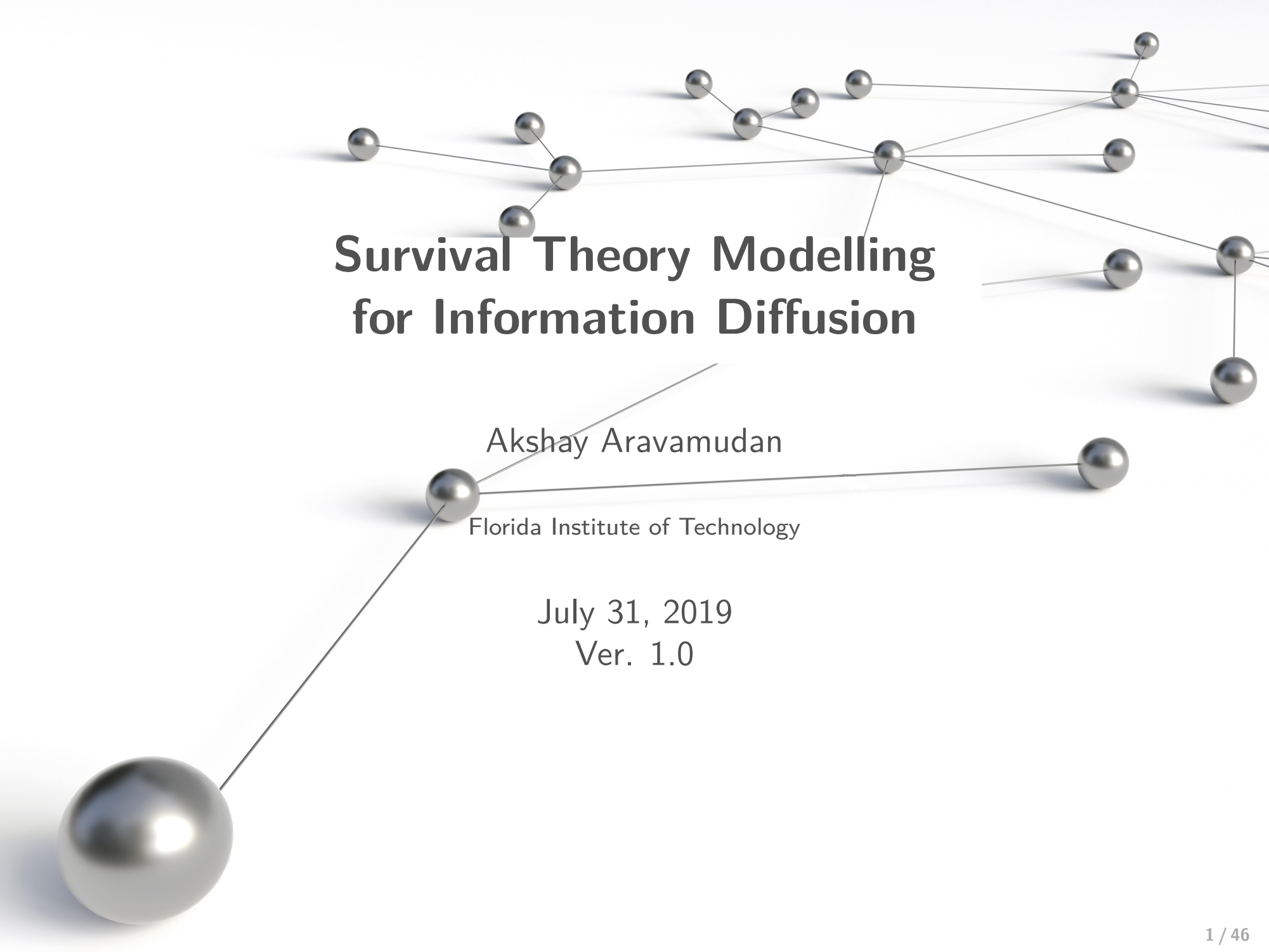


Survival Theory Modelling for Information Diffusion

Akshay Aravamudan

Florida Institute of Technology

July 31, 2019
Ver. 1.0

For queries regarding thesis or presentation contact:

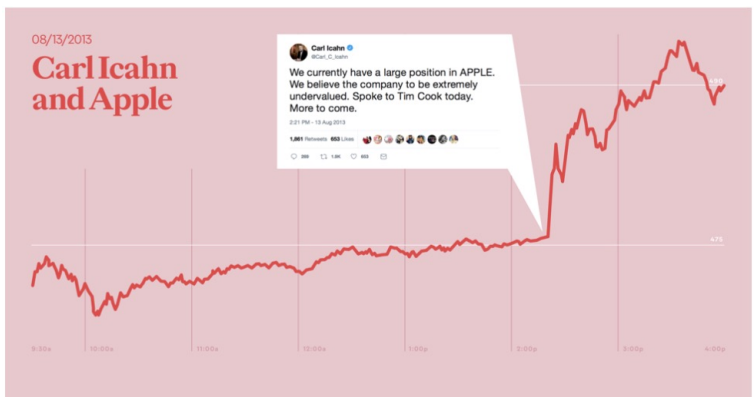
Akshay Aravamudan

aaravamudan2014@my.fit.edu

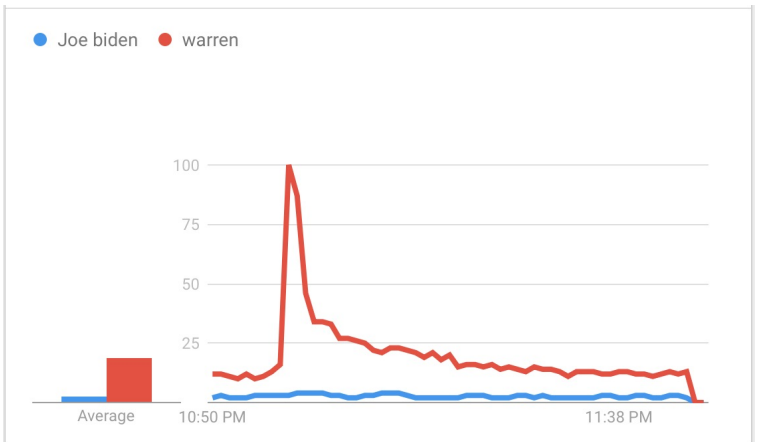
Outline

Introduction

- Information Diffusion refers to spreading of information in a network.
- **Why model the diffusion of information ?**
  - 'Digital' polling for campaigns
  - News propagation for stock portfolios.
  - Marketing campaigns for products.
  - Fake news mitigation.



(a) Effect of a tweet on apple stock. [1]



(b) Google search trend for Joe Biden and Elizabeth Warren after debates. [2]

[1] Source:https://www.ogilvy.com/feed/11-tweets-that-turned-the-stock-market-upside-down/
[2] Source:https://trends.google.com/trends/?geo=US

Introduction

- Born out of need to infer a graph from information cascades. Why not simply use network by observation ?
- Different types of information propagate differently.Context of information produces different kinds of graphs. **The Quality of the link is important**
- Link could represent the rate of information transfer between nodes. [3]
- We have adopted an existing model called NetRate.
- Information Diffusion is modelled as a survival process.

[3] The transmission rate between node i and j α_{ji} has varying interpretation based on underlying distribution

Contributions

- Incorporating parent information in NetRate Modelling
- A comprehensive description of Expectation Maximization Algorithm for NetRate.
- Time Transformation for Goodness of Fit.
- Efficient Simulation Scheme.
- Model for Predicting Dynamic Virality using NetRate assumptions.

Stochastic Point Processes

Stochastic Point Processes (SPP) are used to model events that fall in the time-axis

- Simplest SPP: The Poisson process.
  Inter-arrival times are exponentially distributed with a constant rate λ [also referred to as homogeneous Poisson process]. It enjoys the following properties:
  - Independent increment property: Inter-arrival times are independent of previous occurrences
  - Stationary increments property: Number of arrivals in an interval depends only on the duration of the interval, not the exact time.
- Now, consider that the rate function is a deterministic function of time, $\lambda = \lambda(t)$. This is referred to as a non-homogeneous SPP. Consider the following definition for the rate function:

$$\lambda(t \mid \mathcal{H}_t) = \lim_{h \to 0} \frac{\mathcal{P}\{N_{t+h} - N_t = 1 \mid \mathcal{H}_t\}}{h} \tag{1}$$

  Where $\mathcal{H}_t$ is the history. The process is said to adapt to the natural filtration. The "database" maintaining the history automatically collects itself as events occur.

Survival Analysis

- Study of the lifetime distribution of events. There is a concept of finality: Death, Infection.
- This indicates an important contrast from SPPs that for a realization of a single process, an event takes places only once.
- The intensity function is usually referred to as the Hazard function.

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathcal{P}\{t < T \leqslant t + \Delta t \mid T > t\}}{\Delta t} \tag{2}$$

  Where T is the time of death.

- We use the Intensity-Survival Probability (I-SP) representation of the lifetime distribution:

$$f(t) \triangleq \lambda(t) \underbrace{\exp\{-\Lambda(t)\}}_{S(t)} \tag{3}$$

  where $\Lambda(t) = \int_0^t \lambda(\tau) d\tau$

Survival analysis: intensity function

We associate a lifetime distribution $f(t)$ with well-known distributions like exponential and rayleigh using the Intensity-Memory kernel (I-MK) representation [4]. This enables us to represent the lifetime distribution based on the memory kernel ϕ and Integrated memory kernel ψ:

$$f(t) \triangleq \phi(t) \exp\{-\psi(t)\} [\![t \geqslant 0]\!] \tag{4}$$

The above expression is a valid PDF (integrates to 1). Some common I-MK representations are:

MK $\phi(t)$	IMK $\psi(t)$	PDF $f(t) = \phi(t) \exp\{-\psi(t)\}$
1	t	Unit rate Exponential
t	$\frac{t^2}{2}$	Unit scale Rayleigh
$\frac{1}{t}[\![t \geqslant \beta]\!]$	$\ln\left(\frac{t}{\beta}\right)[\![t \geqslant \beta]\!]$	Unit shape Power-law $\beta > 0$

[4] In Survival Analysis literature, the Hazard function is a linear combination of the memory kernel $\phi(\,)$ and log-survival function is a linear combination of the integrated memory kernel $\psi(\,)$

Outline

Model Description and Assumptions

NetRate models the infection time between two nodes as a survival process.

- An information diffusion network is modelled by an symmetric directed graph of $\mathcal{N}$ users where every edge between nodes j to i is weighted by a transmission rate of $\alpha_{ji} \geqslant 0$
- At any given time t, infection events of non-infected nodes are independent of each other.
- A node can be infected only once per cascade.
- The lifetime distribution of an uninfected node is governed by the hazard rate:

$$h_i(t \mid \mathcal{H}_t^c) \triangleq \sum_{j \in \mathcal{J}^c(t)} \alpha_{j,i} \phi(t - t_j^c), i \notin \mathcal{I}^c(t) \tag{5}$$

  Where $\mathcal{J}^c(t)$ is the set of infected nodes in cascade c

Model Description and Assumptions

- Since $i \notin \mathcal{I}^c(t)$, The infection dynamics of the entire network is modelled as a multivariate survival process with the following Conditional Intensity Function:

$$\lambda_i(t|\mathcal{H}_t^c) \triangleq [\![t \leqslant t_i^c]\!] h_i(t|\mathcal{H}t^c) = \\ = [\![t \leqslant t_i^c]\!] \sum_{j \in \mathcal{I}^c(t)} a_{j,i} \phi(t - t_j^c) \qquad i \in \mathcal{N} \tag{6}$$

- All information diffusion episodes have a common right-censoring time, T.

An infection events is a tuple (i, t) which represents node i infected at time t
An information cascade is a sequence of infection events in chronological order $\mathcal{H}_{T^+} \triangleq \{t_{j_1}, t_{j_2}, \ldots, t_{j_n}\}$. The set of infected nodes in a cascades is represented as:

$$\mathcal{I}^c(t) \triangleq \left\{j \in \mathcal{N} : t_j^c < t\right\} \tag{7}$$

NetRate Considerations

The assumptions of the NetRate model constrains our scenarios to which this model are applicable.

The events of interest follow a binary state mechanism. Such models are referred to as Susceptible Infected (SI) models. Once a susceptible node is infected, it remains infected for the entire diffusion episode.

The first infection should have already taken place.

- News articles published in blogs and websites.
- Spreading of a disease in a population.
- Spreading of a particular piece of information in a network, rumors for instance.

We are only constrained by the existance and quality of the data.

NetRate Likelihood

Likelihood defines how well the parameters of the model describe the data. It is often in a machine learning context to learn the parameters of the model.

The likelihood expression is built upon the likelihood of a single information cascade. Consider a transmission matrix $\mathbf{A} = \{a_{ji} \mid j, i, \in \mathcal{N}\}$

$$L(\mathbf{A}|\mathcal{H}_T^c) \triangleq f\left(\underbrace{\{t_i^c\}_{i \in \mathcal{I}_{\setminus 1}^c(T)}}_{\text{Infected nodes in cascade}}, \bigcap_{k \notin \mathcal{I}^c(T)} \underbrace{\{T_k^c > T\} \Big| t_{j_1^c}^c}_{\text{Nodes have survived the cascade}} \right) \tag{8}$$

$$\ell(\mathbf{A}|\mathcal{H}_T^c) \triangleq \ln L(\mathbf{A}|\mathcal{H}_T^c) \tag{9}$$

$$= \sum_{i \in \mathcal{I}_{\setminus 1}^c(T)} \left(\ln \lambda_i(t_i^c|\mathcal{H}_{t_i^c}^c) - \Lambda_i(t_i^c|\mathcal{H}_{t_i^c}^c) \right) - \sum_{i \notin \mathcal{I}^c(T)} \Lambda_i(T|\mathcal{H}_T^c) \tag{10}$$

Where, $\Lambda_i(t_i^c|\mathcal{H}_{t_i^c}^c)$ is called the integrated intensity function and can be shown to be represented as

$$\Lambda_i(t|\mathcal{H}_t^c) = \sum_{j \in \mathcal{I}^c(t)} a_{j,i} \psi(t - t_j^c) \qquad i \notin \mathcal{I}^c(t) \tag{11}$$

NetRate **Likelihood**

Without going into details, we arrive upon the final expression for the likelihood by considering following cases for a node i

- Infection of node i is observed in cascade c and its parent is unknown.
- Infection of node i is observed in cascade c and its parent is observed.
- Infection of node i is not observed in cascade.
- node i is the first infected node in the cascade (then we conclude that $\alpha_{ji} = 0$)

$$\ell(\mathbf{a}_i|\mathcal{H}_T) = \ell_1(\mathbf{a}_i|\mathcal{H}_T) + \ell_2(\mathbf{a}_i|\mathcal{H}_T) + \ell_3(\mathbf{a}_i|\mathcal{H}_T) + \ell_4(\mathbf{a}_i|\mathcal{H}_T) \tag{12}$$

In order to make it computationally efficient, we create sets to store the node indexes and pre-computed values. This log-likelihood is concave due to composition of linear and concave functions. (for the memory kernels mentioned earlier).

Likelihood term 1

$$\ell_1(\mathbf{a}_i|\mathcal{H}_T) = -\sum_{j \in \mathcal{J}_1(i)} a_{j,i} D_1(j|i) \tag{13}$$

$$D_1(j|i) = \begin{cases} \sum_{c \in \mathcal{C}_1(j|i)} \psi(t_i^c - t_j^c) & j \in \mathcal{J}_1(i) \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

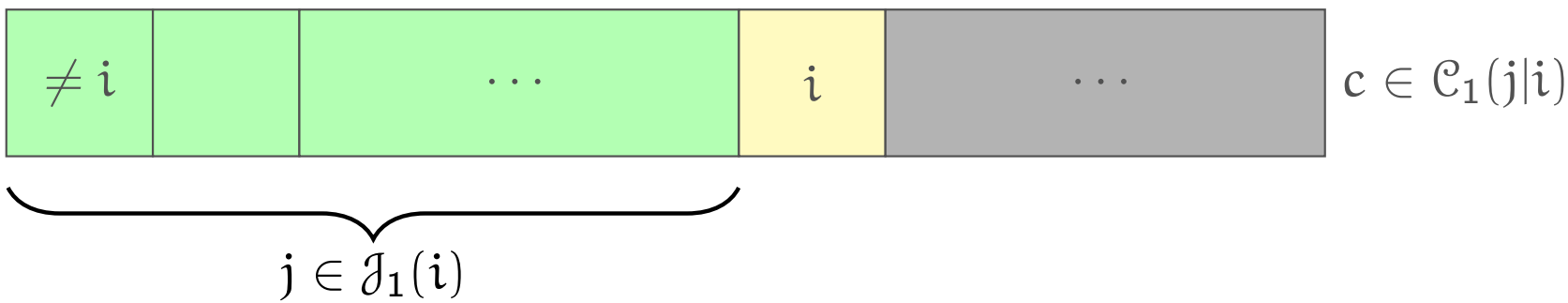


Figure: Figure showing cascade c's contribution to $\ell_1(\mathbf{a}_i|\mathcal{H}_T)$. Highlighted in green are the node indices that will belong to $\mathcal{J}_1(i)$. For such type of cascades, i's infection is observed but not as the first infected node. Parent information is irrelevant for this term.

Likelihood term 2

$$\ell_2(\mathbf{a}_i|\mathcal{H}_T) = \sum_{j \in \mathcal{J}_2(i)} D_2(j|i) \ln a_{j,i} \tag{15}$$

$$D_2(j|i) = \begin{cases} |\mathcal{C}_2(j|i)| & j \in \mathcal{J}_2(i) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

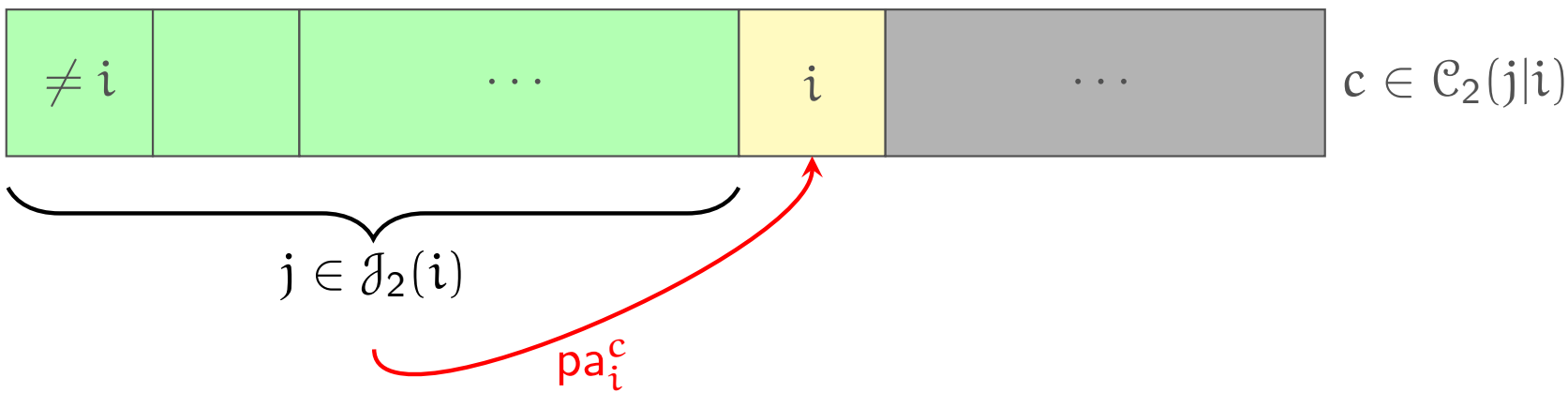


Figure: Figure showing cascade c's contribution to $\ell_2(\mathbf{a}_i|\mathcal{H}_T)$. Highlighted in green are the node indices that will belong to $\mathcal{J}_2(i)$. The red arrow conveys that one of the nodes j is the observed parent of i in this cascade. Again, for such type of cascades, i's infection is observed but not as the first infected node.

Likelihood term 3

$$\ell_3(\mathbf{a}_i|\mathcal{H}_T) = \sum_{c\in\mathcal{C}} [\![i \in \mathcal{I}^c_{u,\setminus 1}(T)]\!] \ln \sum_{j\in\mathcal{N}} [\![j \in \mathcal{I}^c(t^c_i)]\!] a_{j,i} \phi(t^c_i - t^c_j) \tag{17}$$

(18)

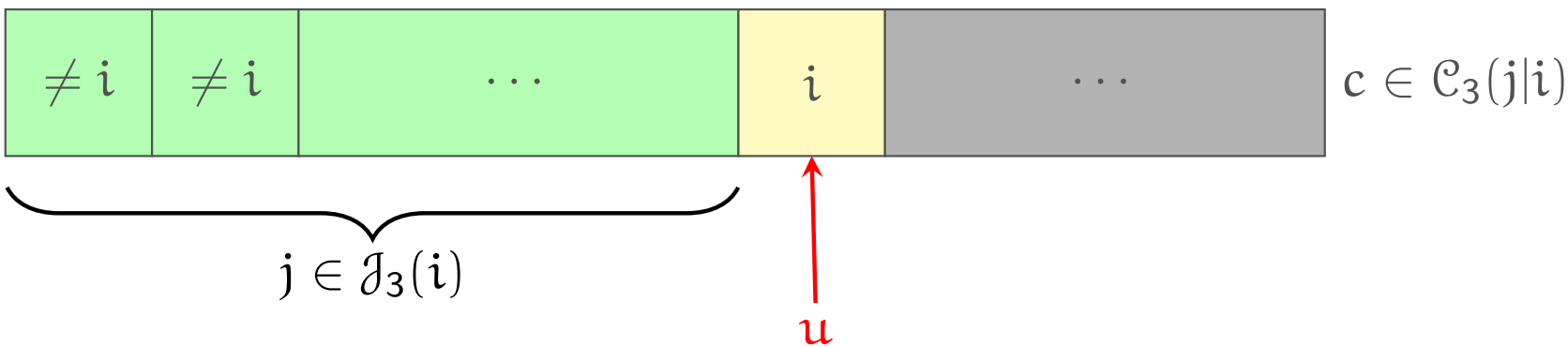


Figure: Figure showing cascade c's contribution to $\ell_3(\mathbf{a}_i|\mathcal{H}_T)$. In this case, i's infection has been observed. Highlighted in green are the node indices that will belong to $\mathcal{J}_3(i)$. The red arrow conveys that i's parent has not been observed in this cascade. This necessarily means that $j^c_1, j^c_2 \neq i$, *i.e.*, it was neither the first or second infected node in the cascade.

Likelihood term 4

$$\ell_4(\mathbf{a}_i|\mathcal{H}_T) = \sum_{j \in \mathcal{J}_4(i)} a_{j,i} D_4(j|i) \tag{19}$$

$$D_4(j|i) = \begin{cases} \sum_{c \in \mathcal{C}_4(j|i)} \psi(T^c - t_j^c) & j \in \mathcal{J}_4(i) \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

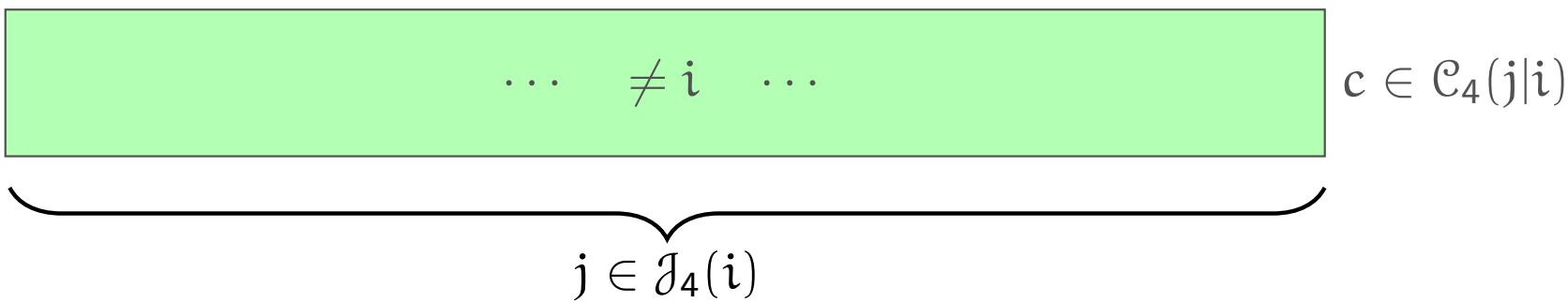


Figure: Figure showing cascade c's contribution to $\ell_4(\mathbf{a}_i|\mathcal{H}_T)$. Highlighted in green are the node indices that will belong to $\mathcal{J}_4(i)$. This is a cascade, where i's infection is not observed by the right-censoring time T.

On the optimization of NetRate

Since the number of information cascades is small, we use an ℓ_1 regularizer. The optimization problem is therefore formulated as

$$\max_{\mathbf{a}_i \in \mathbb{R}_+^N} \underbrace{[\ell(\mathbf{a}_i|\mathcal{H}_T) - \nu_i \|\mathbf{a}_i\|_1]}_{\ell^R(\mathbf{a}_i|\mathcal{H}_T) \triangleq} \qquad i \in \mathcal{N} \tag{21}$$

- ν_i is the regularization coefficient.
- It can be divided into subproblems, each solved independently.
- Given the likelihood, one option is to perform gradient descent. The constraint implies that the gradient steps must follow a projected gradient descent.
- To avoid escaping feasible region, we can use line search to find the best learning rate parameter. This, however is computationally expensive.
- So, we use an Expectation Maximization (EM) Algorithm which provides us with a closed form solution for parameter updates.

Expectation Maximization

- EM algorithm works by finding a minorizer for the log-likelihood. More specifically, it lower bounds the $\ell_3(\mathbf{a}_i|\mathcal{H}_T)$ using the Jensen Inequality[5].

$$\ln \sum_{j \in \mathcal{I}^c(t_i^c)} \alpha_{j,i} \phi(t_i^c - t_j^c) \geqslant \sum_{j \in \mathcal{I}^c(t_i^c)} (p_{j,i}^c)' \ln \alpha_{j,i} + \text{const} \tag{22}$$

  where $(p_{j,i}^c)'$ stands for the probability of j being the parent node of node i

- The lower bound of the ℓ_3 terms is

$$\bar{\ell}_3(\mathbf{a}_i|\mathbf{a}_i', \mathcal{H}_T) = \sum_{j \in \mathcal{J}_3(i)} D_3'(j|i) \ln \alpha_{j,i} + \text{const} \tag{23}$$

  where

$$D_3'(j|i) \triangleq \begin{cases} \sum\limits_{c \in \mathcal{C}_3(j|i)} (p_{j,i}^c)' & j \in \mathcal{J}_3(i) \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

- The EM algorithm is divided into two steps: EM Setup and EM Updates.

[5] The Jensen inequality upper bounds the log of sums to a sum of logs.

EM Algorithm Setup and Updates

The Setup of the EM algorithm involved generating the following sets beforehand. This involves iterating over all the cascades per sub problem. This saves us from having to iterate over cascades in the training part.

- Set $\mathcal{D}(i)$ of relevant node indices to sub problem i
- Node index sets: $\mathcal{J}_1(i)$, $\mathcal{J}_2(i)$, $\mathcal{J}_3(i)$ and $\mathcal{J}_4(i)$
- The sets with pre-calculated ϕ,ψ values: $D_1(j|i), D_2(j|i)$ and $D_3(j|i) \quad \forall j \in \mathcal{N}$
- The cascade index set $\mathcal{C}_3$ and the ψ values associated with $D_3(j|i)$. We store $\mathcal{C}_3$ as a dictionary so that it can be iterated through to obtain the value of $D_3(j|i)$ per EM update iteration.

Using the pre-calculated sets, we can design a learning algorithm with the following update step :

$$a^*_{k,i} = \frac{D_2(k|i) + D'_3(k|i)}{\nu_i + D_4(k|i) + D_1(k|i)} \qquad k \in \mathcal{D}(i) \tag{25}$$

$$a^*_{k,i} = 0 \qquad k \notin \mathcal{D}(i) \tag{26}$$

NetRate Training using EM

- Divide the data-set into training and validation (9:1 ratio)
- We decide on a set of regularizers $\{\nu_i\}$ and choose the one that produces the highest validation likelihood after learning parameters $\alpha_{ji}, j \in \mathcal{N}$
- Fix maximum number of iterations.
- Break if $\|a_{ji}(previous) - a_{ji}(current)\|_\infty < \epsilon$, where ϵ is some tolerance.
- Fix a memory kernel and choose the one which produces the best validation likelihood (after having chosen regularizer).

An Experiment: Effect of Partial Parentage Information in Data

In order to investigate influence of parentage information, we generated 5000 cascades with varying parent probability generation. The tolerance used was $\epsilon = 10^{-7}$

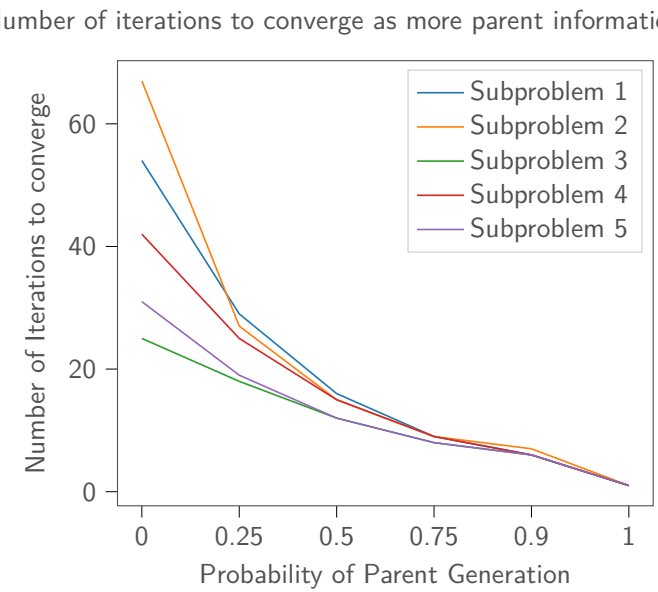


Figure: The probability of parentage is increased and we find the number of iterations required to converge

As the number of parent events increase, the iterations required to converge decreases. Finally, when all the events in the cascade have parent information, we get a closed form solution. Hence it converges in 1 iteration.

Outline

Transforming infection times

After parameter estimation for NetRate, we need to see how well the model fits the data. Other the direct comparison of training log-likelihood, we need to compare the results with a known distribution.

- We propose the following time transformation:

$$\tau_i^c = \Lambda_i(t_i^c | \mathcal{H}_{t_i^c}^c) = \sum_{j \in \mathcal{I}^c(t_i^c)} \alpha_{j,i} \psi(t_i^c - t_j^c) \qquad \text{for } i \in \mathcal{I}_{\setminus 1}^c(t_i^c) \tag{27}$$

$$\tau_i^c = \Lambda_i(T | \mathcal{H}_T^c) = \sum_{j \in \mathcal{I}^c(T)} \alpha_{j,i} \psi(T - t_j^c) \qquad \text{for } i \notin \mathcal{I}^c(T) \tag{28}$$

- We proved that this transformed time comes from the unit rate exponential distribution.
- We can now compare the distributions by generating a P-P (Probability-Probability) plot. We also generate a (Kolmogorov Smirnov)K-S test statistic and make conclusions with the p-value.

Transformation Algorithm

Algorithm 1 Algorithm for Transforming infection time in an information cascade

Input: transmission Matrix: $A = \{a_{ji} \mid j, i \in \mathcal{N}\}$, node ID: i, cascade cas, right Censoring time T, Integrated Memory kernel $\psi(t)$

$\tau_i^c \leftarrow 0$

if i is infected in cas **then**

  for $j \mid j \in$ infected nodes in cas **do**

    if $t_j < t_i$ **then**

      $\tau \leftarrow \tau + a_{ji}\psi(t_i - t_j)$

    end if

  end for

else

  for $j \mid j \in$ infected nodes in cas **do**

    $\tau \leftarrow \tau + a_{ji}\psi(T - t_j)$

  end for

end if

Output: transformed time τ_i^c

Outline

Simulation Methodology

Simulation was instrumental for debugging the model, since we had a ground truth to work with. Formal proof is based on proof for multivariate Hawkes process found in Ogata's 1981 paper. [1]

- Calculate **upper bound** of intensities.
- Simulate the **waiting time** using ground process (sum of all intensities).
- Recalculate intensities based on new waiting time.
- **thinning** accept points by simulation by comparing ground process of upper bound and updated intensity based on waiting time.
- **randomly sample a node ID** of uninfected nodes with a probability proportional to the conditional intensities evaluated at that time.
- **randomly sample a parent node ID** of infected nodes with a probability proportional to the conditional intensities evaluated at that time.
- **repeat until** right censoring time T.

Case study: A visual verification on ground truth data

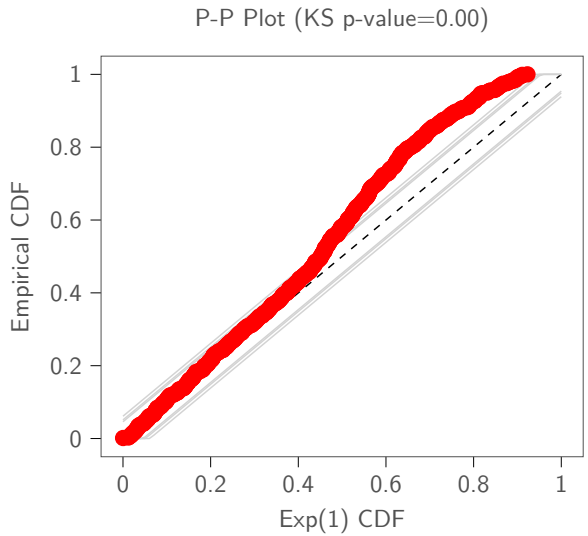


Figure: P-P plot when simulated cascades have some survived nodes

Some Observations:

- When some nodes have survived, the P-P plot deviates from the 45° line and the p-value is always zero (which is not good).
- Whenever there are a lot of survived nodes, The p-value obtained is zero. This was a repeated observation for other examples as well.
- when we compared inferred values with ground truth, a similar situation as above caused deviation from the ground truth.

Case study: A visual verification on ground truth data

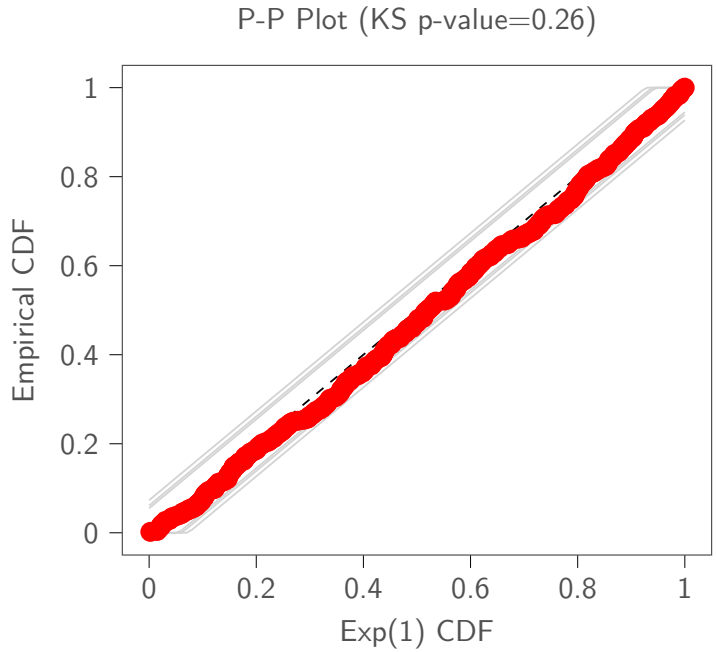


Figure: P-P plot when simulated cascades have some survived nodes

Some Observations:

- In the perfect case when all the nodes are infected within the right censoring time, a high p-value is obtained indicating a good fit. This situation was replicated using other examples as well.

Outline

Dynamic Virality Model: excess number of infections

The assumptions of NetRate, specifically the one about infection events in a cascade being independent of each other allow us to model the Probability Mass Functions (PMF) of the excess number of infections. We treat the number of future infections per node as a Bernoulli Random Variable (RV).

$$N_i(T + t_x)|\mathcal{H}_T \overset{\text{iid}}{\sim} \text{Bernoulli}(\pi_i(t_x|T, \mathcal{H}_T)) \tag{29}$$

This helps us model the total number of excess infections as in a given time interval t_x after the observed time of the latest infection, T. The set $\mathcal{I}(T)$ contains all the infected nodes before T.

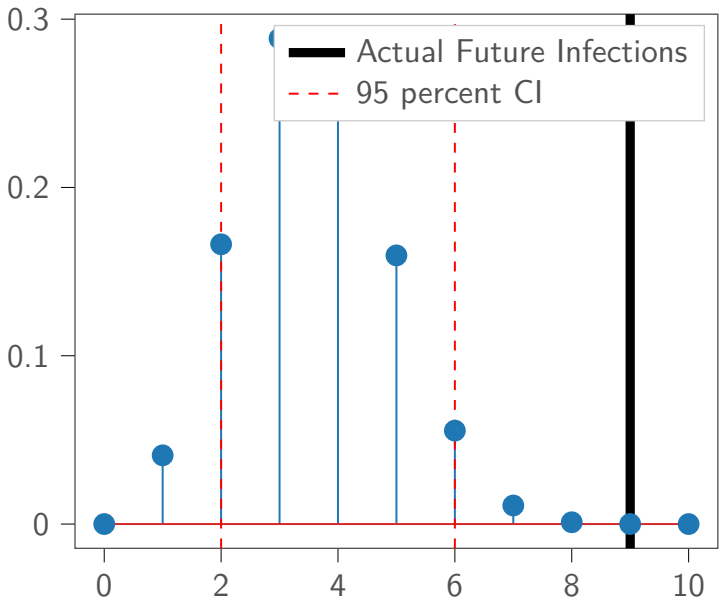
$$N_x(t_x|T) = \sum_{i \notin \mathcal{I}(T)} N_i(T + t_x) \tag{30}$$

This lets us define the PMF of the excess number of infections as a convolution between individual PMFs defined as,

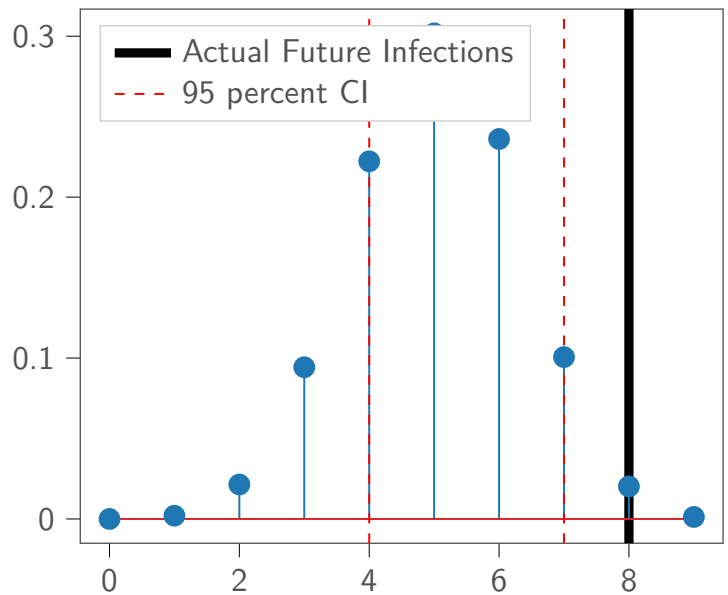
$$p_{N_x(t_x|T)}(\cdot|\mathcal{H}_T) = \underset{i \notin \mathcal{I}(T)}{*} p_{N_i(T+t_x)}(\cdot|\mathcal{H}_T) \tag{31}$$

Prediction for a test cascade I

For large example with a lot of nodes surviving cascades, we mentioned earlier that ground truth would not be approached. So, we simulated for a small example with 10 nodes where all the nodes are infected in cascades.
For a test cascade, we fixed the first n infections and generated PMF for future infections.

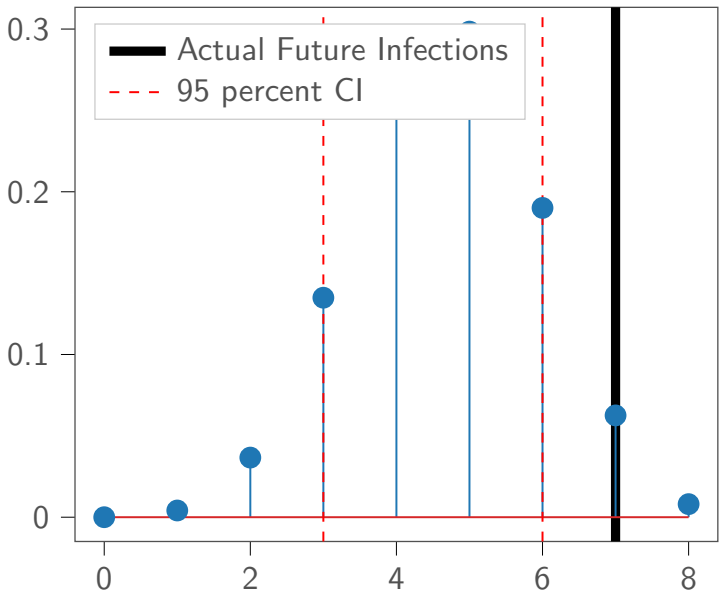


(a) $n = 1$

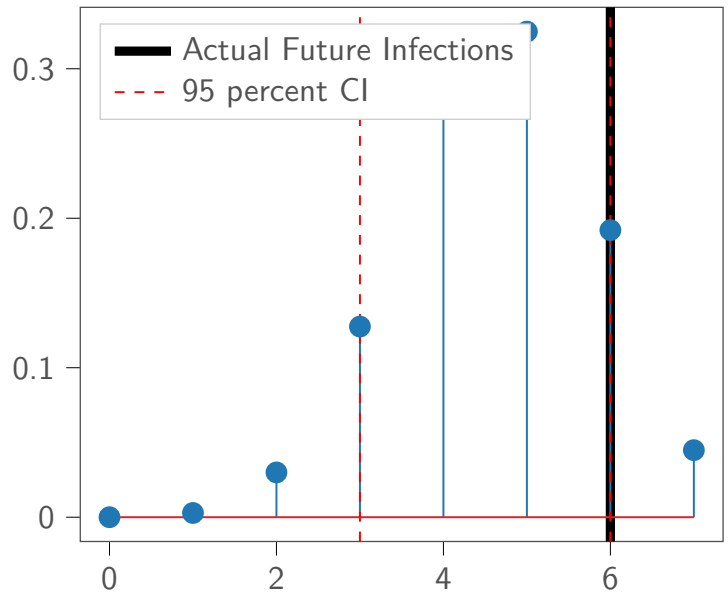


(b) $n = 2$

Prediction for a test cascade II

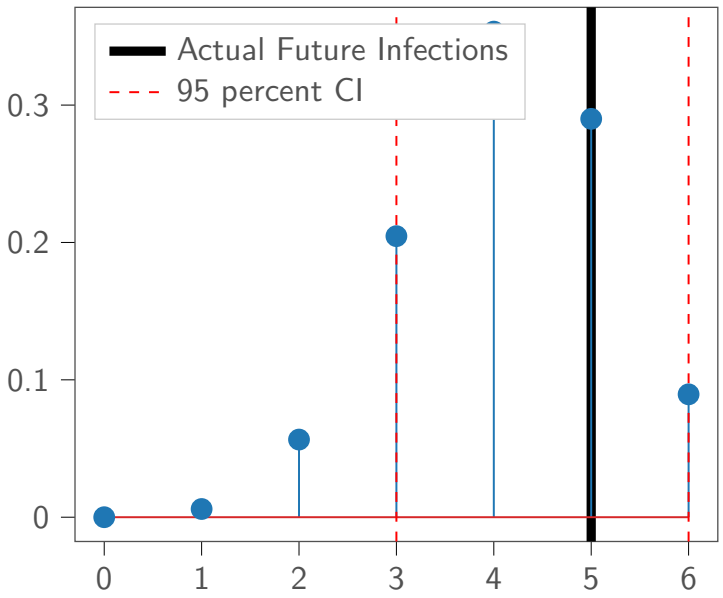


(c) $n = 3$

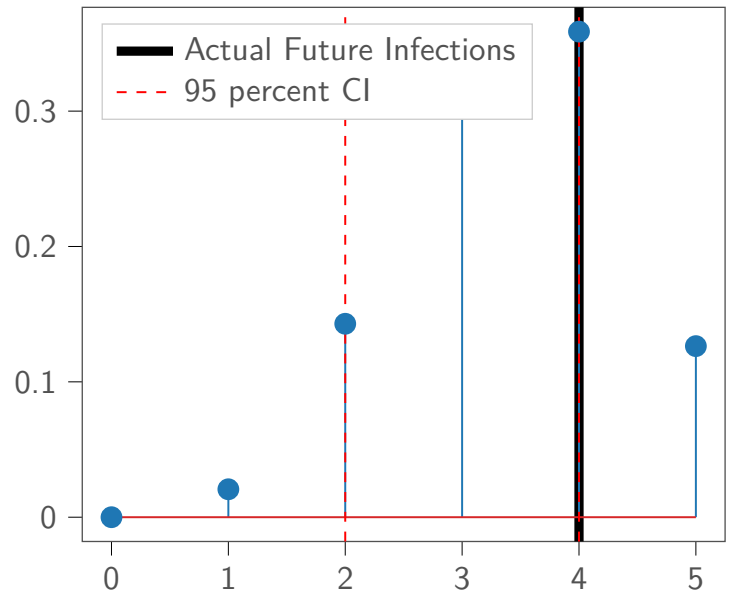


(d) $n = 4$

Prediction for a test cascade III

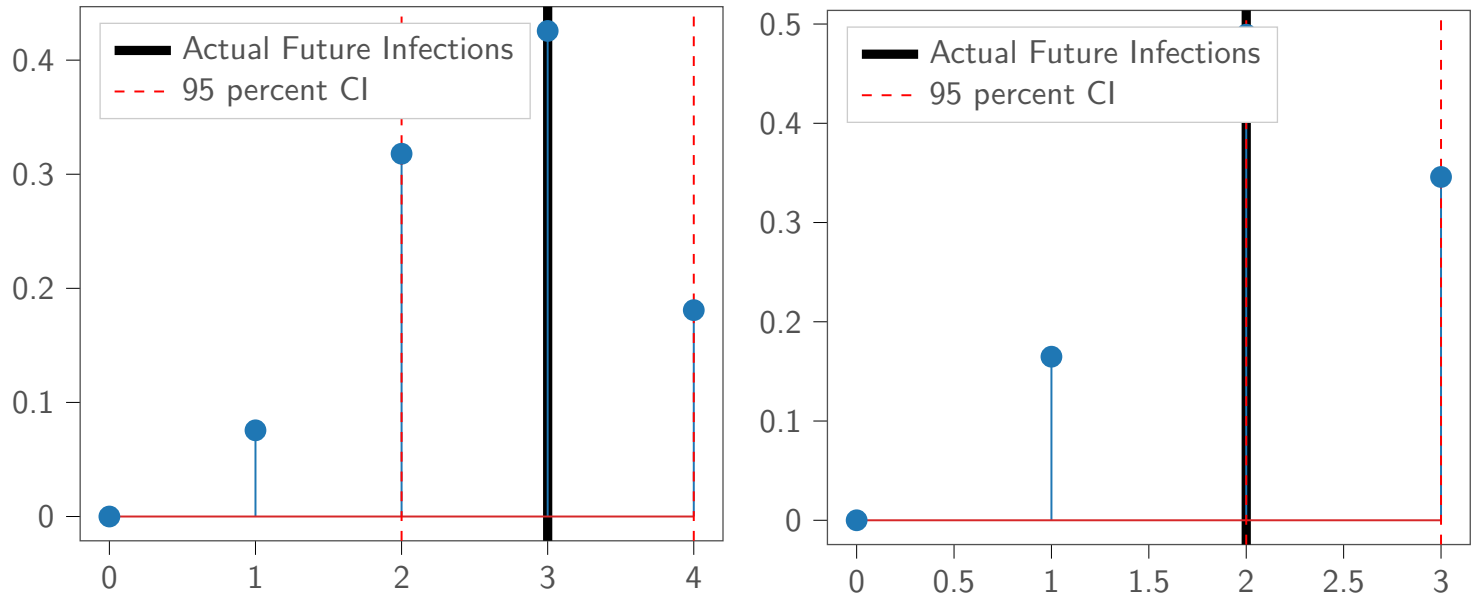


(e) $n = 5$



(f) $n = 6$

Prediction for a test cascade IV



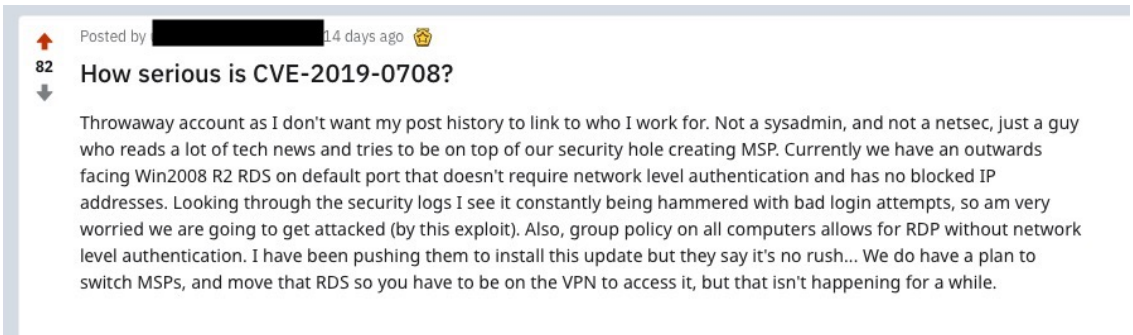
(g) $n = 7$ **(h)** $n = 8$

Figure: The above Figure are the PMF Generated for future infections as the number of observed infections are increased.

Outline

CVE Information Cascades I

For working with real-world data, we were provided with CVE (Common Vulnerabilities as Exploits) data. Discussions about the vulnerability constitutes an information cascade.
One CVE represents one information cascade.



(a) Example of an event from Reddit that contributes to the CVE cascade

CVE Information Cascades II

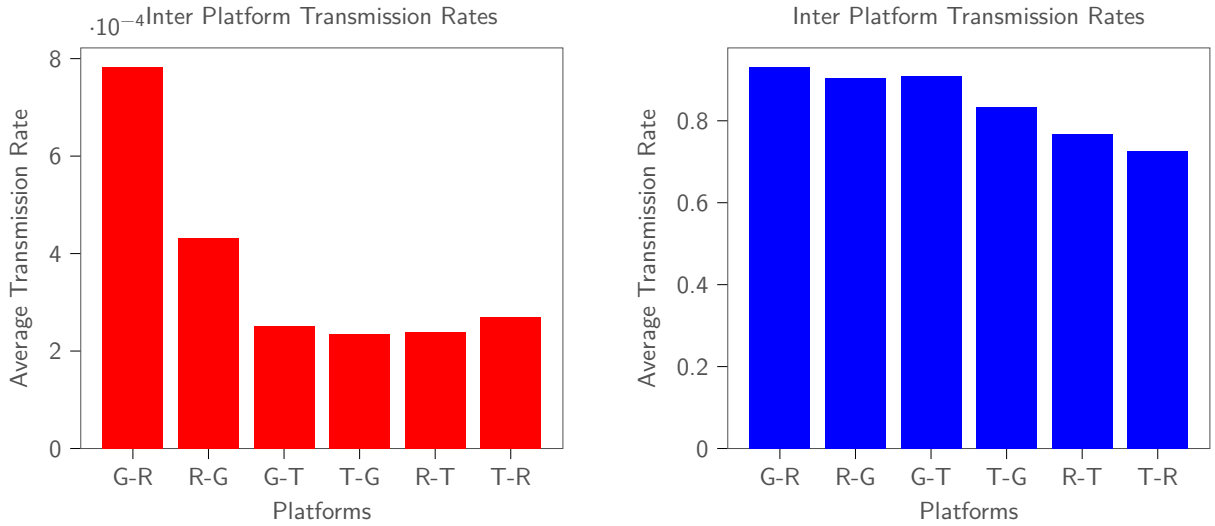
(b) Example of an event from Github that contributes to the CVE cascade

(c) Example of an event from Twitter that contributes to the CVE cascade

We compared two kinds of vulnerabilities: Exploited and non-Exploited. Exploited CVEs referred to vulnerabilities which have been exploited by malicious users. These have **High Priority** for the stakeholders. Some statistics about the data.

- Exploited CVEs: 940 cascades, 4248 nodes
- Non-Exploited CVEs: 12129 cascades, 6737 nodes

NetRate on CVE: Some Results



(a) Average inter-platform transmission rates for Exploit CVEs

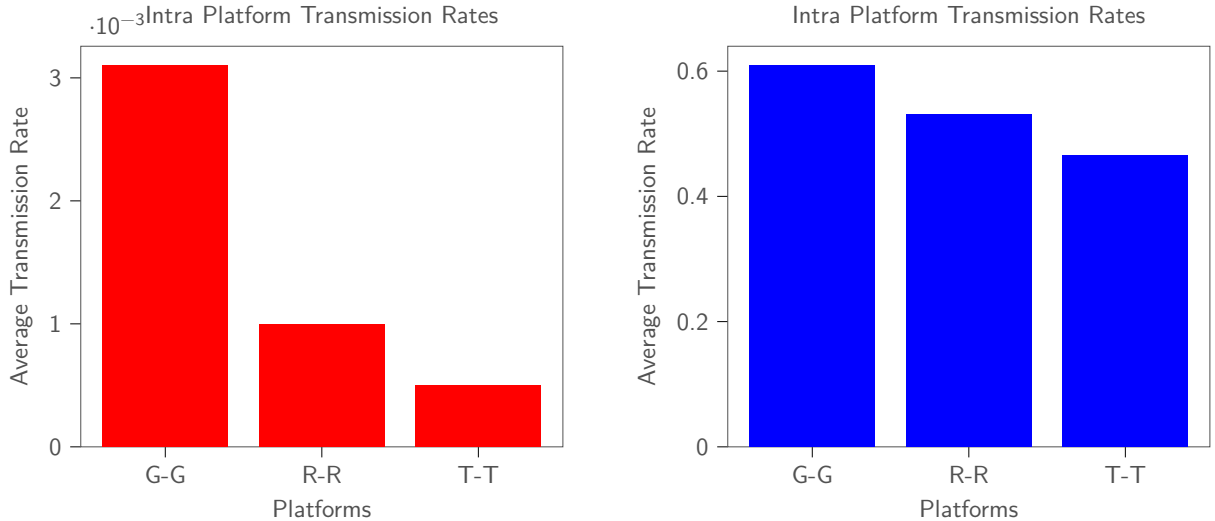
(b) Average inter-platform transmission rates for Non-exploit CVEs

Figure: G - Github, R- Reddit, T - Twitter

Observations and conclusions

- Discussions propagating from Github to Reddit contribute more to exploited CVEs.
- Github plays a strong role in exploited CVEs on account of direct access to code.

NetRate on CVE: Some Results



(a) Average intra-platform transmission rates for Exploit CVEs

(b) Average intra-platform transmission rates for Non-exploit CVEs

Figure: G - Github, R- Reddit, T - Twitter

Observations

- Exploited CVEs have more Github activity compared to Non-exploited.
- Exploit CVEs have a smaller transmission rate, indicating less involvement within the community. Which explains why they were exploited in the first place.

Outline

Conclusion

- NetRate models infection between two nodes with a lifetime distribution. Experiments on CVE data suggest power-law distribution provides best fit based on validation likelihood.
- Time transformation of infection times allows to perform a goodness of fit.
- Future popularity of a cascade can be modelled as bernoulli RVs, whose convolution gives the PMF of excess infections.
- Comparison with Hawkes process allows us to use Ogata's modified thinning algorithm. The methodology also allows us to simulate the parent node of infection events.
- Explored a real-world example of software vulnerabilities.

Future Work

Non-Parametric estimation of ψ function

- Determining the shape of the ψ function non-parametrically as opposed to choosing from a given set of distributions.
- This is an active field of research, some techniques out there for Hawkes process: Online non-parametric learning Xu et al. [2], set of basis functions Yang et al. [3]

Incorporating features to diffusion model.

- CVE's tend to diffuse quickly. In a network of 1000+ nodes, CVEs only touch 4 or 5 nodes.
- Some are immune to infection. This implies there are certain characteristic about the CVE that render it immune to infections.
- Incorporating feature into the model so prediction task can be more accurate.

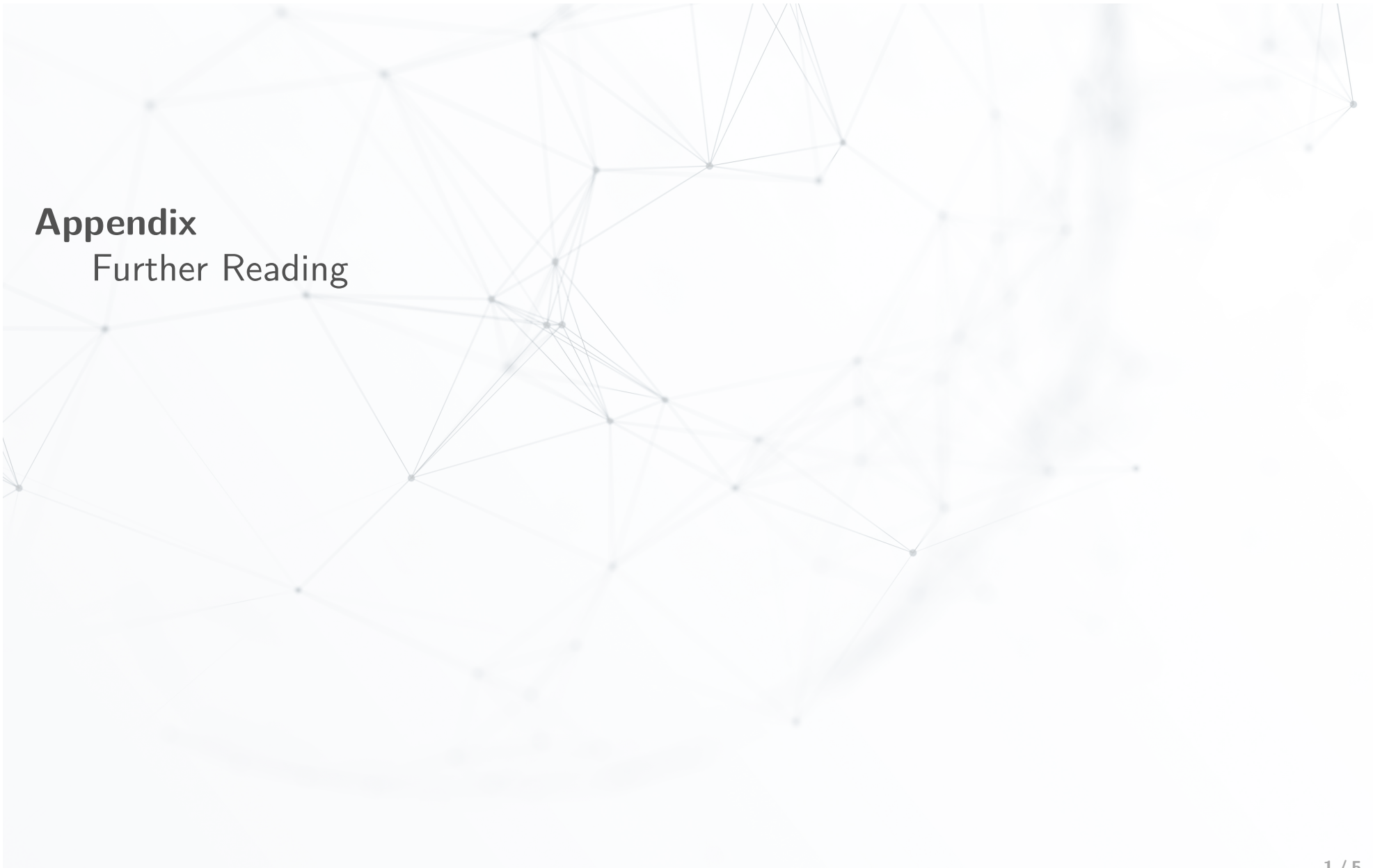
Future Work

Some relevant conferences for publishing

- ICML [6]: July 2020
- NIPS [7]: Dec 2020
- AAAI [8]: Feb 2020

[6] International Conference on Machine learning
[7] Conference on Neural Information Processing Systems
[8] Association for advancement of Artifical Intelligence

Outline



Appendix

Further Reading

Further Reading I

[1] Y. Ogata.
On lewis' simulation method for point processes.
IEEE Transactions on Information Theory, 27(1):23–31, January 1981.
doi:10.1109/TIT.1981.1056305.

[2] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha.
Learning granger causality for hawkes processes.
In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1717–1726, New York, New York, USA, 20–22 Jun 2016. PMLR.
URL: http://proceedings.mlr.press/v48/xuc16.html.

[3] Yingxiang Yang, Jalal Etesami, Niao He, and Negar Kiyavash.
Nonparametric hawkes processes: Online estimation and generalization bounds.
NIPS, 01 2018.

This Page is left empty intentionally

A Mutually and Self-Exciting SPP: The Hawkes Process

In real-world networks involving human interaction, events arrive in following ways:

- Arrive independently of its own volition (say at a constant rate): Self-Exciting
- Arrive as a response to another events occurrence : Mutual-Excitation

This process is best encapsulated by the following shape of intensity function:

$$\lambda(t \mid \mathcal{H}_t) = \underbrace{\mu}_{\textit{Self-Excitation}} + \underbrace{\sum_{i:t>T_i} g(t - T_i)}_{\textit{Mutual-Excitation}} \tag{32}$$

Where g is the defined as the memory kernel (different from our memory kernel definition) The above expression is defined for a single dimensional process, When there are multiple users/nodes in a network, it is referred to as a Multi-Variate Hawkes process.
We will show that the NetRate is analogous to the Multivariate Hawkes process, which we will find useful for our contributions to the model.

More on Dynamic Virality

$$\pi_i(t_x|T,\mathcal{H}_T) \triangleq \mathbb{E}\{N_i(T+t_x)|\mathcal{H}_T\} = \mathbb{P}\{t_i \leqslant T+t_x|\mathcal{H}_T\} \tag{33}$$

$$N_i(T+t_x)|\mathcal{H}_T \overset{\text{iid}}{\sim} \text{Bernoulli}(\pi_i(t_x|T,\mathcal{H}_T)) \tag{34}$$

$$N_x(t_x|T) = \sum_{i\notin\mathcal{I}(T)} N_i(T+t_x) \tag{35}$$

$$p_{N_x(t_x|T)}(\cdot|\mathcal{H}_T) = \underset{i\notin\mathcal{I}(T)}{*} p_{N_i(T+t_x)}(\cdot|\mathcal{H}_T) \tag{36}$$

$$p_{N_i(T+t_x)}(n|\mathcal{H}_T) = \pi_i(t_x|T,\mathcal{H}_T)[\![n=1]\!] + \tag{37}$$

$$+ (1-\pi_i(t_x|T,\mathcal{H}_T))\,[\![n=0]\!] \tag{38}$$