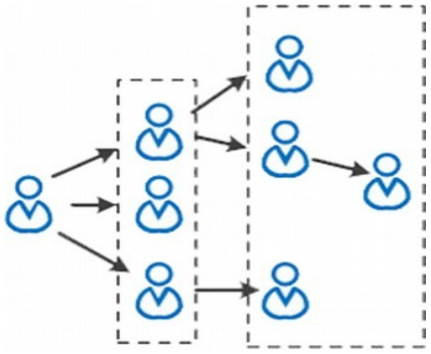
Behavioral Data Science

UTS Data Science

UTS UNIVERSITY OF TECHNOLOGY SYDNEY



Modeling information flow in Online Social Media using Hawkes Point Processes

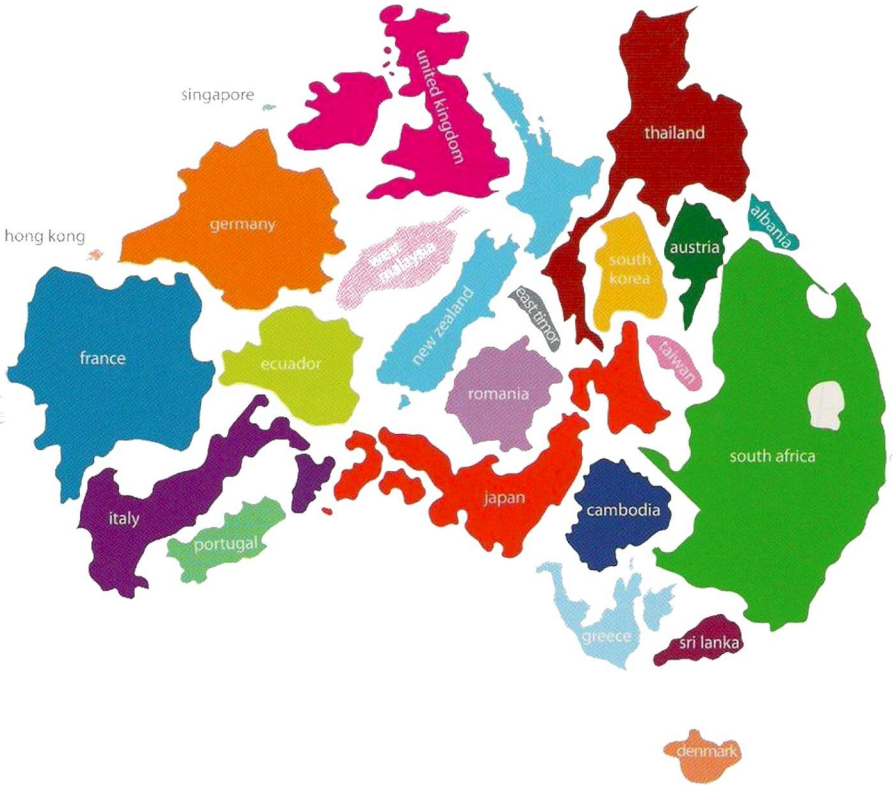
Marian-Andrei Rizoiu | Behavioral Data Science
Marian-Andrei.Rizoiu@uts.edu.au
https://www.behavioral-ds.science

Australia

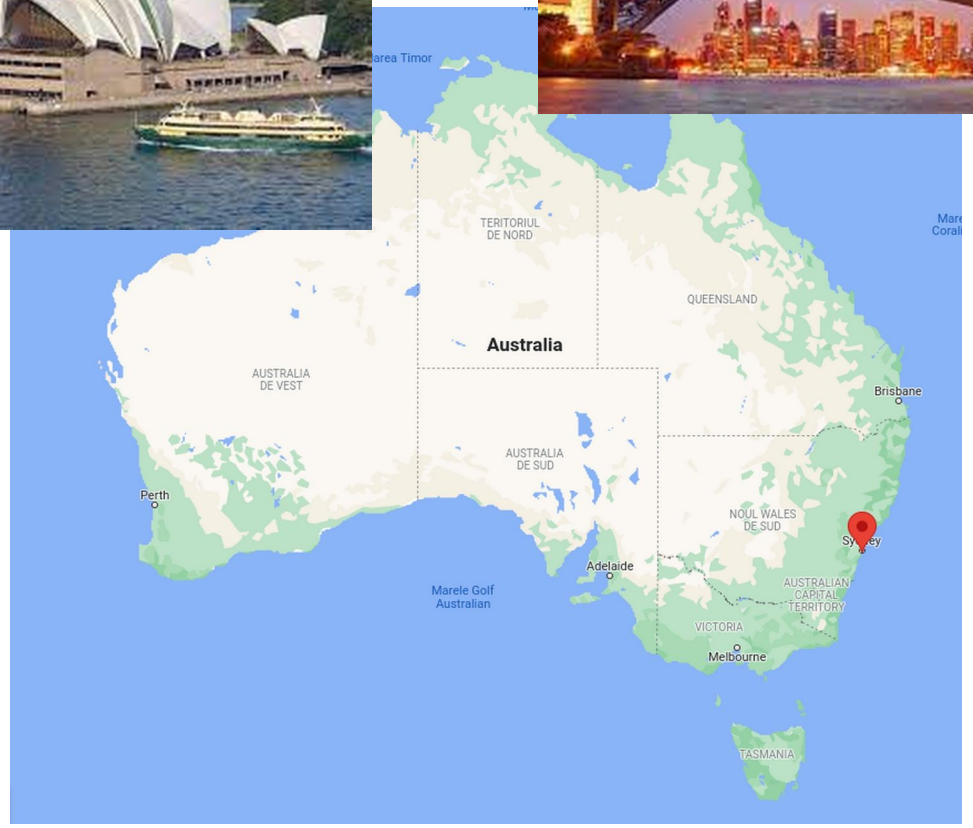
The world map, according to Australians

Australia

The world map, according to Australians



Australia is a big BIG place



Located in Sydney, Australia

A city campus, iconic brutalist style blended with modern buildings

The research group

Behavioral Data Science

2 PostDocs, 7 PhD, 1 Masters, 1 assistant prof.



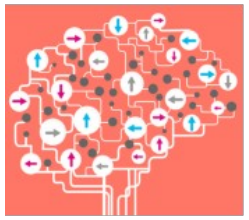




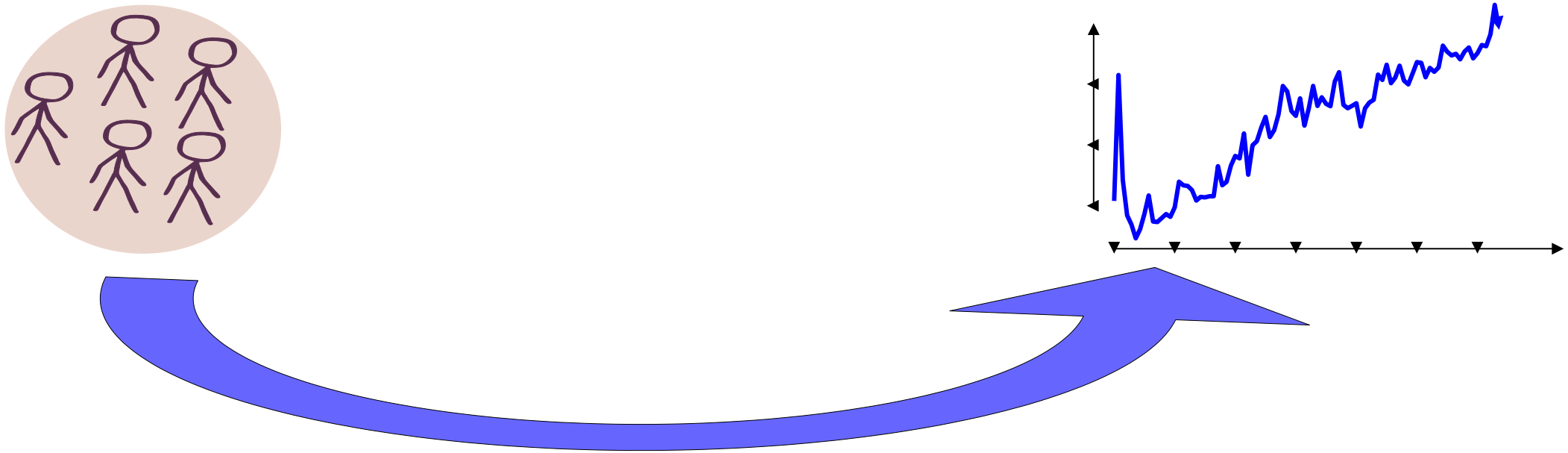




The Behavioral Data Science



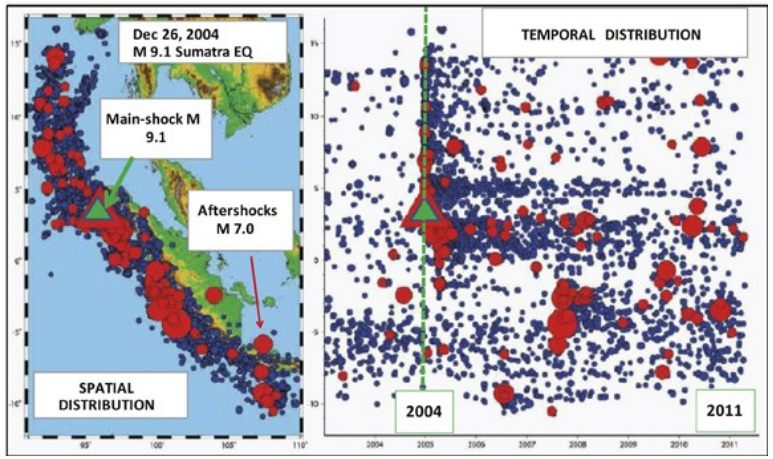
Behavioral Data Science



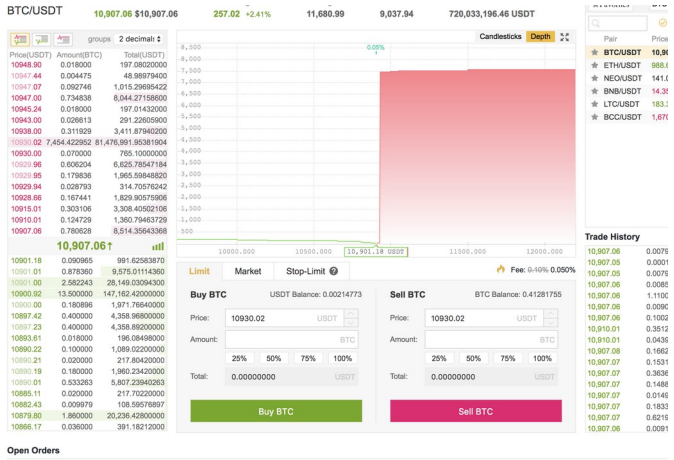
information diffusion
epidemics spreading
behavioral modeling

Self-excitation in real-world data sources

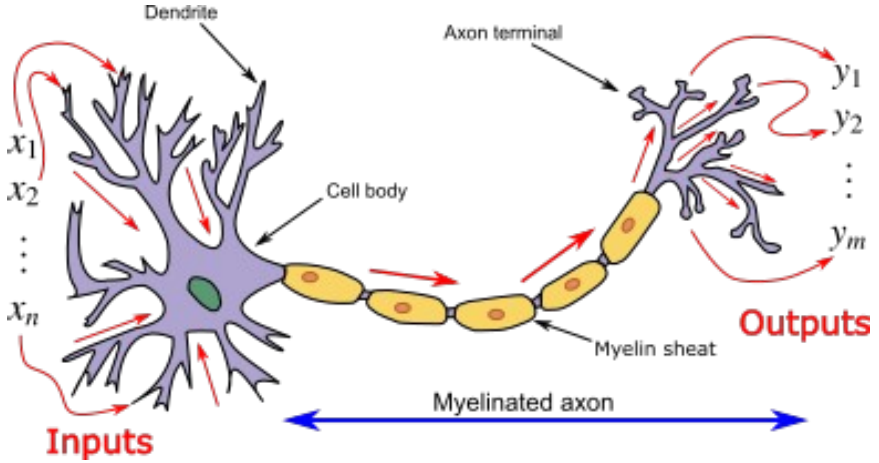
Self-excitation: the occurrence of an event increases the likelihood of future events.



Earthquake aftershocks
[Hawkes 1971, Ogata 1978]

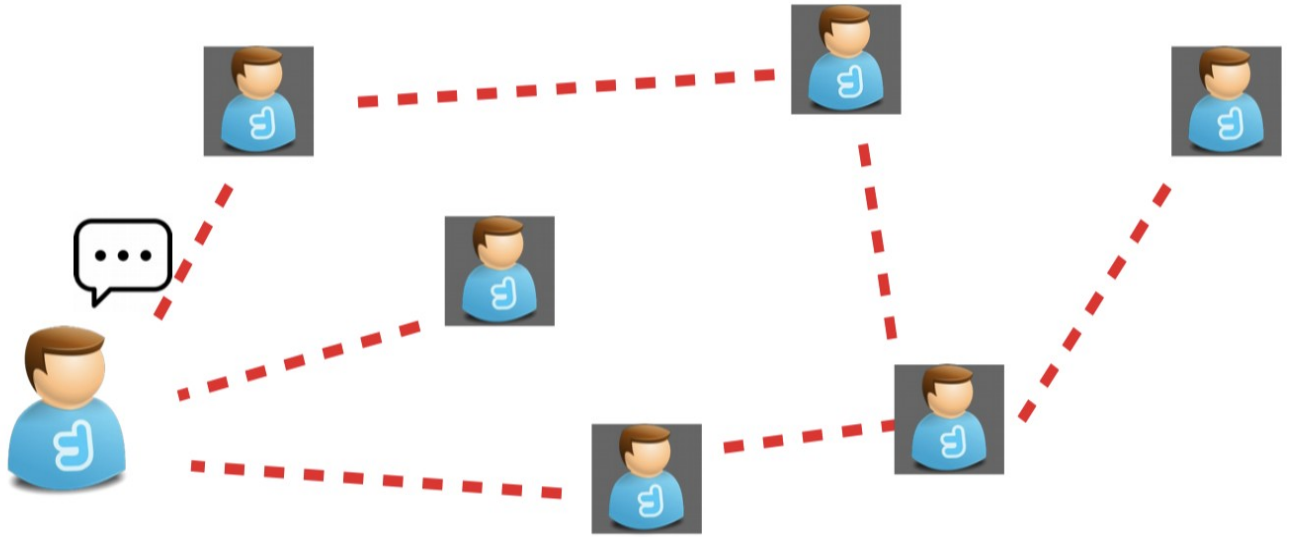


Finance
[Chavez-Demouli & McGil 2012, Bacry et al 2015]

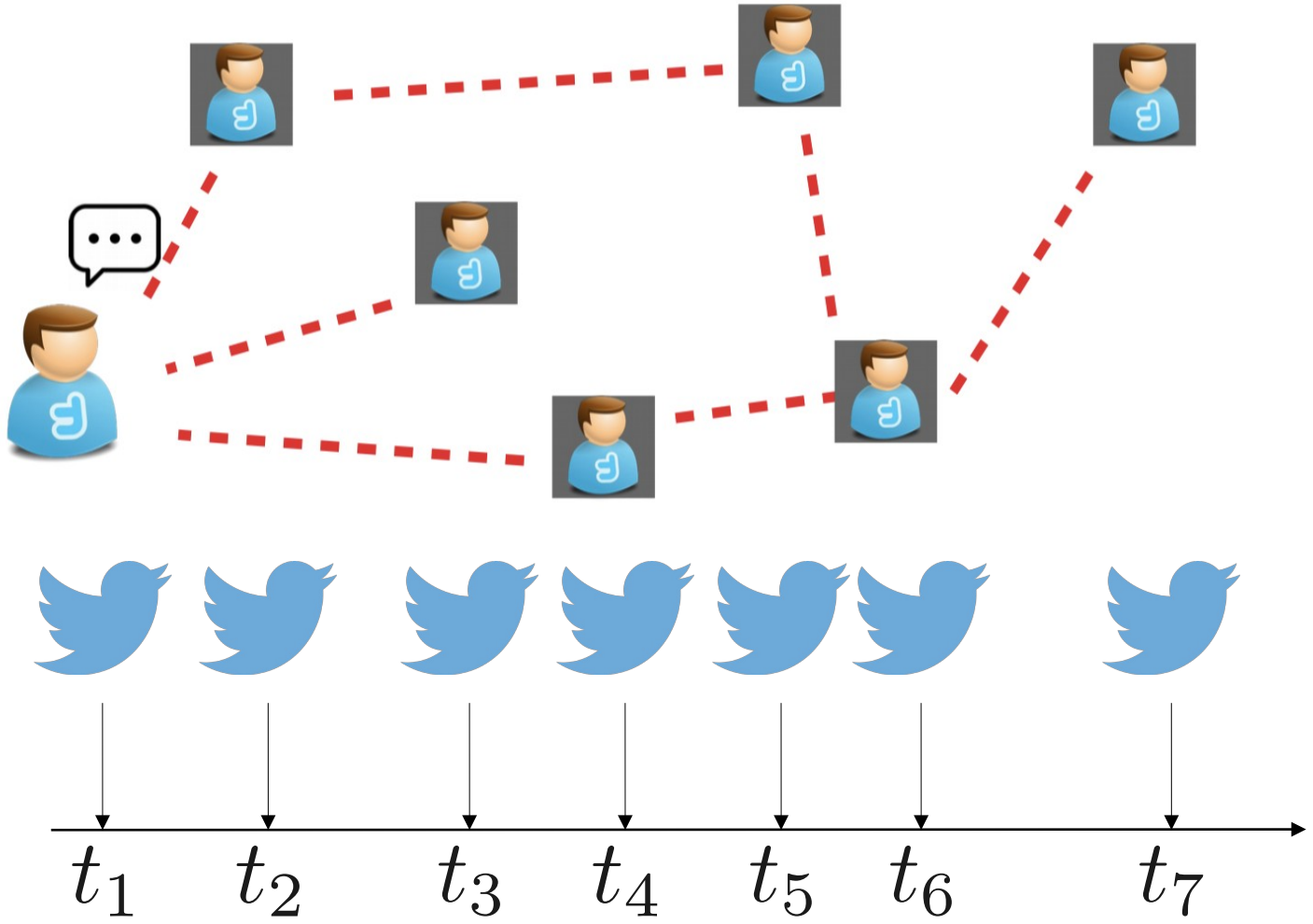


Neural science
[Johnson 1996, Zhou et al 2021]

Reshare cascades in online social platforms



Reshare cascades in online social platforms



Reshare cascades: a collection of time stamped reshare events (retweets) of an online post (tweet).

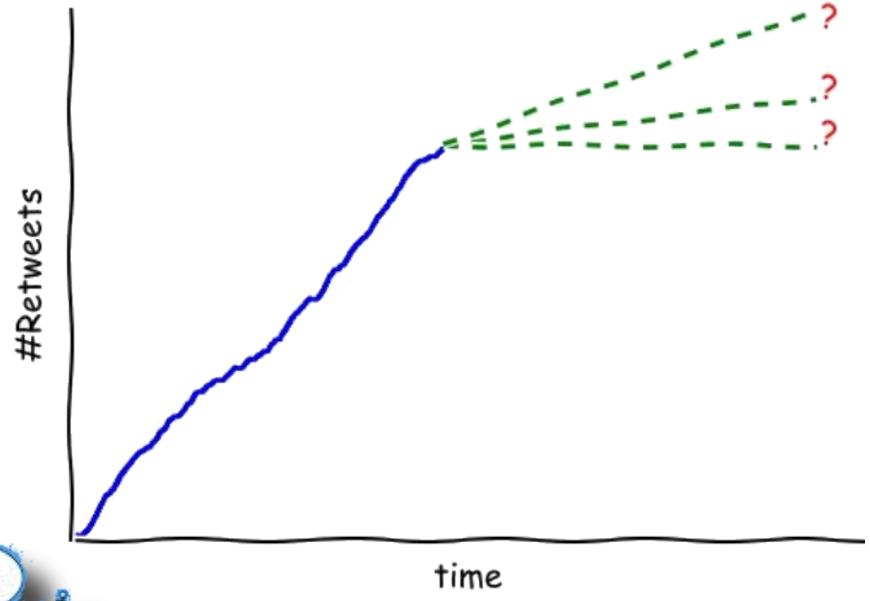
Why do we care about it? Applications

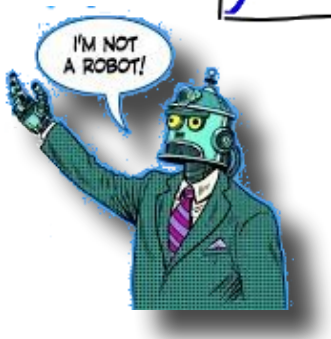
Understand and predict online popularity

Information spread, influence and attention

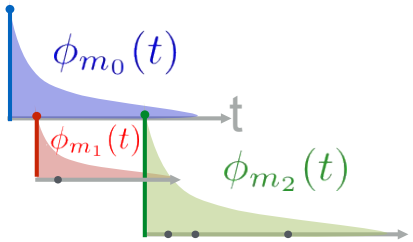
Early detection of inauthentic and coordinated behavior

Understand and detect controversial and “fake” news, mis- and dis-information



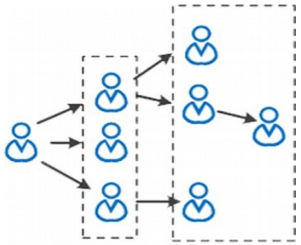


Presentation plan

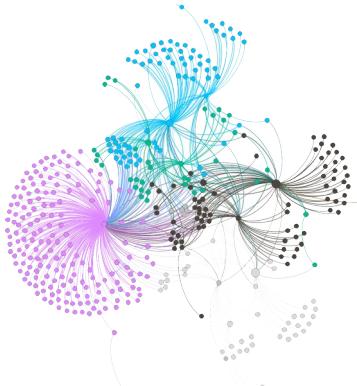


Gentle introduction to Hawkes process theory

Tutorial: Basic Hawkes process operations using *evently*



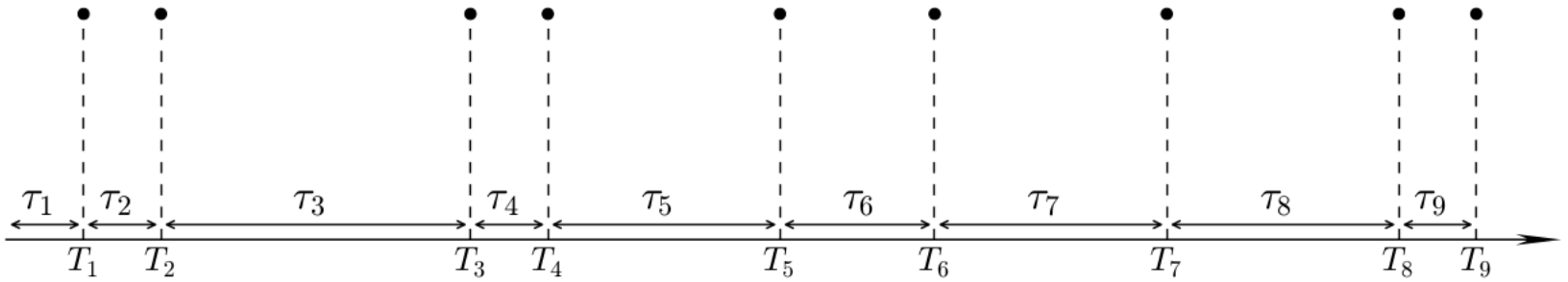
Advanced tutorial: analyze social media using Hawkes (and *evently)*



Applications (using *evently* and *birdspotter)*:

- Detect controversial news sources
- Spot the (influential) socialbot

Point process



A random process – a collection of random variables – the event times

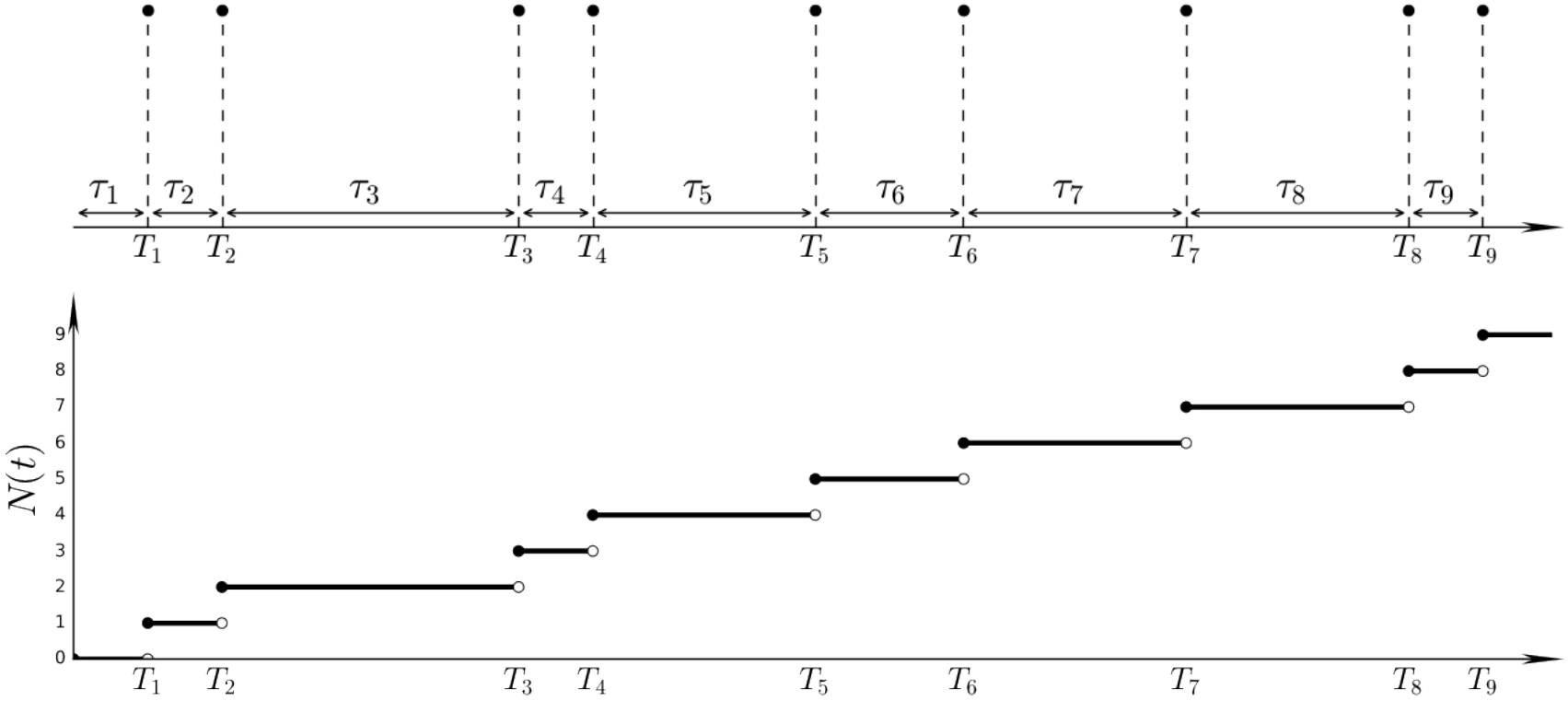
T_i event times

τ_i inter-arrival times

$$T_i := \sum_{j=1}^{i} \tau_j$$

Point process

$$N(t) := \sum_{i \geq 1} 1_{\{t \geq T_i\}}$$



Equivalent counting process. A random function defined on time t ≥ 0, takes integer values 1, 2, .. .
The number of events of the point process by time t

Homogeneous Poisson process

The inter-arrival times are exponentially distributed $\tau_i \sim Exp(\lambda)$

λ is the event intensity of the homogeneous Poisson process

$$f(\tau = t) = \lambda e^{-\lambda t}, \text{ for } t \geq 0$$

Memorylessness property: the probability of having to wait an additional t time units after already having waited m time units is the same as the probability of having to wait t time units when starting at time o

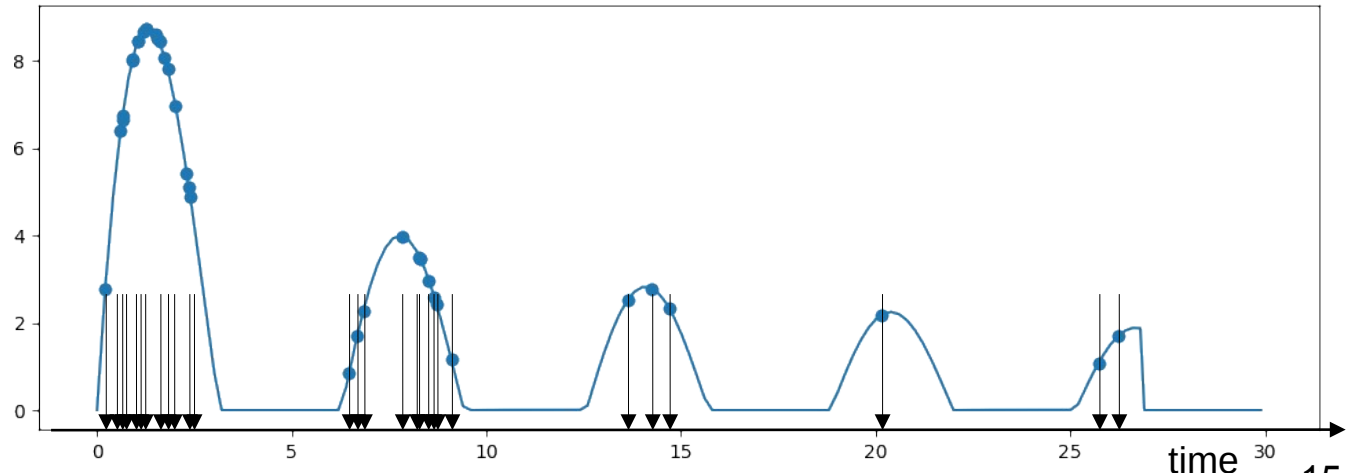
$$\mathbb{P}(\tau > t + m | \tau > m) = \mathbb{P}(\tau > t)$$

Inhomogeneous Poisson process

The rate of event arrivals is a function of time, i.e. $\lambda = \lambda(t)$

The conditional intensity function:

$$\lambda(t|\mathcal{H}_t) = \lim_{h \to 0} \frac{\{N_{t+h} - N_t = 1 | \mathcal{H}_t\}}{h} \qquad \mathcal{H}_t = \{T_1, T_2, ..., T_{N_t}\}$$



Inhomogeneous Poisson process

The rate of event arrivals is a function of time, i.e. $\lambda = \lambda(t)$

The conditional intensity function:

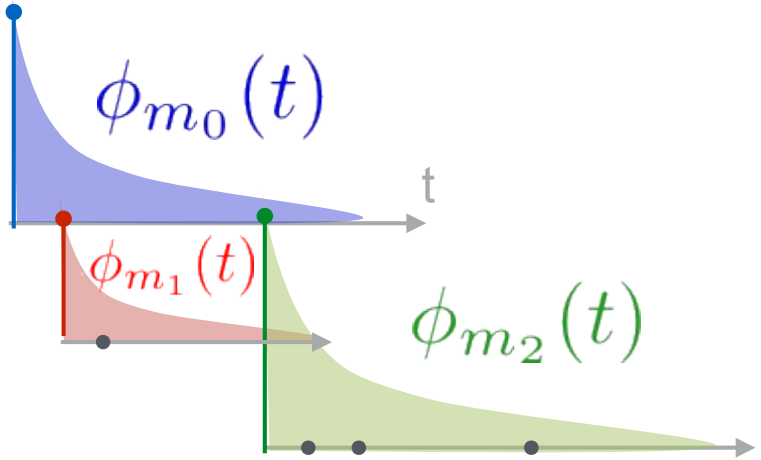
$$\lambda(t|\mathcal{H}_t) = \lim_{h \to 0} \frac{\{N_{t+h} - N_t = 1 | \mathcal{H}_t\}}{h} \qquad \mathcal{H}_t = \{T_1, T_2, ..., T_{N_t}\}$$

Proportional to the probability of observing an event at time t

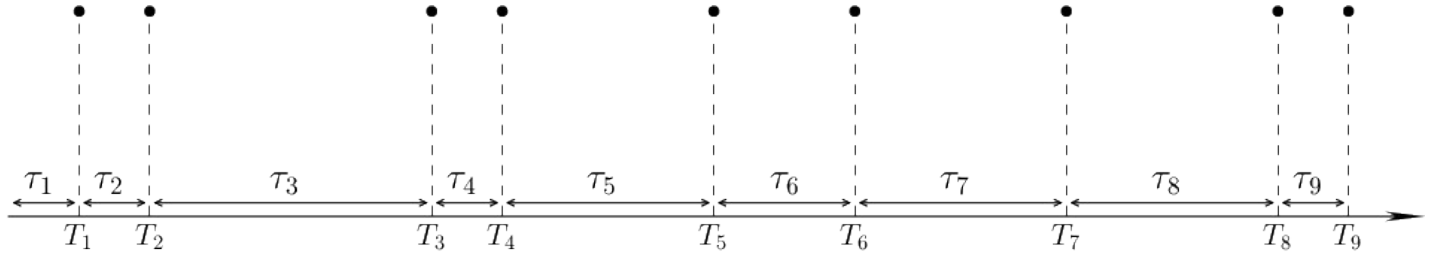
$$\begin{aligned}
&\mathbb{P}(N_{t+h} = n + m \,|\, N_t = n) = \lambda(t)h + o(h) && \text{if} \quad m = 1 \\
&\mathbb{P}(N_{t+h} = n + m \,|\, N_t = n) = o(h) && \text{if} \quad m > 1 \\
&\mathbb{P}(N_{t+h} = n + m \,|\, N_t = n) = 1 - \lambda(t)h + o(h) && \text{if} \quad m = 0
\end{aligned}$$

Hawkes process – definition

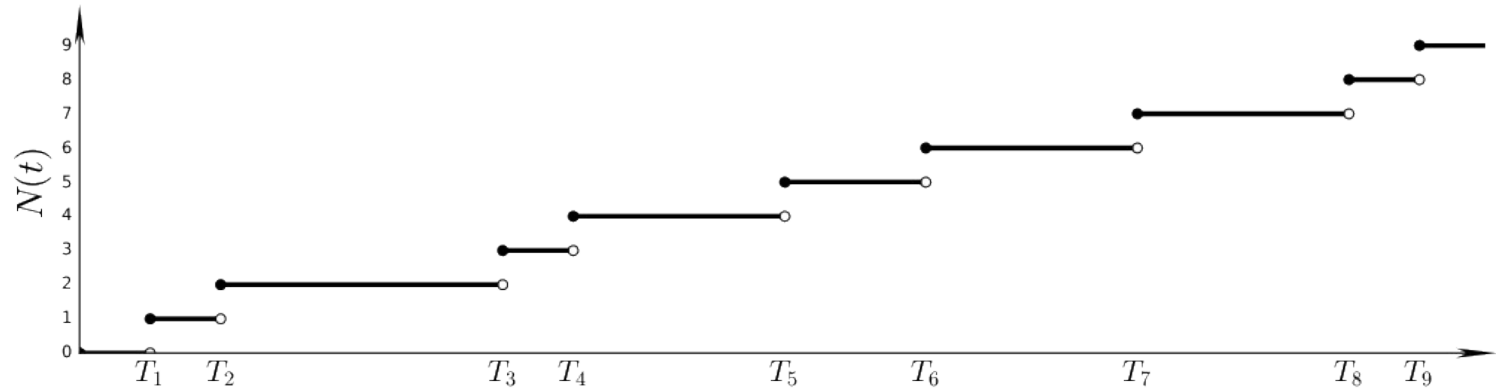
$$\lambda(t|\mathcal{H}_t) = \underbrace{\mu(t)}_{\text{base intensity (exogenous)}} + \underbrace{\sum_{i:t>T_i} \phi(t - T_i)}_{\text{self-excitation (endogenous)}}$$



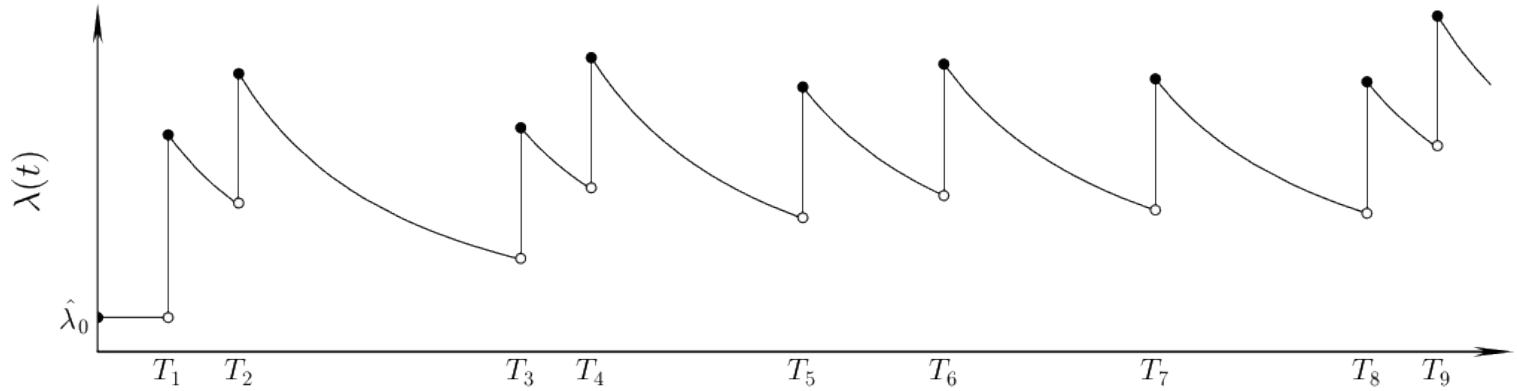
Hawkes process – definition



event times T_i



counting process $N(t)$

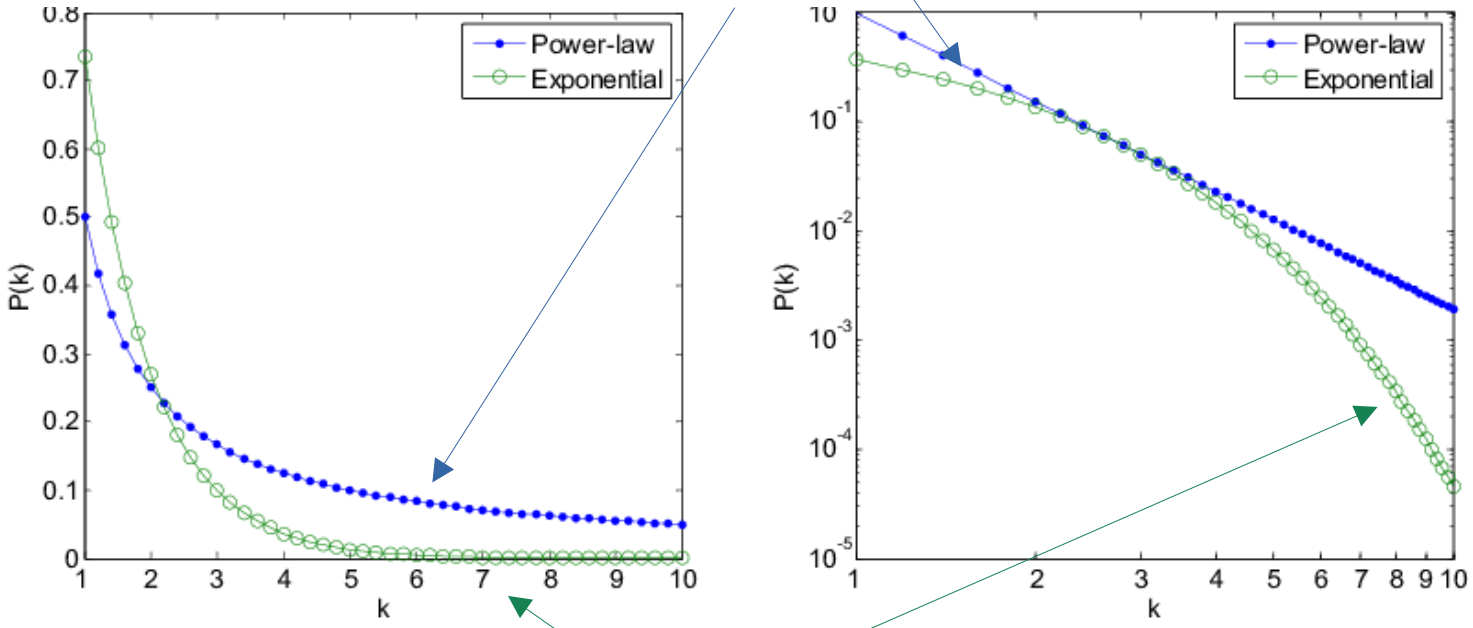


conditional intensity function $\lambda(t|\mathcal{H}_t)$

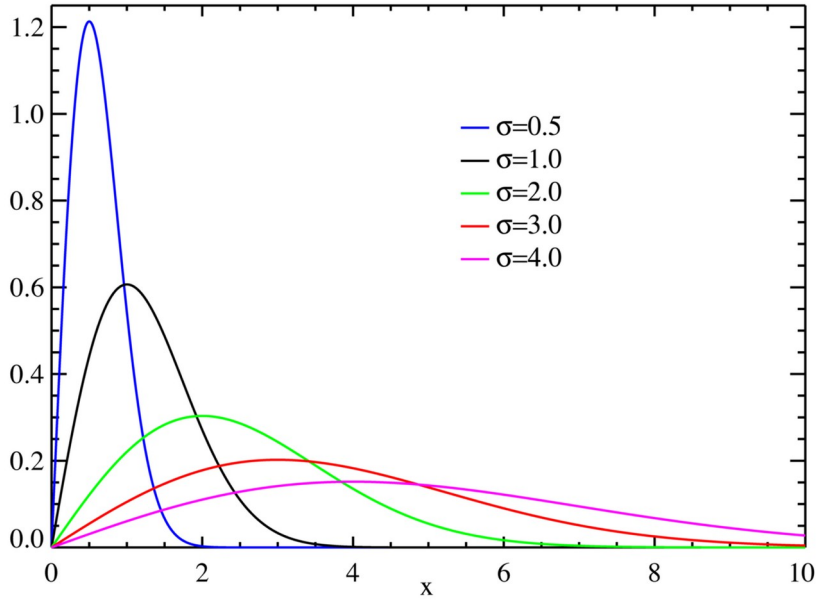
Hawkes process – choice of kernel

Common choices: Exponential, Power-Law, Rayleigh

Power-law has
a long tail.



Exponential
decays very fast.



The “Twitter” kernel

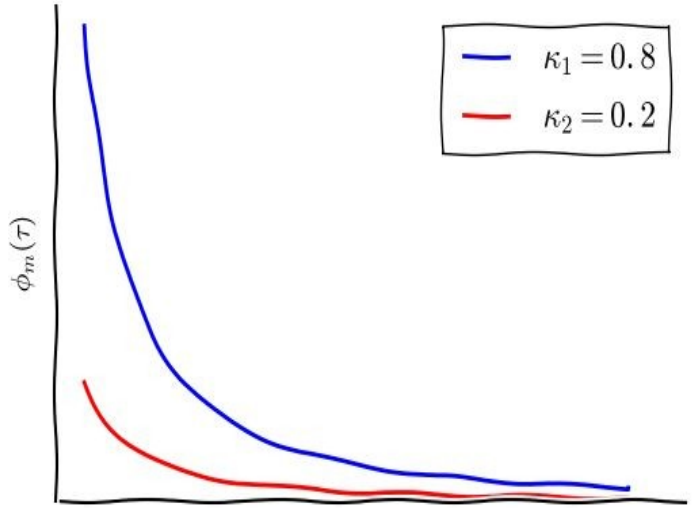
$$\phi_m(t - T_i) = \kappa m^{\beta} (t - T_i)^{-(1+\theta)}$$

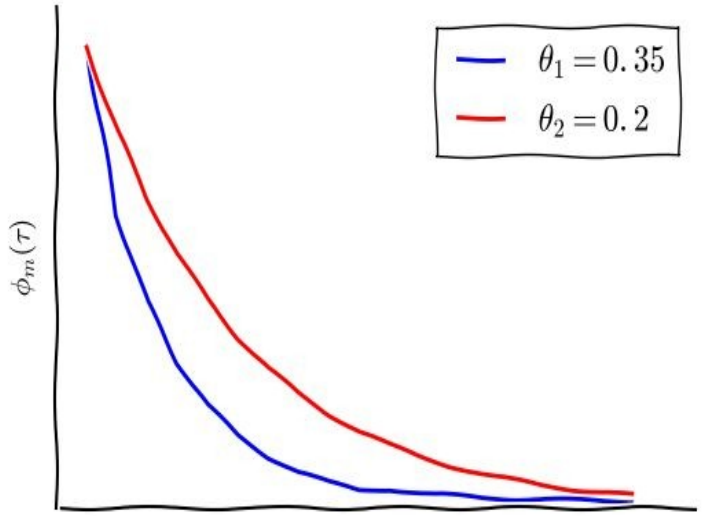
the rate of ‘daughter’ events

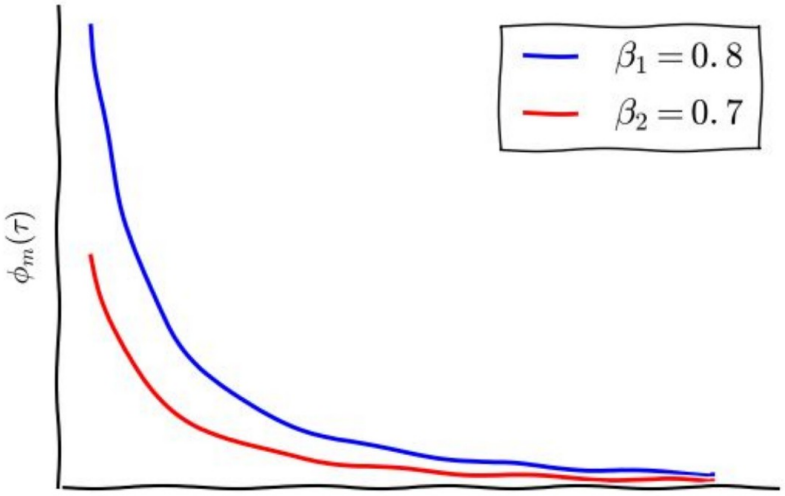
content virality

user influence

memory





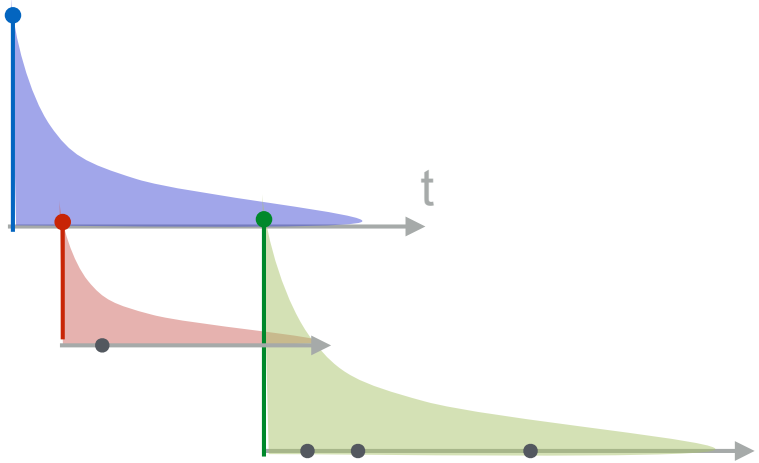


S. Mishra, M.-A. Rizoiu, and L. Xie, "Feature Driven and Point Process Approaches for Popularity Prediction, " in Proc. International Conference on Information and Knowledge Management (CIKM ’16), Indianapolis, USA, p. 1069-1078, 2016.

Hawkes process – branching structure

$$\lambda(t|\mathcal{H}_t) = \mu(t) + \sum_{i:t>T_i} \phi(t - T_i)$$

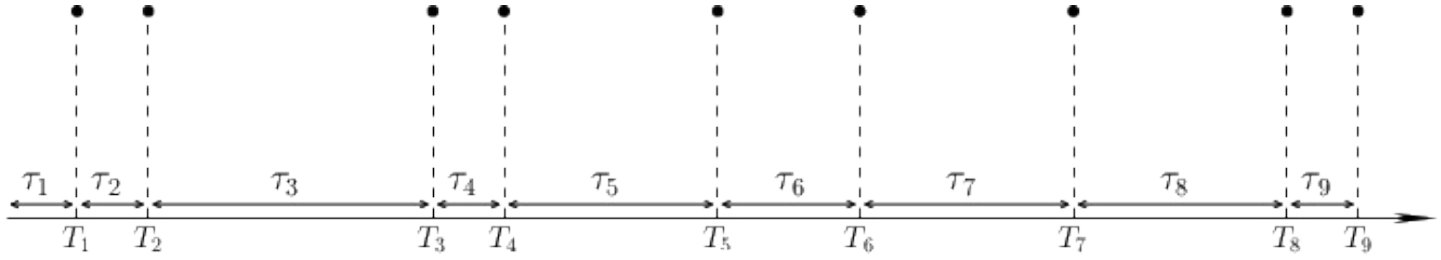
Immigrants arrive following a Poisson process of intensity $\mu(t)$



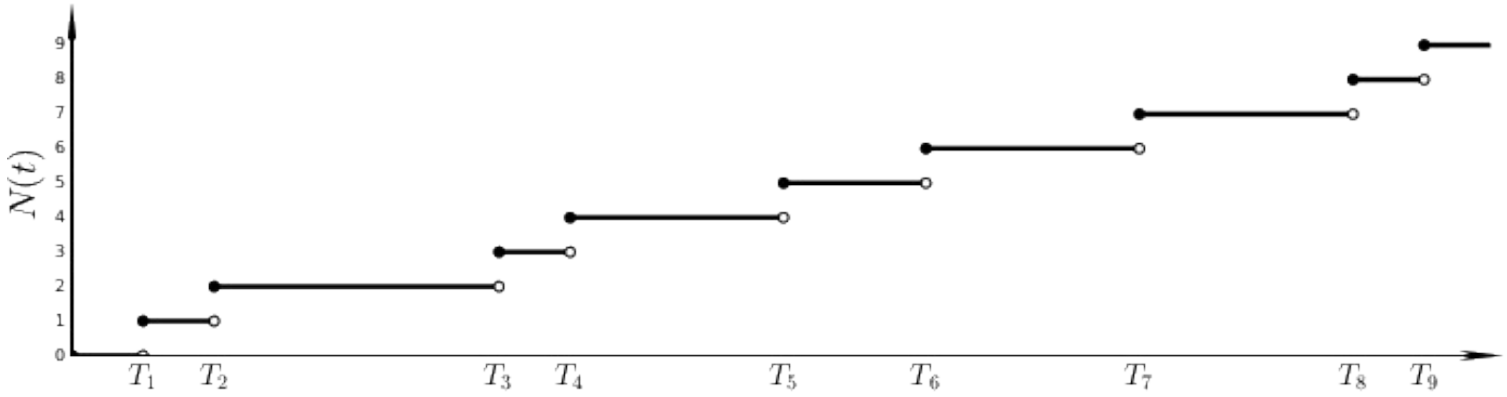
Offspring generation

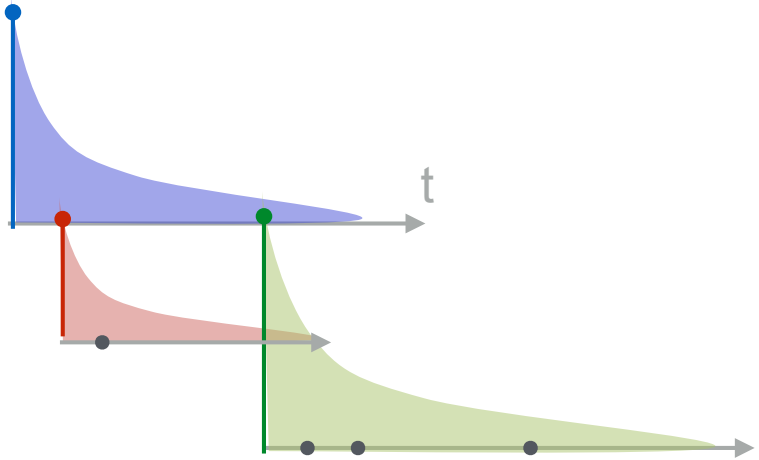
Hawkes process – branching structure

$$\lambda(t|\mathcal{H}_t) = \mu(t) + \sum_{i:t>T_i} \phi(t - T_i)$$



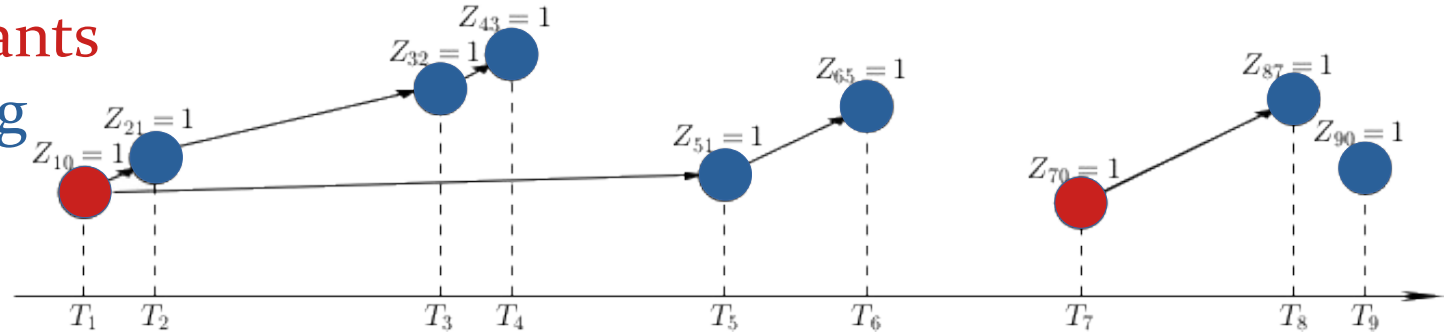
Immigrants arrive following a Poisson process of intensity $\mu(t)$





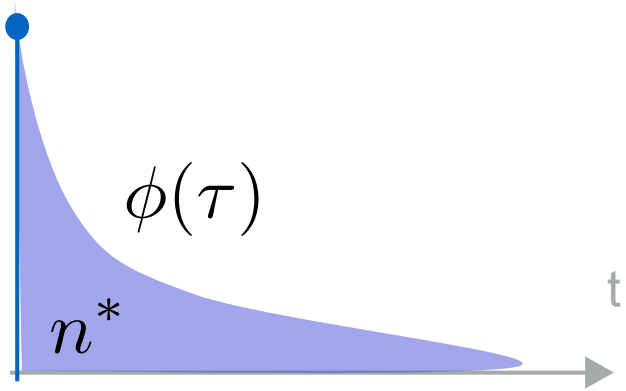
Offspring generation

immigrants
offspring



Hawkes process – branching factor

$$\lambda(t|\mathcal{H}_t) = \mu(t) + \sum_{i:t>T_i} \phi(t - T_i)$$



Branching factor. The expected number direct offspring spawned by a single event.

$$n^* = \int_0^\infty \phi(\tau) d\tau$$

$n^* < 1$ Subcritical regime

$n^* > 1$ Super-critical regime – infinite number of events

Compute the popularity – final size

[Mishra et al 2015]

Final popularity $N_\infty = \sum_{i=0}^{\infty} A_i$

A_i – size of each generation of offspring

Expected number of events in a generation

$$A_i = A_{i-1}\, n^* = A_{i-2}\, (n^*)^2 = \ldots = A_0\, (n^*)^i = (n^*)^i\,, i \geq 1$$

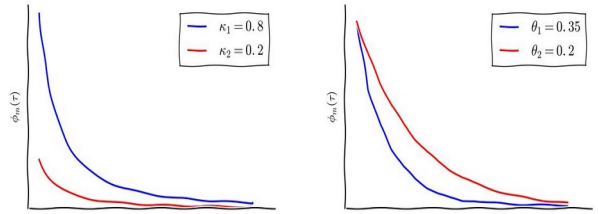
The final popularity is the sum of a converging geometric progression $(n^* < 1)$

$$N_\infty = \sum_{i=0}^{\infty} A_i = 1 + n^* + (n^*)^2 + (n^*)^3 + \ldots = \frac{1 - (n^*)^\infty}{1 - n^*} \simeq \frac{1}{1 - n^*}$$

Mishra, S., Rizoiu, M.-A., & Xie, L. (2016). Feature Driven and Point Process Approaches for Popularity Prediction. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management - CIKM '16 (pp. 1069–1078). Indianapolis, IN, USA: ACM Press.

Hawkes process – Fitting parameters

$$\phi_m(t - T_i) = \kappa m^{\beta}(t - T_i + c)^{-(1+\theta)}$$



Hawkes process – Fitting parameters

$$\phi_m(t - T_i) = \kappa m^{\beta}(t - T_i + c)^{-(1+\theta)}$$

Maximize the log-likelihood function
(usually quadratic and non-convex)

$$\log L(\theta) = \sum_{i=1}^{N(T)} \log \lambda(T_i) - \int_0^T \lambda(t)dt$$

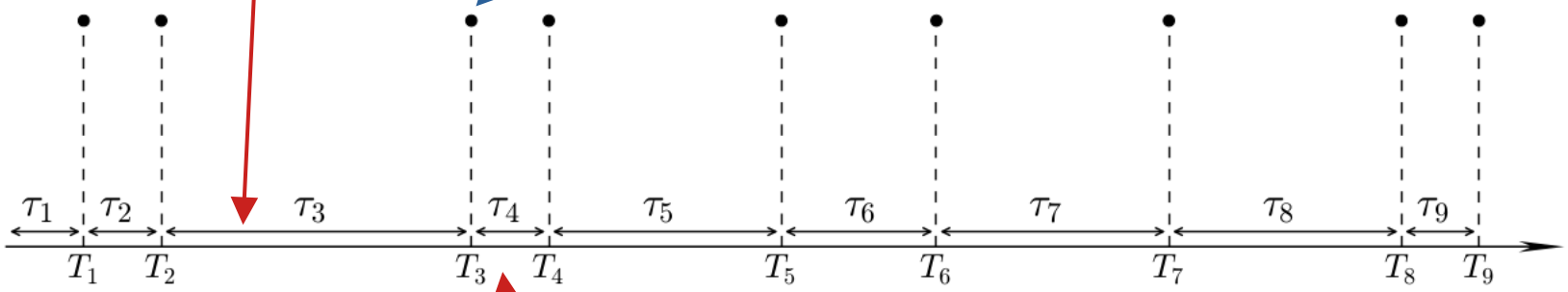
Hawkes process – Fitting parameters

$$\phi_m(t - T_i) = \kappa m^{\beta}(t - T_i + c)^{-(1+\theta)}$$

Maximize the log-likelihood function
(usually quadratic and non-convex)

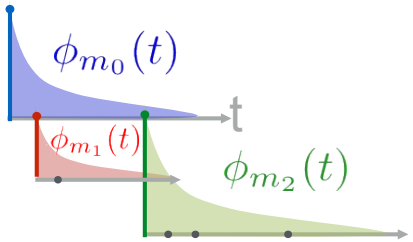
$$\log L(\theta) = \sum_{i=1}^{N(T)} \log \lambda(T_i) - \int_0^T \lambda(t)dt$$

$$\sim \int_{T_2}^{T_3} \lambda(\tau)d\tau \qquad \sim \lambda(T_3)$$



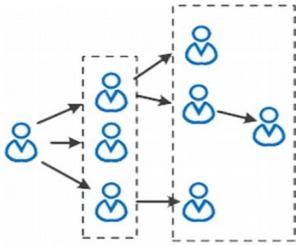
$$\sim \int_{T_3}^{T_4} \lambda(\tau)d\tau$$

Presentation plan

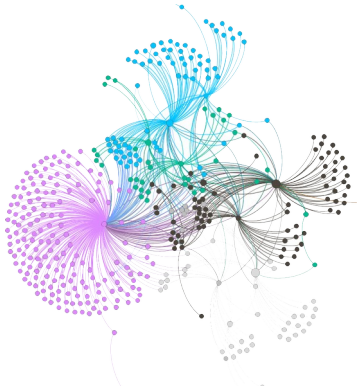


Gentle introduction to Hawkes process theory

Tutorial: Basic Hawkes process operations using *evently*



Advanced tutorial: analyze social media using Hawkes (and *evently)*



Applications (using *evently* and *birdspotter)*:

- Detect controversial news sources
- Spot the (influential) socialbot

evently – a point-process toolbox

Maintainer: Quyu Kong
quyu.kong@uts.edu.au
https://github.com/behavioral-ds/evently

A self-exciting process toolbox in the native R language with limited dependencies, to fit and simulate from around 20 variants including

- Kernel functions (exponential, power-law, Tsallis q-exponential) [Hawkes 1971, Mishra et al 2015, Lima et al 2018]
- Exogeneous functions [Rizoiu et al 2017]
- Marked and unmarked [Mishra et al 2015]
- Finite population [Rizoiu et al 2018, Kong et al 2020]
- Joint fitting [Kong et al 2020]

evently – supported Hawkes kernels

Model	Abbreviation (model_type)	Intensity Function	Parameters
Hawkes process with an exponential kernel function	EXP	$\kappa \sum_{t_i<t} \theta e^{-\theta(t-t_i)}$	K,theta
Hawkes process with a power-law kernel function	PL	$\kappa \sum_{t_i<t} (t-t_i+c)^{-(1+\theta)}$	K,c,theta
HawkesN process with an exponential kernel function	EXPN	$\kappa \frac{N-N_t}{N} \sum_{t_i<t} \theta e^{-\theta(t-t_i)}$	K,theta,N
HawkesN process with a power-law kernel function	PLN	$\kappa \frac{N-N_t}{N} \sum_{t_i<t} (t-t_i+c)^{-(1+\theta)}$	K,c,theta,N
Marked Hawkes process with an exponential kernel function	mEXP	$\kappa \sum_{t_i<t} \theta m_i^{\beta} e^{-\theta(t-t_i)}$	K,beta,theta
Marked Hawkes process with a power-law kernel function	mPL	$\kappa \sum_{t_i<t} m_i^{\beta} (t-t_i+c)^{-(1+\theta)}$	K,beta,c,theta
Marked HawkesN process with an exponential kernel function	mEXPN	$\kappa \frac{N-N_t}{N} \sum_{t_i<t} \theta m_i^{\beta} e^{-\theta(t-t_i)}$	K,beta,theta,N
Marked HawkesN process with a power-law kernel function	mPLN	$\kappa \frac{N-N_t}{N} \sum_{t_i<t} m_i^{\beta} (t-t_i+c)^{-(1+\theta)}$	K,beta,c,theta,N

Twitter kernel

evently – a point-process toolbox

Maintainer: Quyu Kong
quyu.kong@uts.edu.au
https://github.com/behavioral-ds/evently

Designed with an emphasis on online information diffusion modeling.

- Data processing tools for Twitter
- Popularity measures on online information diffusions
- Diffusion embeddings for online users and online content

evently – parameter fitting

Constrained optimization problem.
For the Twitter kernel:

Maximize:

$$\mathcal{L}(\kappa, \beta, c, \theta) = \sum_{i=2}^{n} \log \kappa + \sum_{i=2}^{n} \log \left(\sum_{t_j < t_i} \frac{(m_j)^{\beta}}{(t_i - t_j + c)^{1+\theta}} \right)$$
$$- \kappa \sum_{i=1}^{n} (m_i)^{\beta} \left[\frac{1}{\theta c^{\theta}} - \frac{(T + c - t_i)^{-\theta}}{\theta} \right]$$

Subject to:

$$n^* = \kappa \frac{\alpha - 1}{\alpha - \beta - 1} \frac{1}{\theta c^{\theta}} < 1, \beta < \alpha - 1 \text{ and } \theta > 0$$

evently – parameter fitting

Uses state-of-the-art nonlinear optimizer for learning parameters

- Difficulties: complex objective functions and constrained-optimization problem
- AMPL for efficiently defining optimization problem and connecting to a wide range of global and local solvers.

ampl.com/

IP*Opt*

github.com/coin-or/Ipopt

LGO

www.maximalsoftware.com/solvers/lgo.html

Installation and setup

- Install with **devtools::install_github("behavioral-ds/evently")**
- Simplified installation of AMPL and solvers

```
library(evently)
```

```
*********************************************************
  This package requires AMPL and ipopt
  But they are not in your PATH environment.
  Please specify its binary folder path in ~/.Renviron
  Please also make sure the ipopt binary is in the
  same folder.
*********************************************************

It seems AMPL is not found in your PATH environment, do you want to install it now?
```

```
1: yes
2: no
```

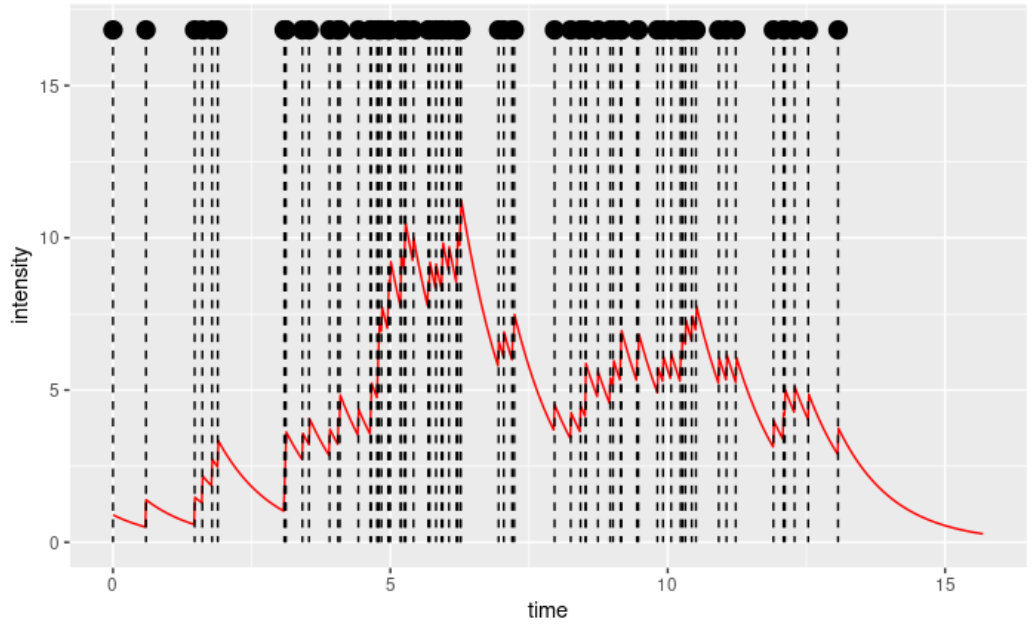
Sampling Hawkes realizations

Select kernel type, parameter set, number of realizations and max simulation time.

```
set.seed(4)
sim_no <- 100
model <- new_hawkes(par = c(K = 0.9, theta = 1), model_type = 'EXP')
data <- generate_series(model = model, Tmax = 15, sim_no = sim_no)
# alternatively, `generate_series` also accepts directly the parameters, without defining a model
# e.g.
# data <- generate_series(par = c(K = 0.9, theta = 1), model_type = 'EXP', Tmax = 15, sim_no = sim_no)

plot_event_series(model = model, cascade = data[1] )
```

Here, sample 100 cascades from an Exponential Hawkes of parameters

$$\kappa = 0.9, \theta = 1$$



Fitting model parameters to data

One line of code for:

- Multiple solvers (local and global)
- Multiple initial points to address non-convexity
- Multi-processor parallel fitting

```
fitted_model <- fit_series(data, model_type = 'EXP', observation_time = 15, cores = 10)
```

Fitting model parameters to data

One line of code for:

- Multiple solvers (local and global)
- Multiple initial points to address non-convexity
- Multi-processor parallel fitting

```
fitted_model <- fit_series(data, model_type = 'EXP', observation_time = 15, cores = 10)
```

```
fitted_model
```

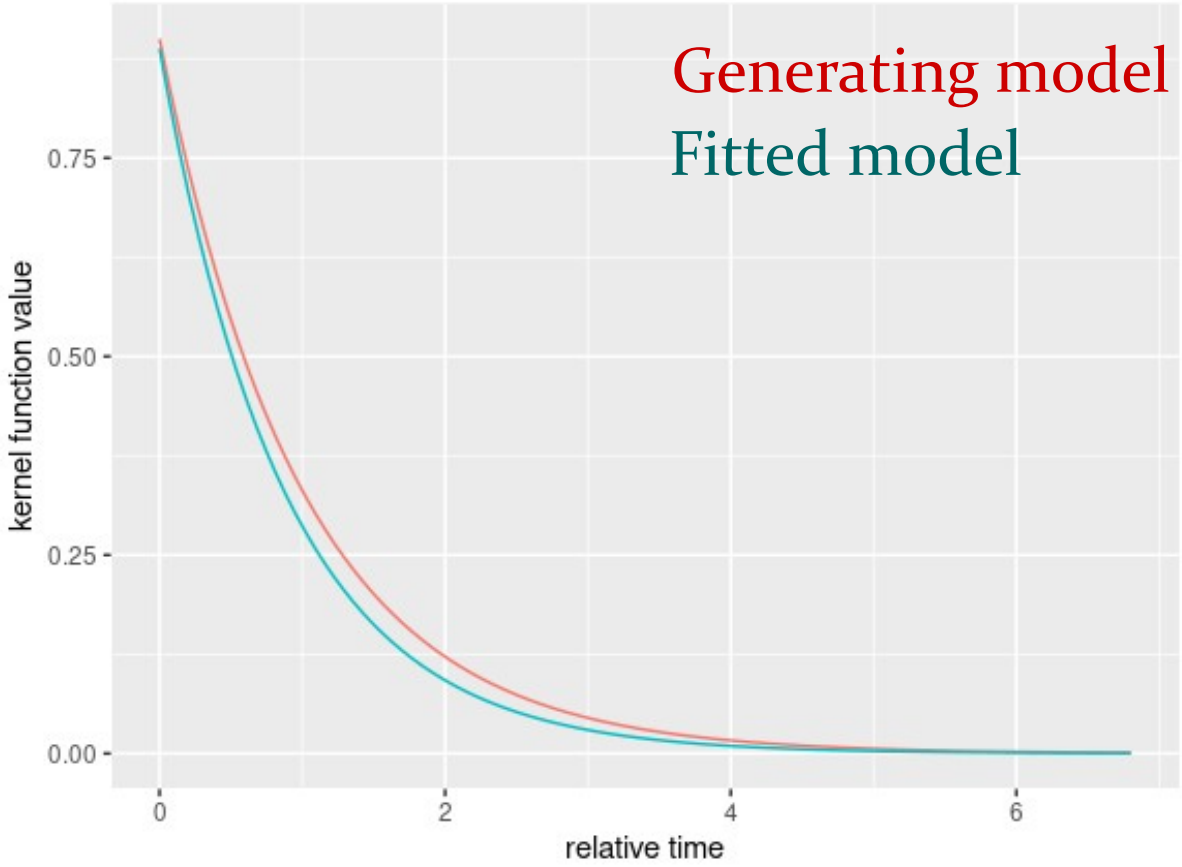
```
- Model: EXP
- No. of cascades: 100
- init_par:
  K 8.75e+00; theta 2.32e+00
- par:
  K 7.84e-01; theta 1.13e+00
- Neg Log Likelihood: 245.7
- lower_bound:
  K 1.00e-100; theta 1.00e-100
- upper_bound:
  K 1.00e+04; theta 3.00e+02
- Convergence: 0
```

Fitted model

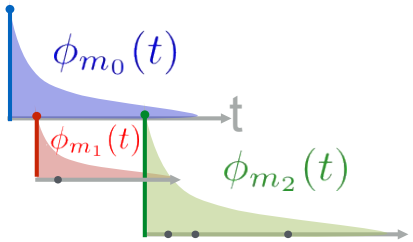
```
model
```

```
- Model: EXP
- init_par:
  K NA; theta NA
- par:
  K 9.00e-01; theta 1.00e+00
- Neg Log Likelihood: NA
- lower_bound:
  K 1.00e-100; theta 1.00e-100
- upper_bound:
  K 1.00e+04; theta 3.00e+02
```

Generating model

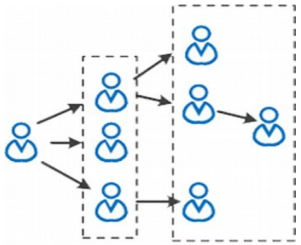


Presentation plan

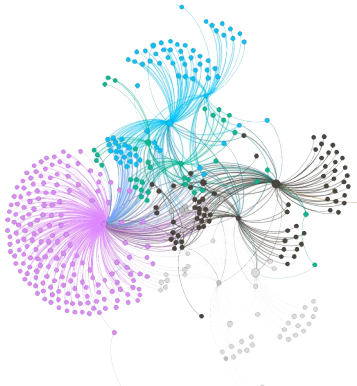


Gentle introduction to Hawkes process theory

Tutorial: Basic Hawkes process operations using *evently*



Advanced tutorial: analyze social media using Hawkes (and *evently)*



Applications (using *evently* and *birdspotter)*:

- Detect controversial news sources
- Spot the (influential) socialbot

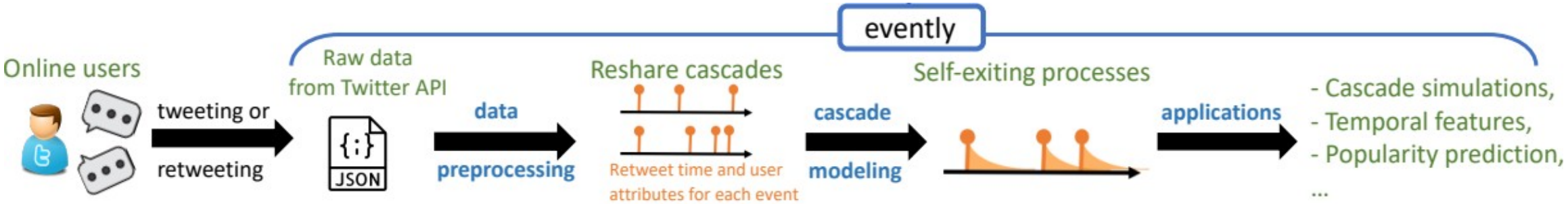
Real-world COVID Twitter dataset

- The dataset used is tweets concerning the coronavirus COVID-19 pandemic in 2020 collected by [Chen et al 2020].
- Characterizing the influence of Twitter users, such as:

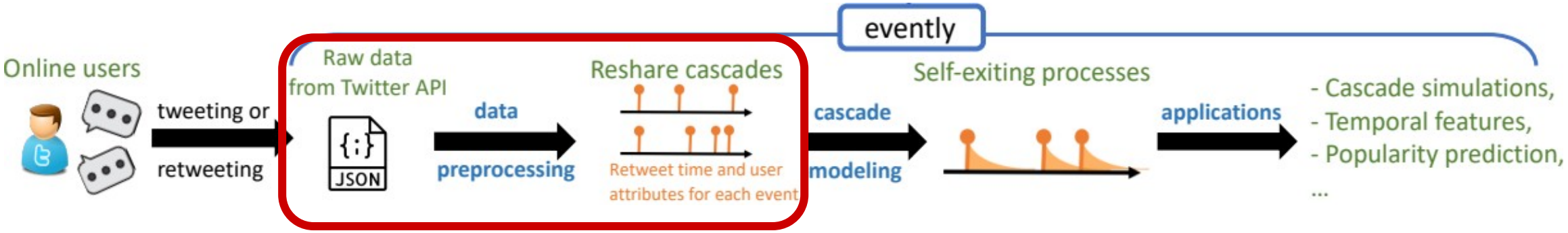
Fan of a Filipino author

Korean pop star

Data analysis pipeline using *evently*



Raw data processing




```
{
 "created_at": "Wed Oct 10 20:19:24 +0000 2018",
 "id": 1050118621198921728,
 "id_str": "1050118621198921728",
 "text": "To make room for more expression, we will
now count all emojis as equal—including those with
gender and skin t… https://t.co/MkGjXf9aXm",
 "user": {},
 "entities": {}
}
```

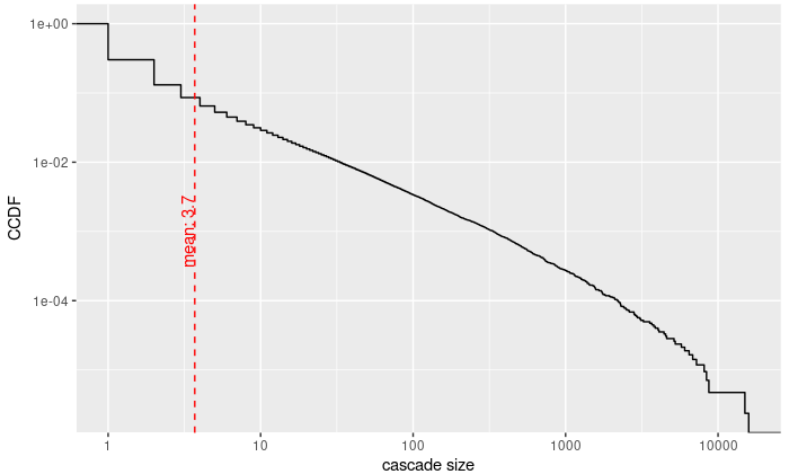
```
cascades ← parse_raw_tweets_to_cascades('corona_2020_01_31.jsonl',
                                        keep_user = TRUE,
                                        keep_absolute_time = TRUE)
```

Dedicated data processing functions for raw data crawled from Twitter APIs.

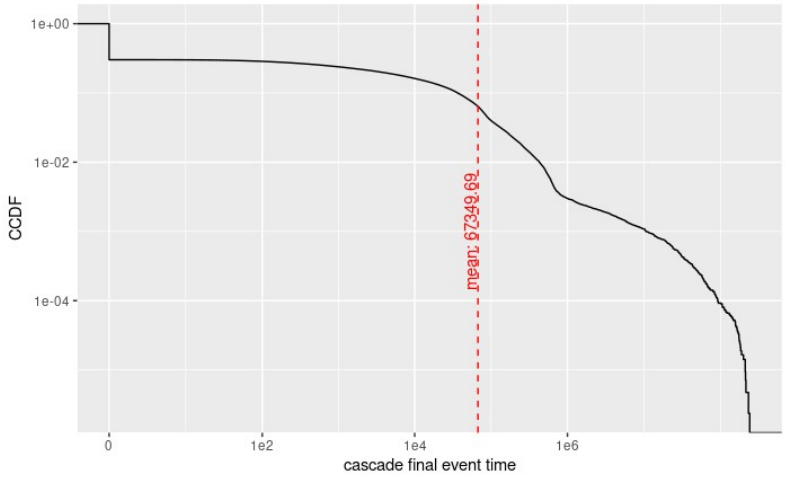
Dataset profiling

COVID dataset: [Chen et al 2020]

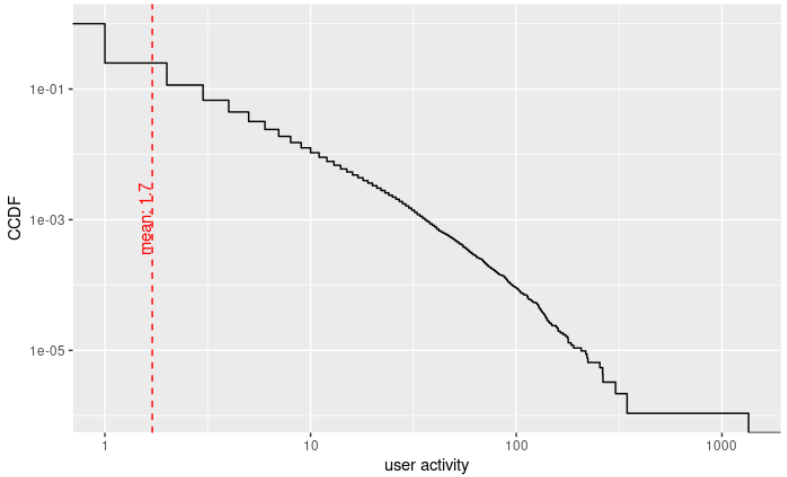
- 31 Jan 2020 (1 day)
- 1.5M unique tweets
- 280K users
- 420K cascades



Cascade size
(# tweets)

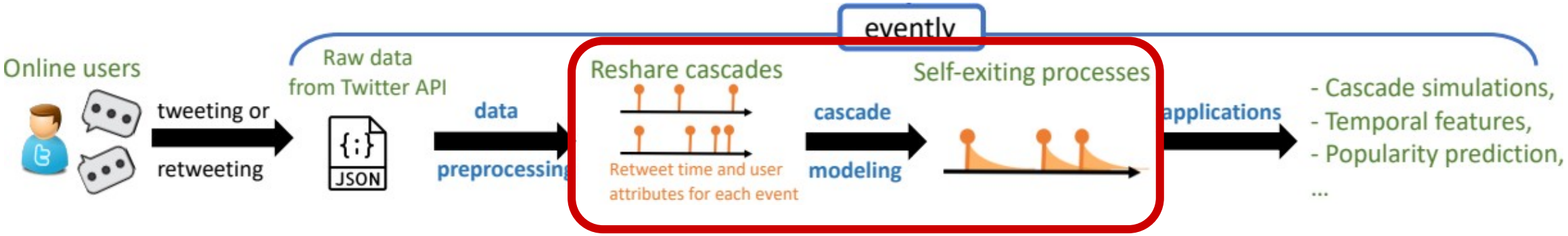


Cascade length
(seconds)



User activity (# cascades
they participate)

Dual mixture model parameter fitting



Joint cascade fitting – all cascades initiated by a user.
Parameters describe groups of cascades, i.e. the user.

Joint modeling using mixture models

[Kong et al 2020]

Kong, Q., Rizoiu, M.-A., & Xie, L. (2020). Describing and Predicting Online Items with Reshare Cascades via Dual Mixture Self-exciting Processes. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 645–654). New York, NY, USA: ACM.

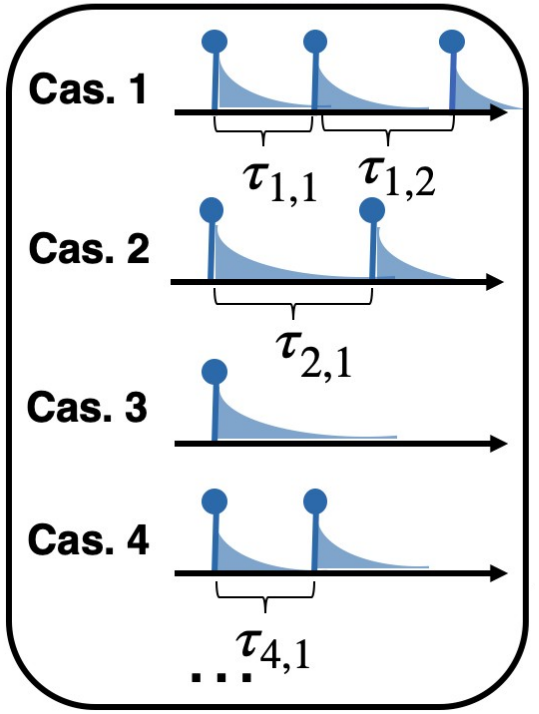
Joint modeling using mixture models

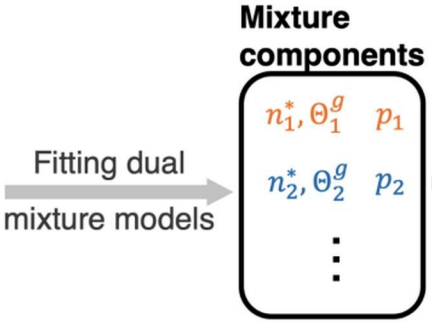
[Kong et al 2020]

Mixtures of latent sets of parameters describe the group of cascades

Log-likelihood separable over the two parameter mixtures:

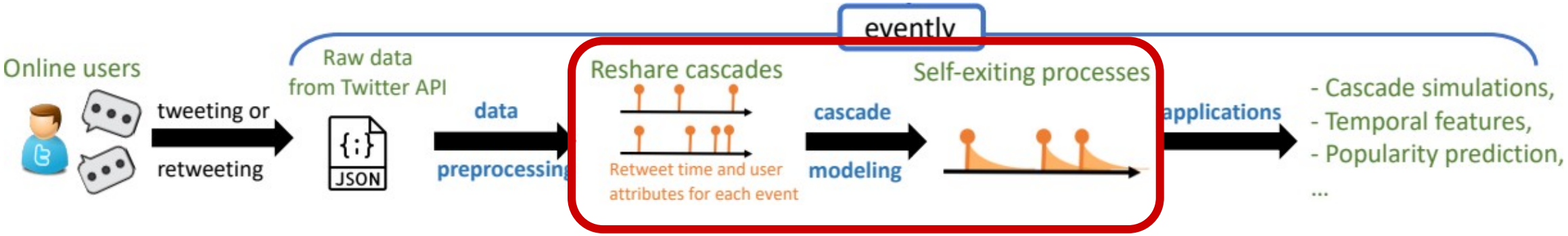
$$\mathcal{L}(n^*, \Theta^g \mid \mathbb{H}) = \mathcal{L}_g(\Theta^g \mid \mathbb{H}) + \mathcal{L}_n(n^* \mid \mathbb{H})$$





Kong, Q., Rizoiu, M.-A., & Xie, L. (2020). Describing and Predicting Online Items with Reshare Cascades via Dual Mixture Self-exciting Processes. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 645–654). New York, NY, USA: ACM.

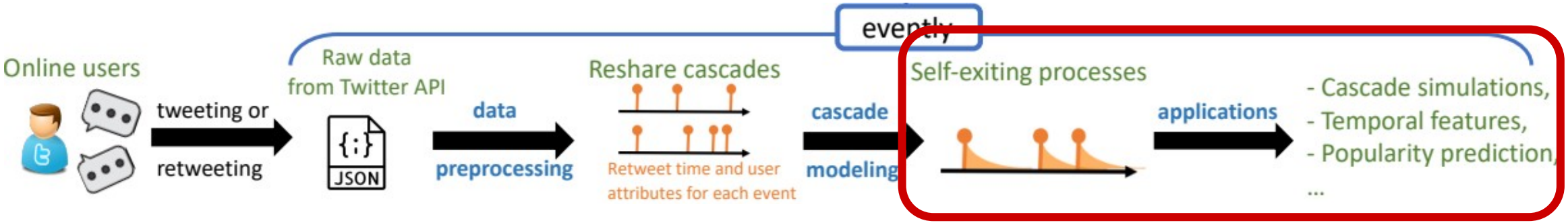
Dual mixture model parameter fitting



Unified APIs for fitting reshare cascades with different variants of Hawkes processes.

```
# model fitting
user_cascades_fitted ← fit_series(data = selected_cascades,
                                  model_type = 'mPL',
                                  observation_time = times, cores = 10)

user_cascades_fitted ← fit_series(data = selected_cascades,
                                  model_type = 'SEISMIC',
                                  observation_time = times, cores = 10)
```

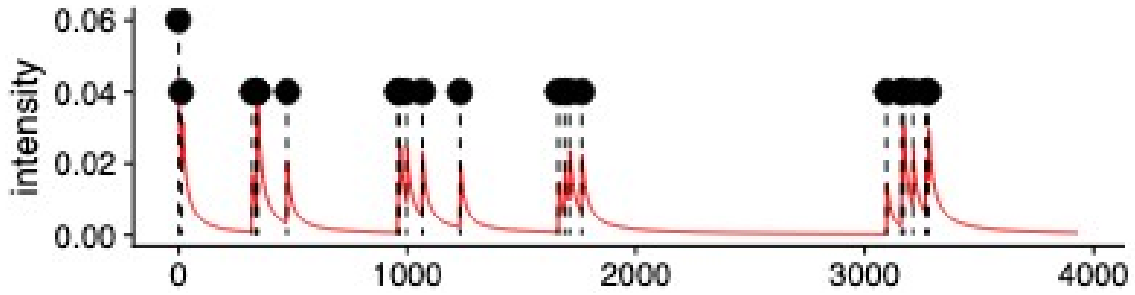
Cascade modeling & analyses



Sampling synthetic cascades from users

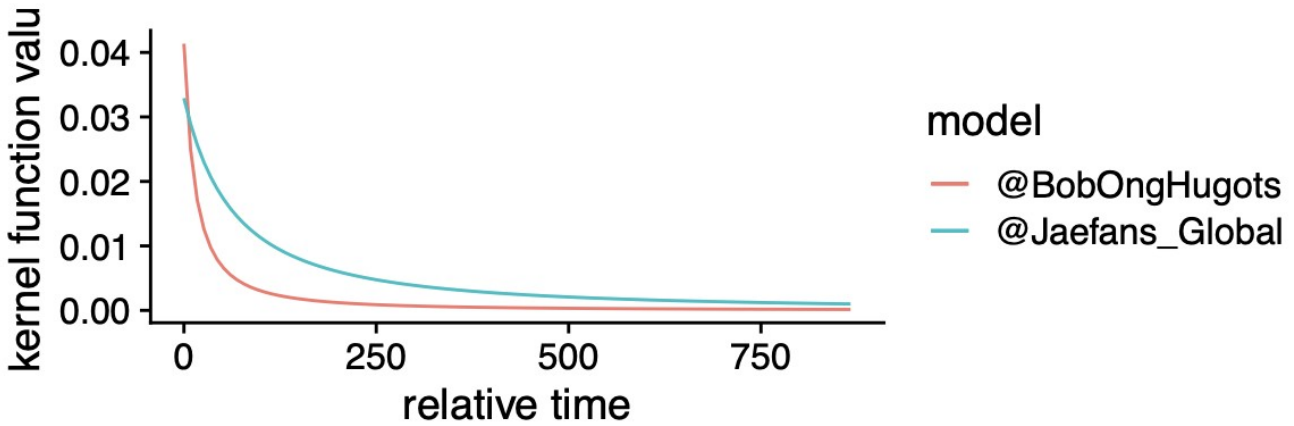
Sample a cascade from the dual mixture model – a hypothetical cascade from a given user.

```
# simulate a new cascade from @BobOngHugots
sim_cascade ← generate_series(user_cascades_fitted, M = user_magnitude)
plot_event_series(cascade = sim_cascade, model = user_cascades_fitted)
```



Inspecting learned models

```
1 plot_kernel_function(user_cascades_fitted) +
2   scale_color_discrete(labels = c("@BobOngHugots", "@Jaefans_Global"))
```



Producing popularity measures

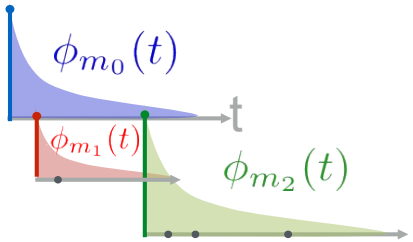
```
predict_final_popularity(user_cascades_fitted, selected_cascade, selected_time)
#> 458.302810

# predict with SEISMIC model, assume we have fitted the SEISMIC model
predict_final_popularity(user_cascades_SEISMIC_fitted, selected_cascade, selected_time)
#> 729.923

get_branching_factor(user_cascades_fitted)
#> 0.7681265

get_viral_score(user_cascades_fitted)
#> 7.408275
```

N_∞

n^*

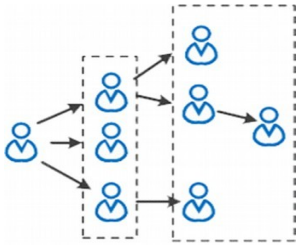
$\nu = \mathbb{E}_{\mathcal{H}_\infty}[N_\infty]$ Expected size of a cascade started by a given user.

Presentation plan

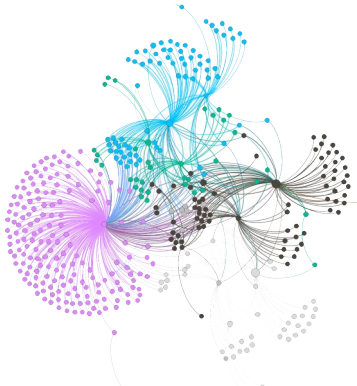


Gentle introduction to Hawkes process theory

Tutorial: Basic Hawkes process operations using *evently*



Advanced tutorial: analyze social media using Hawkes (and *evently)*



Applications (using *evently* and *birdspotter)*:

- **Detect controversial news sources**
- **Spot the (influential) socialbot**

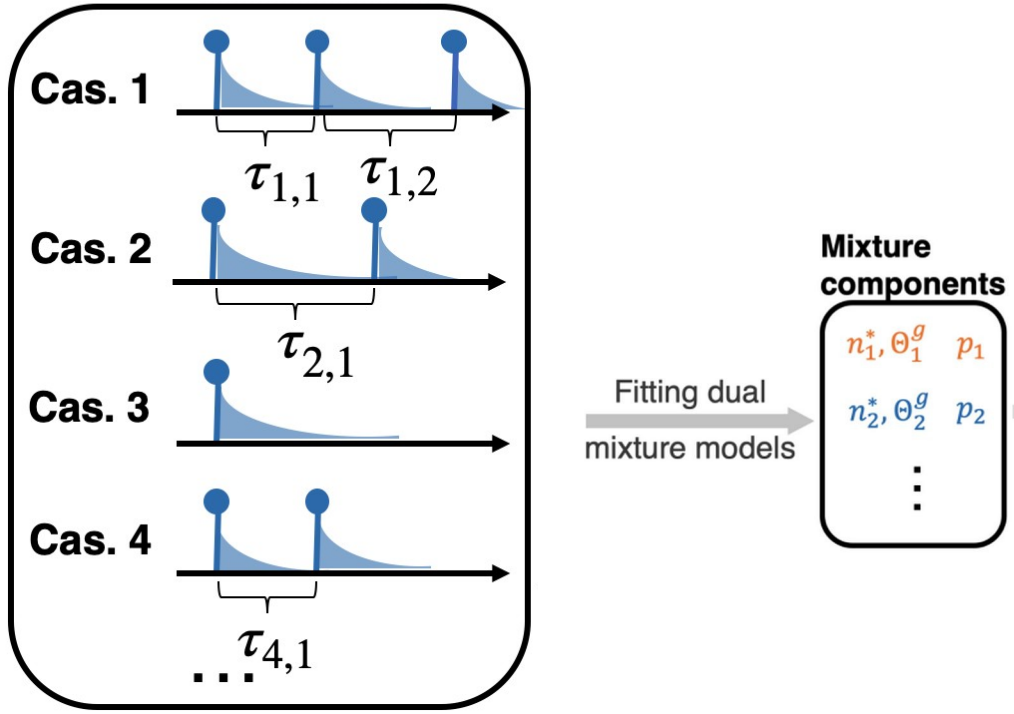
(1) Separating controversial from reputable

Two retweet cascade datasets:

- ATNIX: Australian Twitter News Index
- CNIX: Controversial News Index

(1) Build publisher embeddings from mixture models

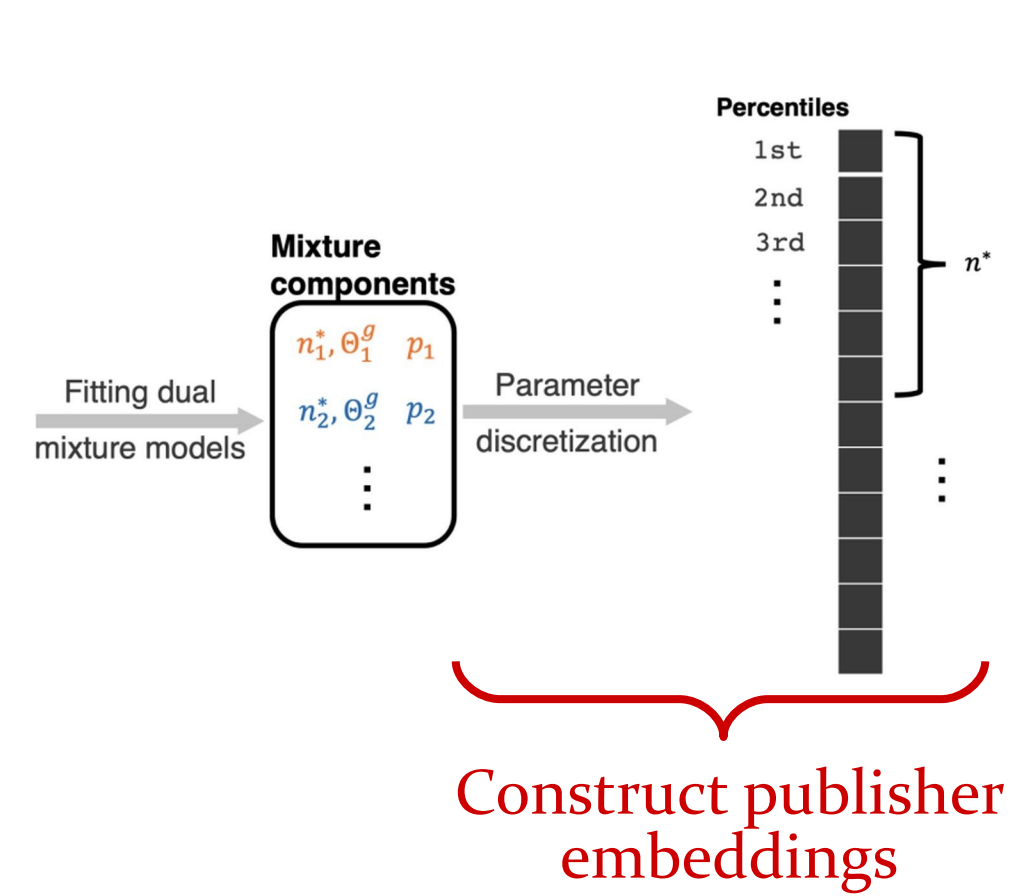
[Kong et al 2020]



Kong, Q., Rizoiu, M.-A., & Xie, L. (2020). Describing and Predicting Online Items with Reshare Cascades via Dual Mixture Self-exciting Processes. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 645–654). New York, NY, USA: ACM.

(1) Build publisher embeddings from mixture models

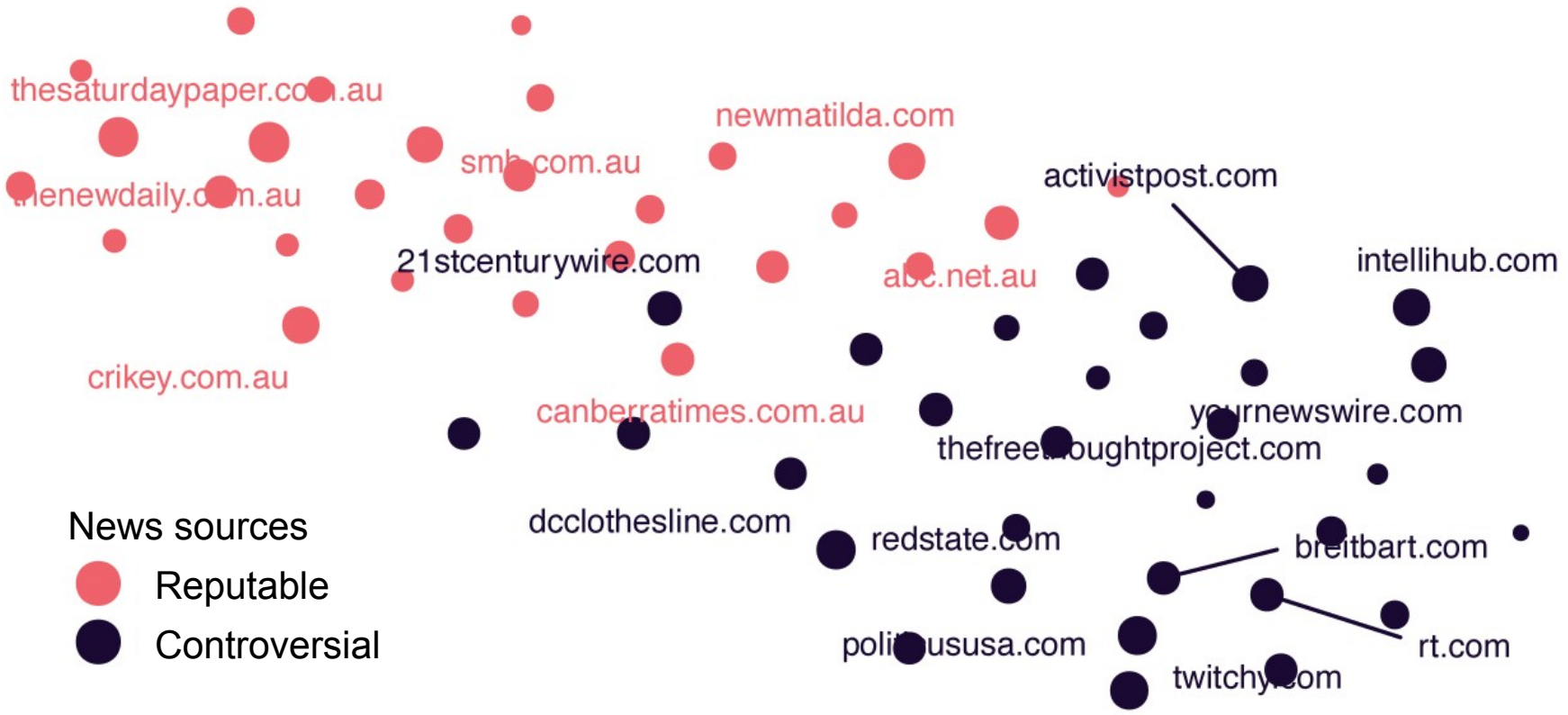
[Kong et al 2020]



Kong, Q., Rizoiu, M.-A., & Xie, L. (2020). Describing and Predicting Online Items with Reshare Cascades via Dual Mixture Self-exciting Processes. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 645–654). New York, NY, USA: ACM.

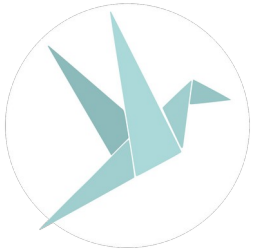
(1) Separating controversial from reputable

[Kong et al 2020]



Reputable and controversial sources are separable based solely on how their information spreads

Detect controversial news without content analysis



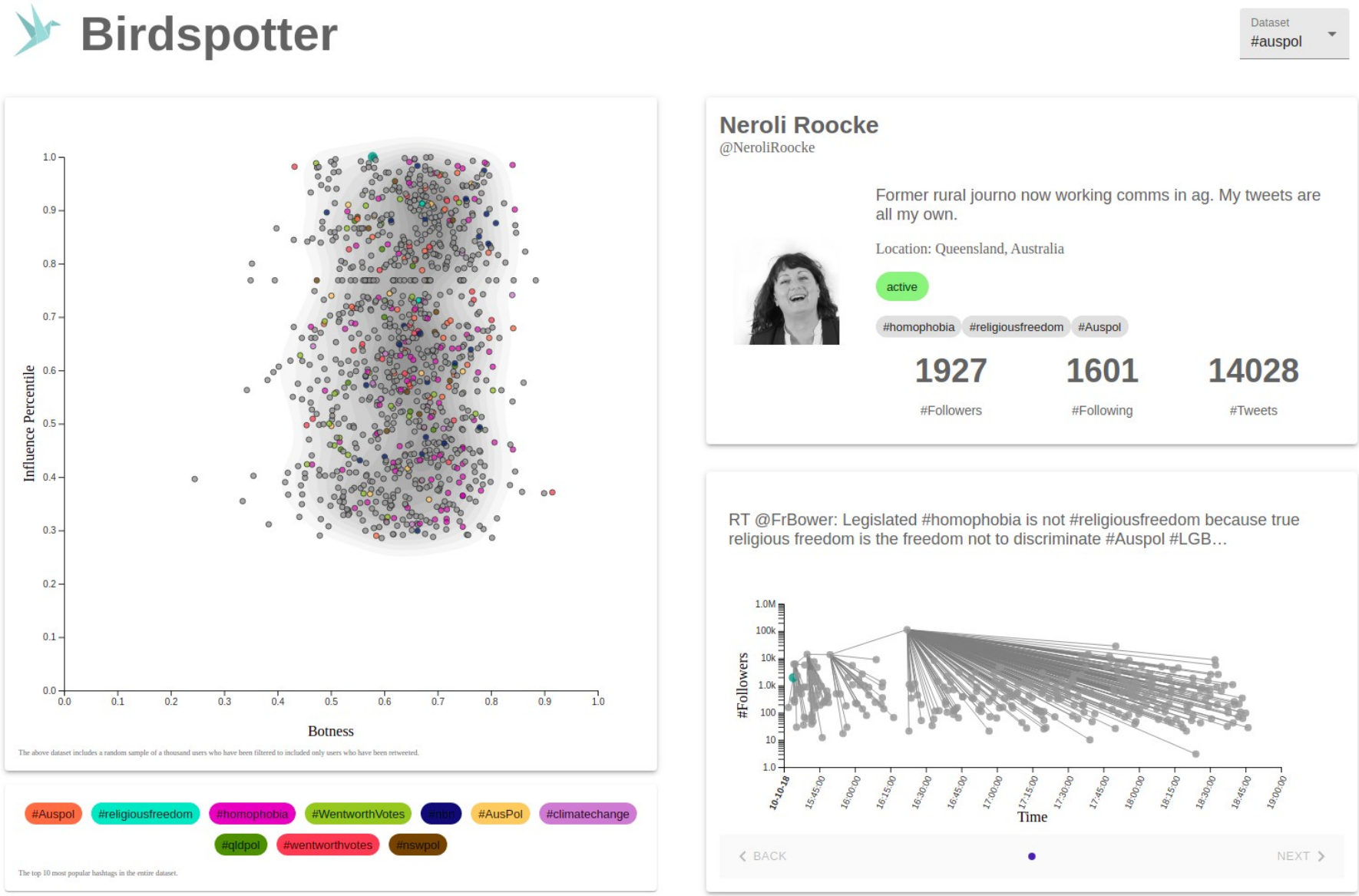
Maintainer: Rohit Ram
rohit.ram@uts.edu.au
https://github.com/behavioral-ds/BirdSpotter

birdspotter

- An **easy-to-use** Twitter user analysis tool
  - Designed for political scientists, sociologists and, data-practitioners
  - Models and labels the attributes of Twitter users
  - Prepackaged a state-of-the-art offline bot detector
  - Quantifies influence via a Tweet Dynamics System
- The **intuitive visualization birdspotter.ml** for dataset exploration and narrative construction.
- A **versatile Twitter user classifier,** trainable.

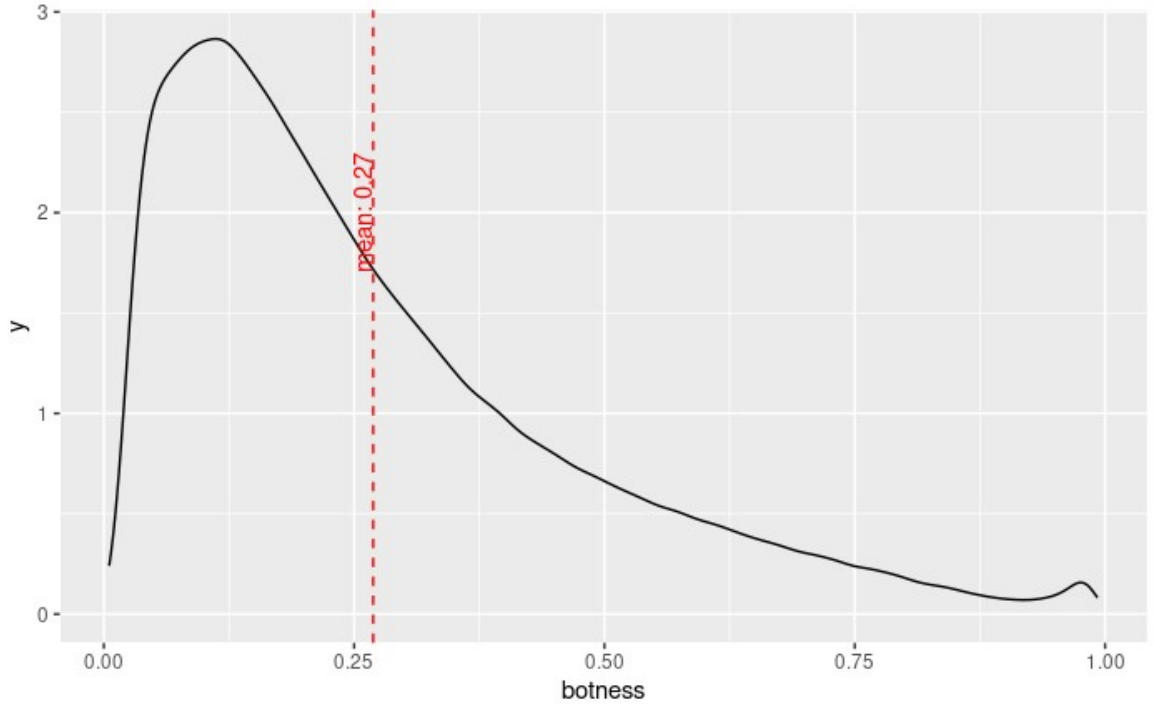
Ram, R., Kong, Q., & Rizoiu, M.-A. (2021). Birdspotter: A Tool for Analyzing and Labeling Twitter Users. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining (pp. 918–921). New York, NY, USA: ACM.

birdspotter.ml web interface

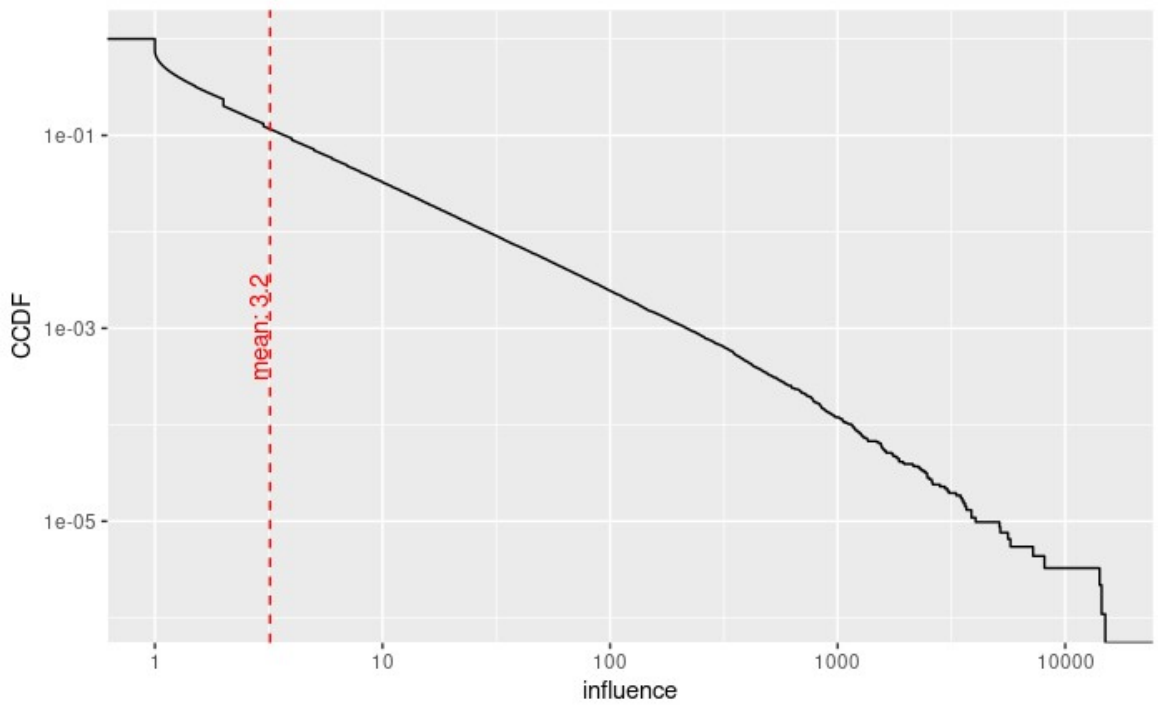


(2) Identify influential inauthentic users

- The 1-day COVID-19 dataset, collected by [Chen et al 2020].
- User online influence and botness are quantified using *birdspotter*

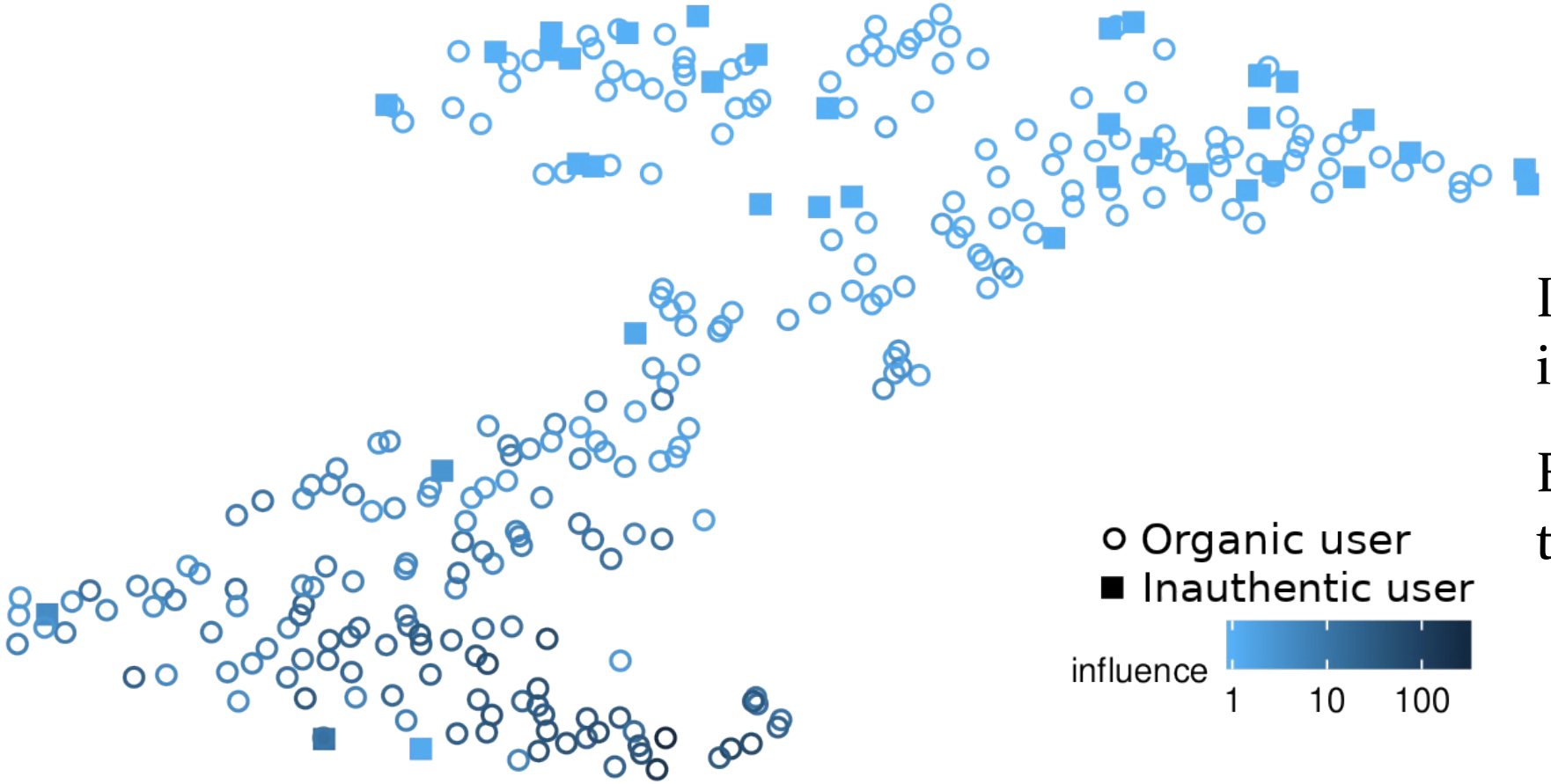


User botness
(likelihood to be a socialbot)



User influence
(average # retweets received by a tweet)

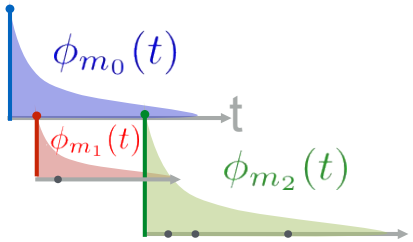
(2) Identify influential inauthentic users



Identify users engaged in influence operations

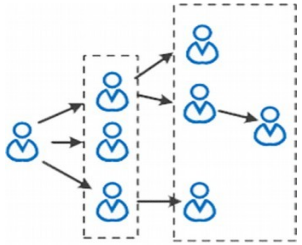
Estimate their impact on the wider community

Presentation plan

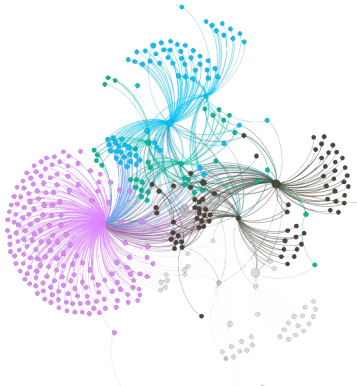


Gentle introduction to point processes, Poisson and Hawkes processes; how to design a kernel for social media.

A quick tutorial on how to leverage *evently* for the most basic Hawkes process operations.



How to analyze social media data using *evently;* how to fit parameters to real-world data, how to compute popularity metrics.



Beyond fitting and popularity: building user embeddings that allow applications such as detecting controversial news sources and socialbot, using solely the reaction of the social system.

Further reading and online resources

Gentle introduction to point processes for online social media:

Rizoiu, M.-A., Lee, Y., & Mishra, S. (2017). Hawkes processes for events in social media. In S.-F. Chang (Ed.), Frontiers of Multimedia Research (pp. 191–218). ACM. https://doi.org/10.1145/3122865.3122874

A more complete, yet accessible presentation for Computer Scientists:

Laub, P. J., Taimre, T., & Pollett, P. K. (2015). Hawkes Processes. Retrieved from http://arxiv.org/abs/1507.02822

A cryptic exhaustive presentation, the bible of point processes:

Daley, D. J., & Vere-Jones, D. (2008). An introduction to the theory of point processes. {V}ol. {I} (Vol. I). https://doi.org/10.1007/b97277

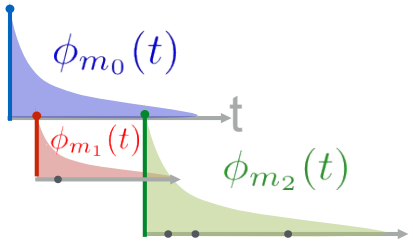
Analyzing online social media with evently and birdspotter tutorials:

https://github.com/behavioral-ds/user-analysis

https://github.com/behavioral-ds/BirdSpotter

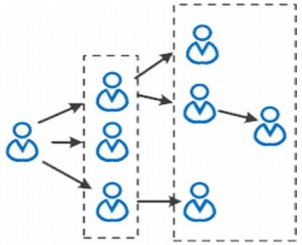
https://www.behavioral-ds.science/evently/

Thank you!

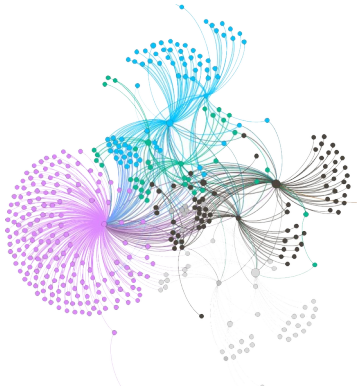


Gentle introduction to point processes, Poisson and Hawkes processes; how to design a kernel for social media.

A quick tutorial on how to leverage *evently* for the most basic Hawkes process operations.



How to analyze social media data using *evently;* how to fit parameters to real-world data, how to compute popularity metrics.



Beyond fitting and popularity: building user embeddings that allow applications such as detecting controversial news sources and socialbot, using solely the reaction of the social system.