence of rubbish edges in online social networks

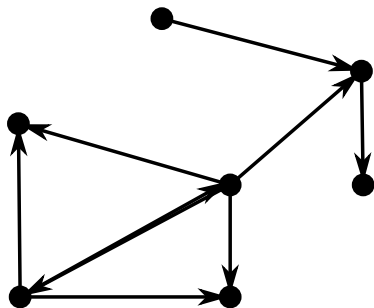


Twitter degree plot
- more than 10 mn
users have degree
 > 100

Overview

- 1 Standard Independent Cascade Model
- 2 Graph Learning Problem
- 3 Algorithms
- 4 Lower Bounds

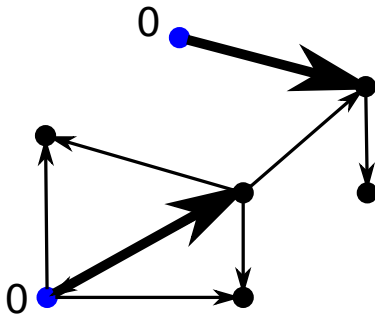
Standard Independent Cascade Model¹



- discrete time
- initial seeds random
- i infects j w.p. p_{ij}
- inactive after one time step

¹ Kempe, Kleinberg and Tardos, 2003.

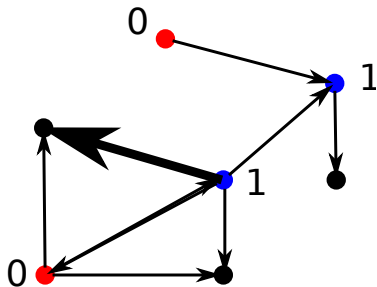
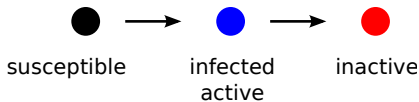
Standard Independent Cascade Model¹



- discrete time
- initial seeds random
- i infects j w.p. p_{ij}
- inactive after one time step

¹ Kempe, Kleinberg and Tardos, 2003.

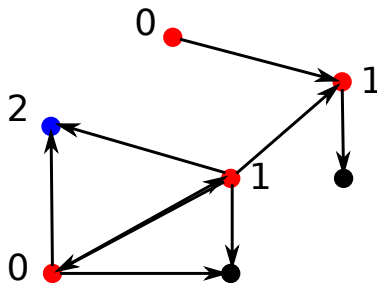
Standard Independent Cascade Model¹



- discrete time
- initial seeds random
- i infects j w.p. p_{ij}
- inactive after one time step

¹ Kempe, Kleinberg and Tardos, 2003.

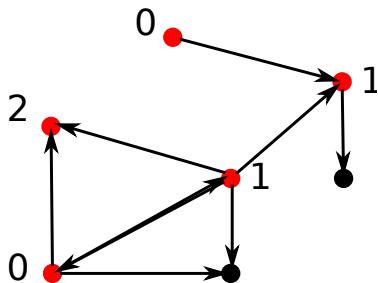
Standard Independent Cascade Model¹



- discrete time
- initial seeds random
- i infects j w.p. p_{ij}
- inactive after one time step

¹ Kempe, Kleinberg and Tardos, 2003.

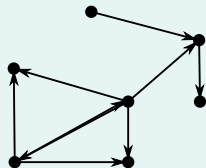
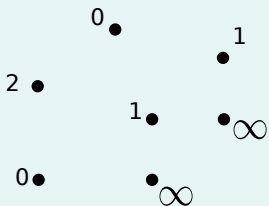
Standard Independent Cascade Model¹



- discrete time
- initial seeds random
- i infects j w.p. p_{ij}
- inactive after one time step

¹ Kempe, Kleinberg and Tardos, 2003.

Graph learning problem



Given node activation times

Find network graph

- clearly not possible with a single cascade
- super graph may be available

Overview of Results

- **Computational** (fast algorithms)
 - show that Maximum likelihood becomes a decoupled convex program \Rightarrow fast distributed implementation
- **Statistical** (how many cascades do we need to obtain the truth?)
 - performance of Maximum likelihood algo.
 - performance of Greedy algo.
 - Information theoretic lower bounds

Maximum likelihood

$$\mathcal{L}(t, p) = \log \mathbb{P}[t|p]$$

infection times edge probabilities

Classical Results

- $\hat{p}(t) := \arg \max_p \mathcal{L}(t, p)$
- Consistency : As $m \rightarrow \infty$ we have $\hat{p}(T) \rightarrow p^*$

However, in our setting

- **Computational**: efficient maximization of likelihood?
- **Statistical**: correctness when $m \ll n$ (high dimensional scaling)?

Maximum likelihood

Our insights into the log likelihood function

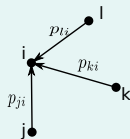
- **Decouples** into *independent* node wise problems

$$\mathcal{L}(t, p) = \sum_i \mathcal{L}_i(t, p_{*i}) \text{ where}$$

$$\mathcal{L}_i(t, p_{*i}) = \log \mathbb{P} [t_i | p_{*i}, t_{\setminus i}]$$

- Becomes **concave** after change of variables

$$\theta_{ij} = -\log(1 - p_{ij})$$



Finding the graph \Rightarrow Solving a convex problem for each node (in parallel)

Maximum likelihood

ML based algorithm for node i

- **Maximize log likelihood:** $\hat{\theta}_{*i} := \arg \max_{\theta_{*i}} \mathcal{L}_i(t, \theta_{*i})$
- **Threshold:** $\hat{\mathcal{N}}_i = \{j \mid \hat{\theta}_{ji} > \eta\}$

So : resolved computational issues

Now : focus on statistical issues - **how many cascades do we need to learn the graph?**

Setting

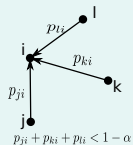
- Each node has probability p_{init} of being a seed
- Each edge is strong enough
- **Correlation decay** \sim far away nodes have negligible effect

Require $\sum_j p_{ji} < 1 - \alpha \forall i$

Distance between seeds and infected nodes not huge

Consequence :

$p_{\text{init}} < \text{Prob of participating in a cascade} < \frac{p_{\text{init}}}{\alpha}$



Main Result

Theorem

For node i with parental degree d_i , if the total number of cascades is

$$m > \frac{c}{p_{\text{init}}} \left(\frac{1}{\alpha^7 \eta^2 p_{i,\text{min}}^2} \right) d_i^2 \log \left(\frac{n}{\delta} \right)$$

then w.p. at least $1 - \delta$, the ML algorithm gives

- 1 no false parents and
- 2 all strong enough parents (i.e., $p_{ji} > \frac{8}{\alpha} (e^{2\eta} - 1)$)

Comments

$$\overbrace{p_{\text{init}} m}^{\sim \text{number of infections}} > c \left(\frac{1}{\alpha^7 \eta^2 p_{i,\text{min}}^2} \right) d_i^2 \log \left(\frac{n}{\delta} \right)$$

threshold min. strength degree network size

- Non asymptotic
- Node by node guarantees
- entire graph recovery via union bounds
- stronger neighbors easier to identify

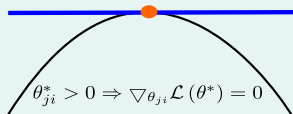
Proof

Main idea

Convex problems \rightarrow first order optimality conditions \rightarrow statistical guarantees

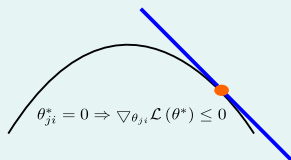
Ideal case

If we had ∞ samples, $\theta^* = \arg \max \mathcal{L}(\theta)$ and so



j is a parent of i

Can infer edges from gradient.



j is NOT a parent of i

Proof

Issue

Finite samples

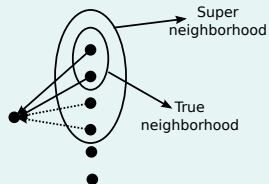
The proof has two steps.

- 1 **Concentration:** $\nabla \hat{L}(\theta^*)$ approximately satisfies the above conditions (via Azuma-Hoeffding)
- 2 **Smoothness & Convex analysis:** $\theta^* \approx \hat{\theta}$

Super Graph

Often, we know a **super-set** of the set of edges.

e.g. Social networks: the entire set of a users online acquaintances
Epidemiology: restrictions based on geography, etc.



This means we have a super set of parents for each node.

ML Algorithm

same as before but
optimize only within super neighborhood
needs only local info

Super Graph

Theorem

For recovery of node i 's parents, need

$$m > \frac{c}{p_{\text{init}}} \left(\frac{1}{\alpha^7 \eta^2 p_{i,\text{min}}^2} \right) d_i^2 \log \left(\frac{D_i}{\delta} \right)$$

Comments

- D_i : size of the super-neighborhood
- Sample complexity independent of n
- Computational complexity reduces as well!

Greedy Algorithm

Recall : j can infect i in only one time step after j is infected.

For node i , start with all cascade samples

Iteratively

Pick the node which gets active **one time-step before i** in the most number of remaining samples

Remove that node and corresponding cascades

Repeat until all cascade samples exhausted

Nodes	i	j	k
Cascade 1	0	1	1
Cascade 2	1	1	0
Cascade 3	2	1	0
Cascade 4	1	0	1
Total	-	2	1

Nodes	i	k
Cascade 1	0	1
Cascade 2	1	0
Total	-	1

Greedy Algorithm

Theorem

If the graph is a tree, just need

$$m > \frac{c}{p_{\text{init}}} \left(\frac{1}{p_{i,\text{min}}} \right) d_i \log \left(\frac{D_i}{\delta} \right)$$

- linear in degree
- correlation decay not needed

Information theoretic lower bounds

Theorem

Set of graphs == all with in-degree at most d .

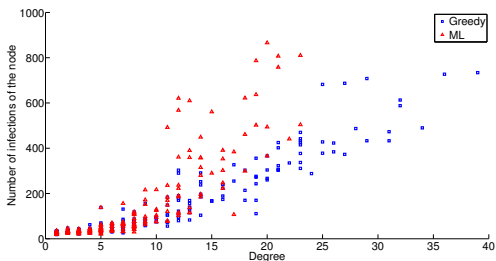
Want: recovery of the graph with prob. at least $1 - \delta$. Then need:

$$m \gtrsim d \log \left(\frac{D}{d} \right)$$

Comments

- Shows ML tight upto a d factor.
- Extended results upto some edit distance γ

Dependence on degree



- twitter sub-graph
- dependence super-linear for ML
- seems linear for greedy though graph is not a tree

Summary

Graph learning possible

- from small no. of cascades
- via fast distributed algorithms
- using local info

Conclusions

Inverse problems: Infer network structure, properties by observing network processes.

- increasingly important
- require new tools and methods

Thanks!

