Statistical Machine Learning

Point Processes
Temporal Point Processes

Spring 2020

(Partly From ICML Tutorial, July 2018)

Outline

Introduction to Point processes (PPs)

Temporal Point processes (TPPs)

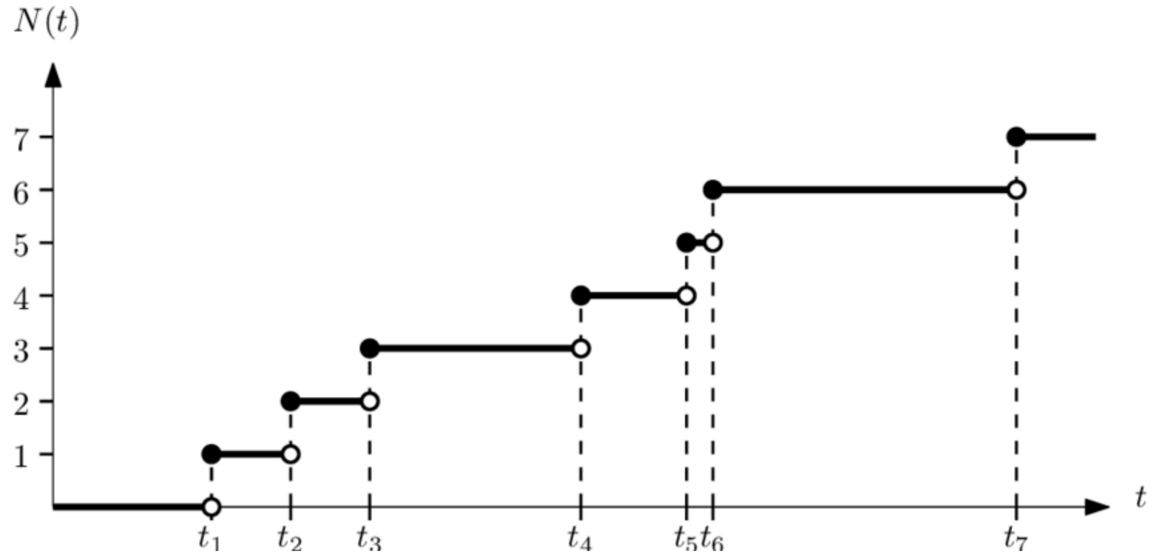
1. Intensity function
2. Basic building blocks
3. Superposition
4. Marks and SDEs with jumps

Models & Inference

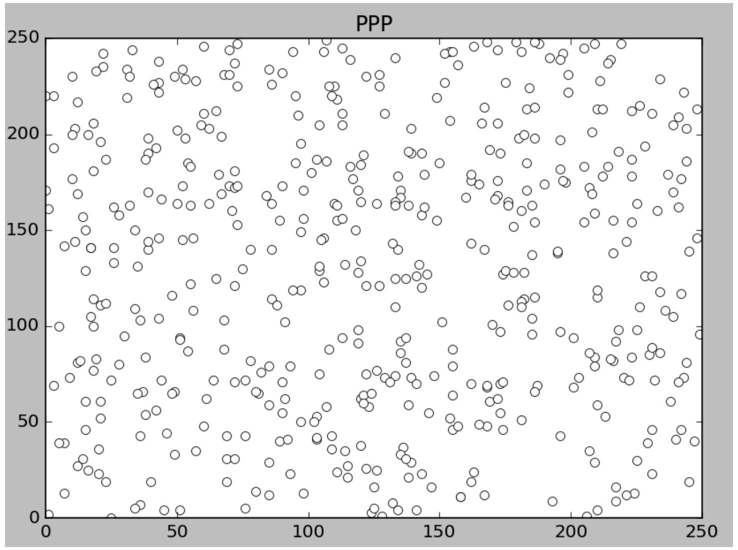
1. Modeling event sequences
2. Clustering event sequences
3. Capturing complex dynamics
4. Causal reasoning on event sequences

Introduction to Point Processes

- Point processes are used to describe data that are localized in space or time.
- A temporal point process is a stochastic, or random, process composed of a time-series of binary events that occur in continuous time.



An example point process realization {t 1 , t 2 ,. .. } and corresponding counting process N (t).



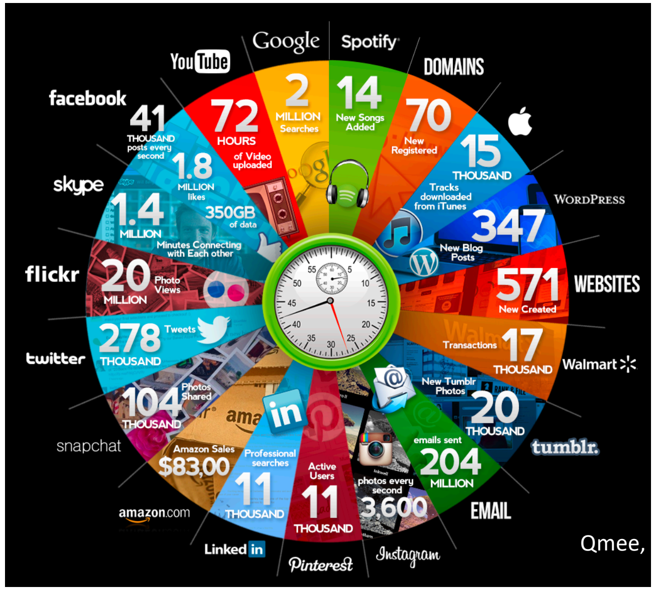
Poisson point process

Introduction to Point Processes

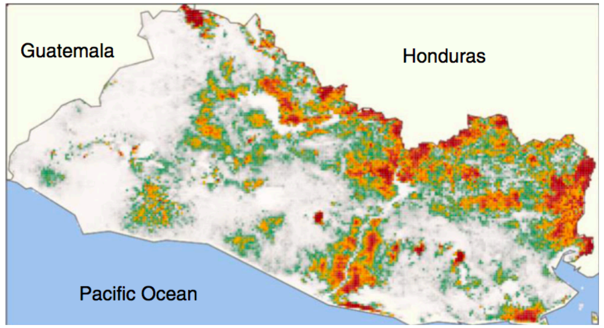
- Point processes are used to describe data that are localized at a finite set of time points. As opposed to continuous-valued processes.
- A point process can take on only one of two possible values, indicating whether or not an event occurs at that time.
- Point processes have many applications in real world data.
- For example: the study of point processes is especially crucial for neural data analysis.
- Brain areas receive, process, and transmit information about the outside world via stereotyped electrical events, called action potentials or spikes.
- Spikes are the starting point for virtually all of the processing performed by the brain. This can be modeled by point processes.

Introduction to Point Processes

More examples: many discrete events in continuous time

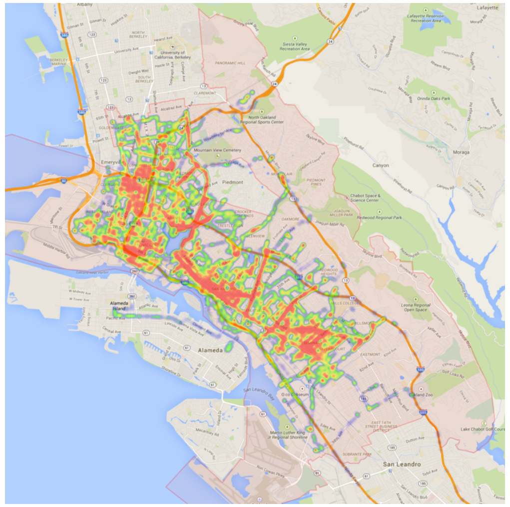


Online actions



Disease dynamics

Financial trading



Mobility dynamics

Introduction to Point Processes

Variety of processes behind these events

Events are (noisy) observations of a variety of complex dynamic processes…

Stock trading

Flu spreading

Article creation in Wikipedia

News spread in Twitter

Reviews and sales in Amazon

Ride-sharing requests

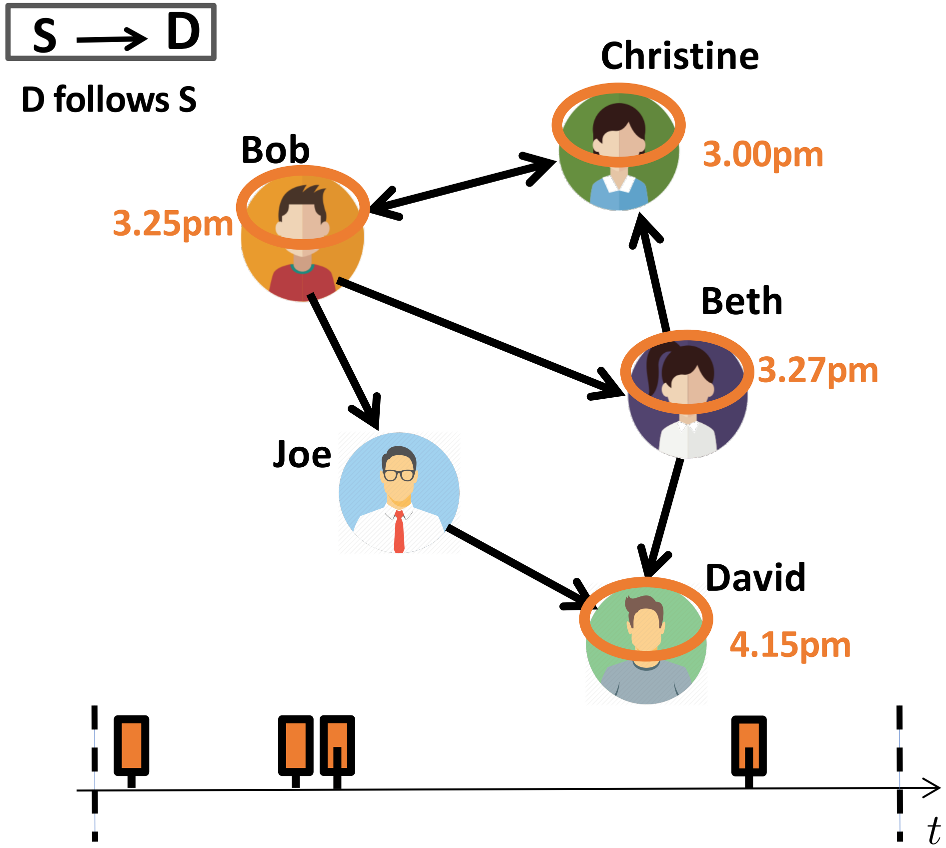
A user's reputation in Quora

FAST

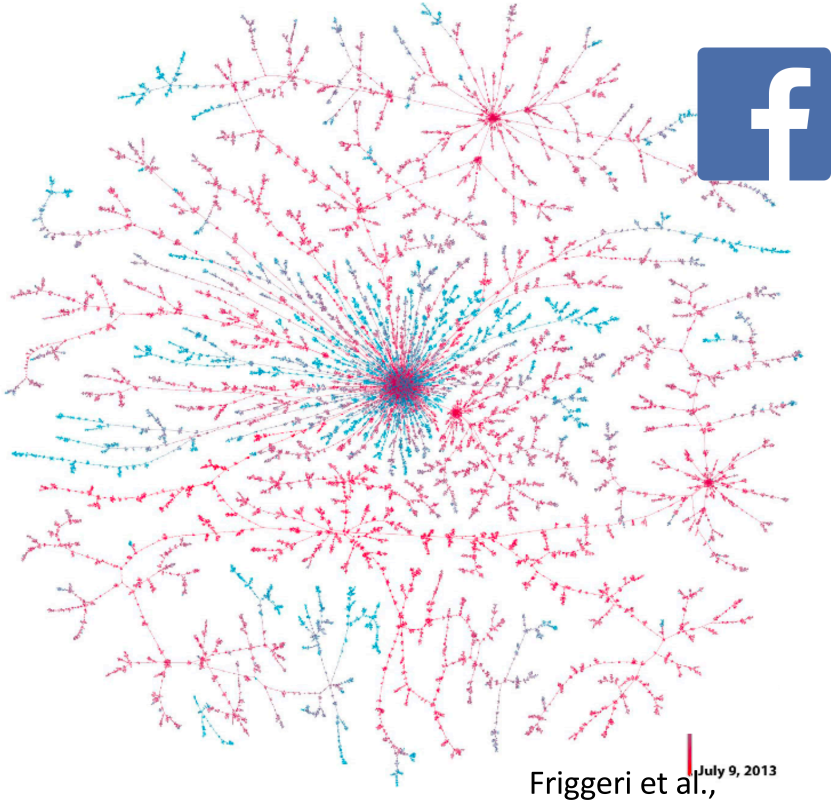
SLOW

…in a wide range of temporal scales.

Point Processes and Information propagation



They can have an impact in the off-line world

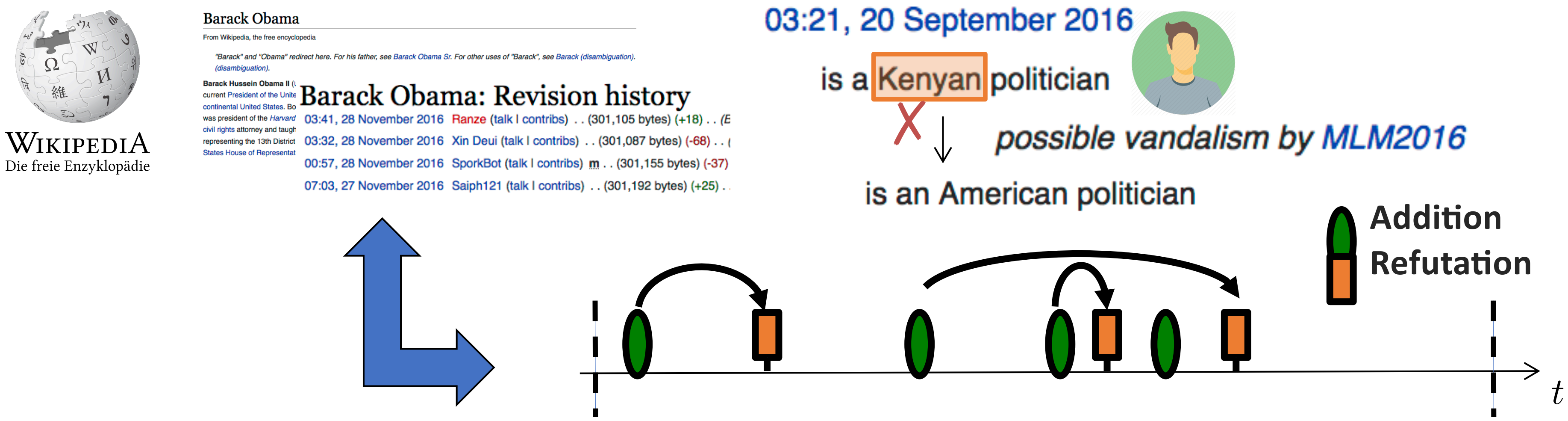


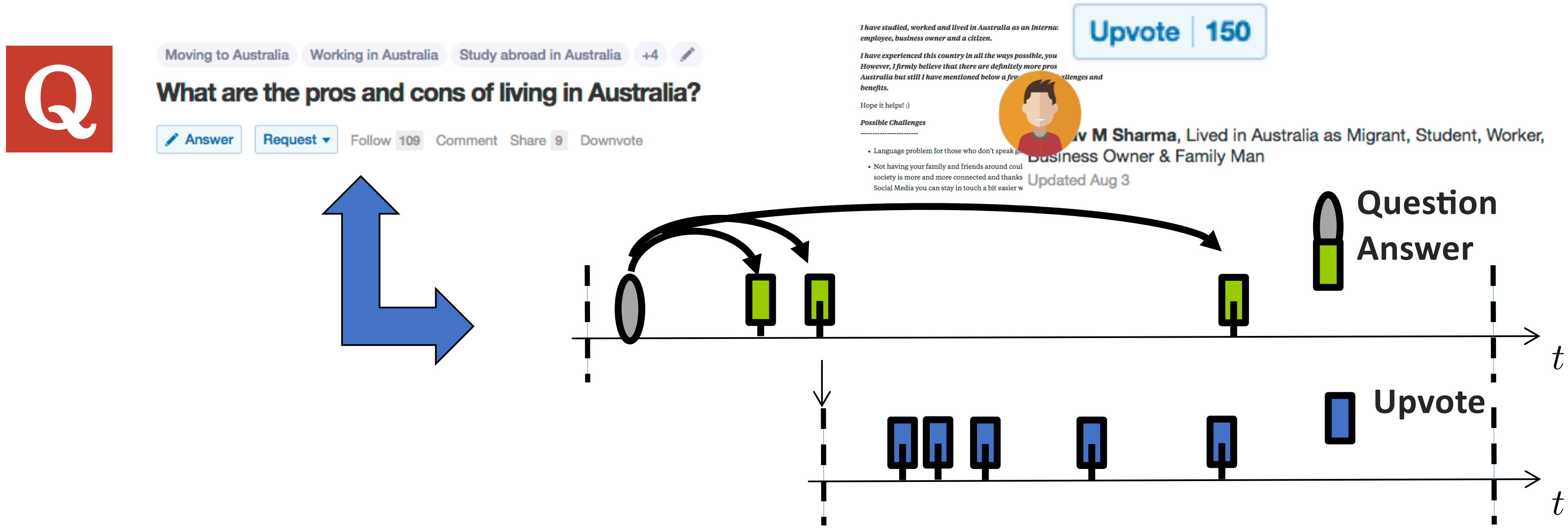
Friggeri et al., 2014

theguardian

Click and elect: how fake news helped Donald Trump win a real election

Point Processes and Information propagation

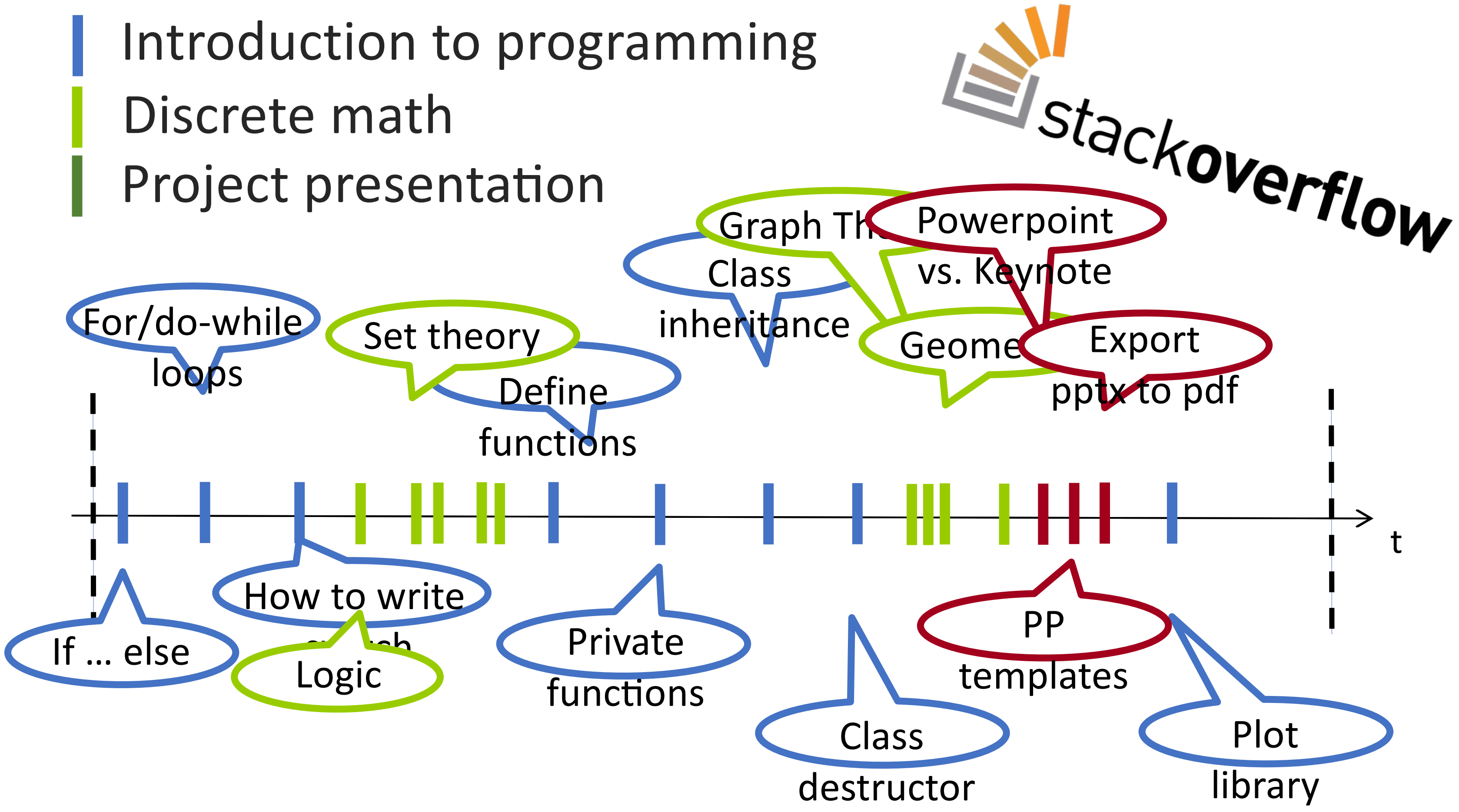




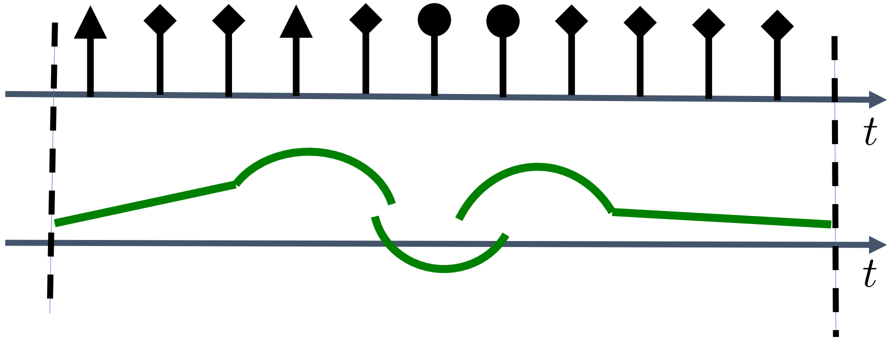
Point Processes and Information propagation

1st year computer science student

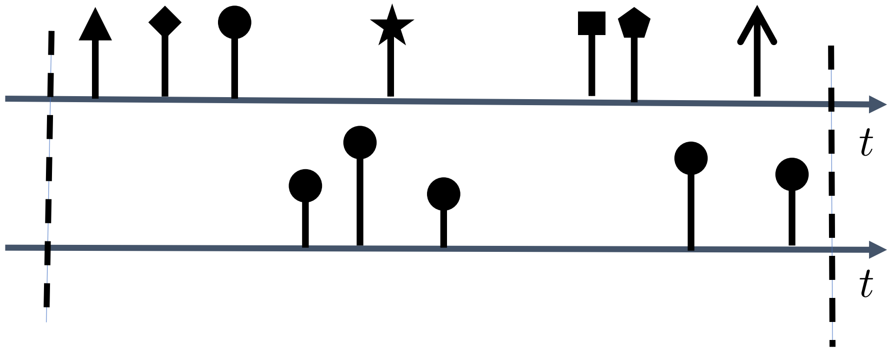
- Introduction to programming
- Discrete math
- Project presentation



Aren't these event traces just time series?

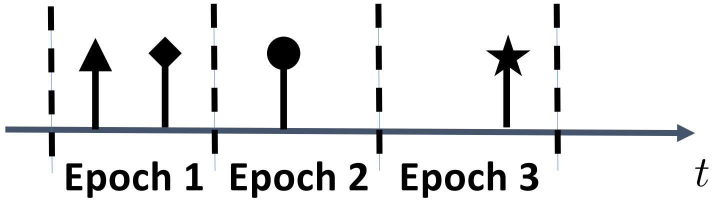


Discrete and continuous times series



Discrete events in continuous time

What about aggregating events in *epochs*?



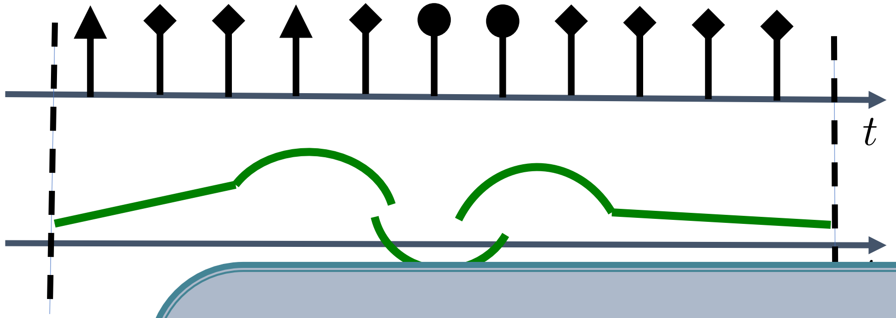
How long is each epoch?

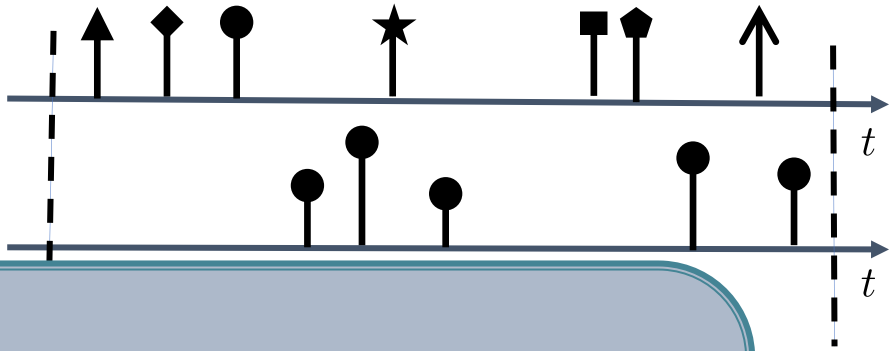
How to aggregate events per epoch?

What if no event in one epoch?

What about time-related queries?

Aren’t these event traces just time series?





Di

The framework of
temporal point processes
provides a ***native representation***

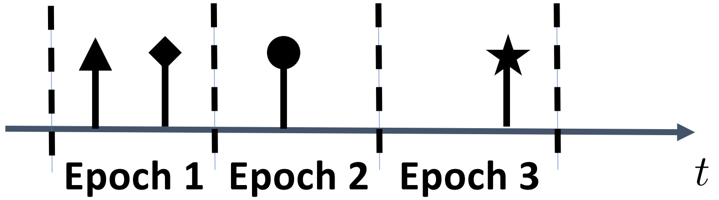
W

events in epochs?

poch?

What if no event in one epoch?

What about time-related queries?



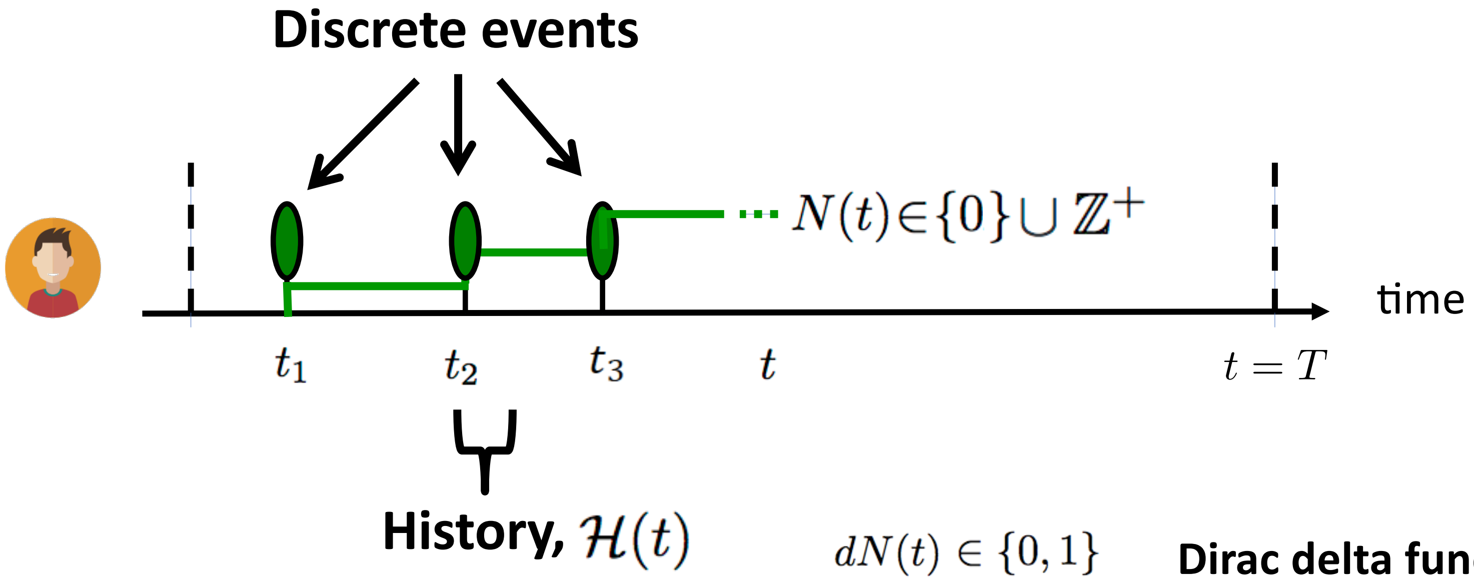
Temporal Point Processes (TPPs):

1. **Intensity function**
2. Basic building blocks
3. Superposition
4. Marks and SDEs with jumps

Temporal point processes

Temporal point process:

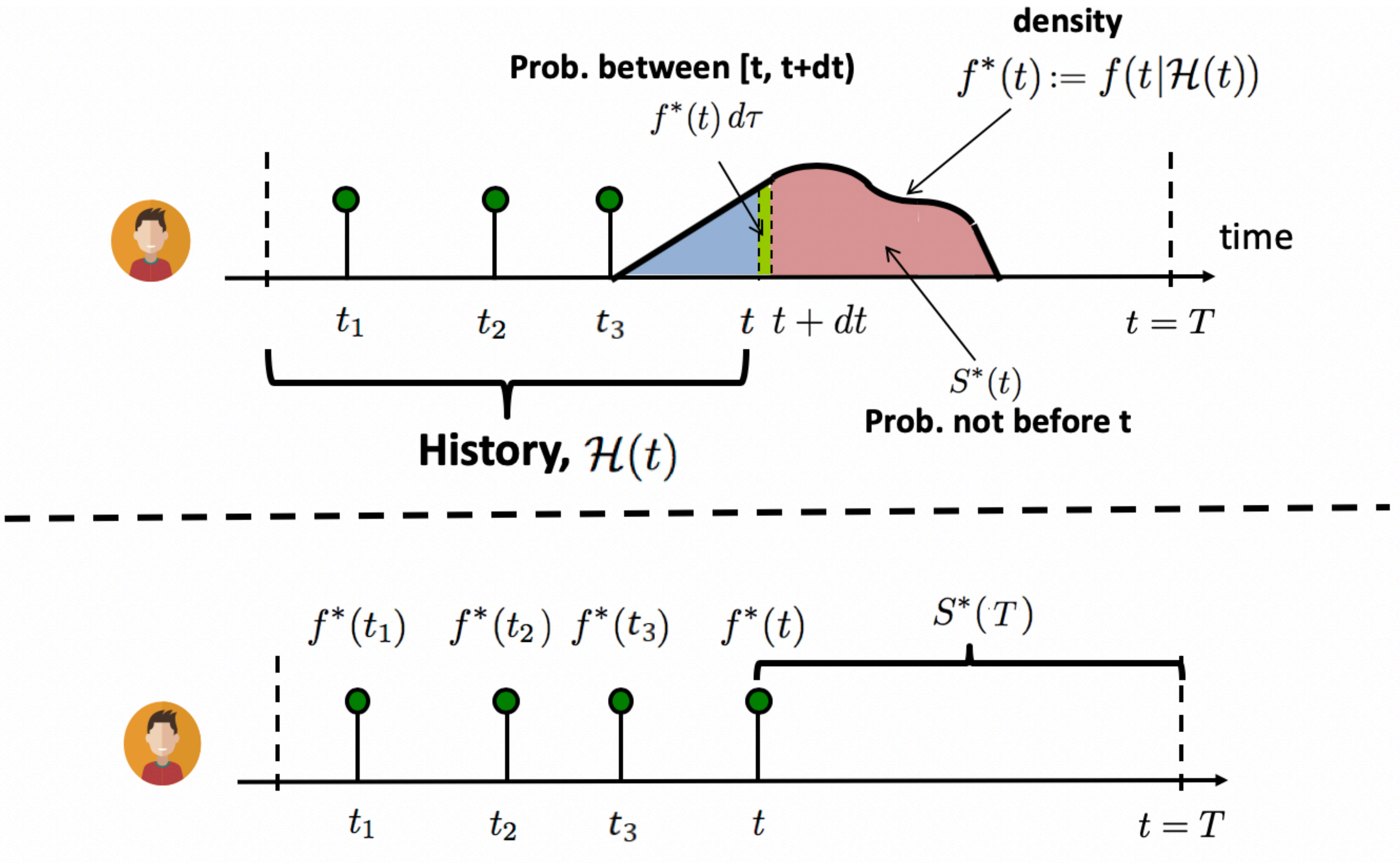
A random process whose realization consists of **discrete events localized in time** $\mathcal{H} = \{t_i\}$



$dN(t) \in \{0, 1\}$ **Dirac delta function**

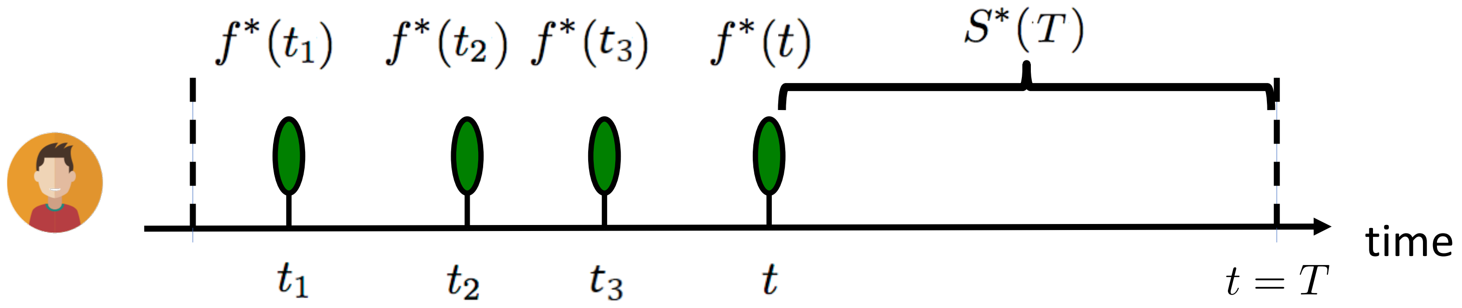
Formally: $N(t) = \int_0^t dN(s)$ ⇒ $dN(t) = \sum_{t_i \in \mathcal{H}} \delta(t - t_i)\, dt$

Model time as a random variable



Likelihood of a timeline: $f^*(t_1)\, f^*(t_2)\, f^*(t_3)\, f^*(t)\; S^*(T)$

Problems of density parametrization (I)



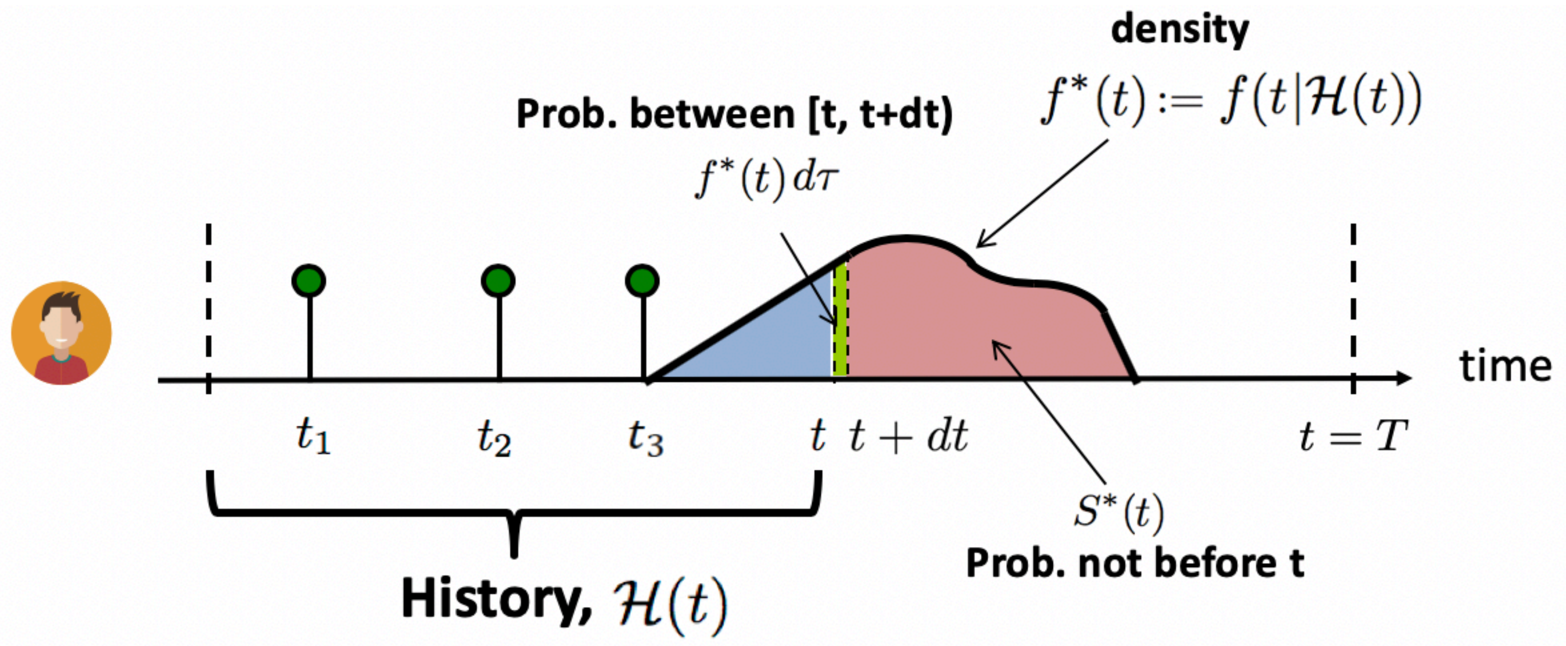
$$f^*(t_1) \quad f^*(t_2) \quad f^*(t_3) \quad f^*(t) \quad S^*(T)$$

$$\frac{\exp\langle w, \psi^*(t_1)\rangle}{Z} \quad \frac{\exp\langle w, \psi^*(t_2)\rangle}{Z} \quad \frac{\exp\langle w, \psi^*(t_3)\rangle}{Z} \quad \frac{\exp\langle w, \psi^*(t)\rangle}{Z} \quad 1 - \int_t^T \frac{\exp\langle w, \psi^*(\tau)\rangle}{Z} d\tau$$

It is difficult for model design and interpretability:

1. **Densities need to integrate to 1 (i.e., partition function)**
2. **Difficult to combine timelines**

Intensity function

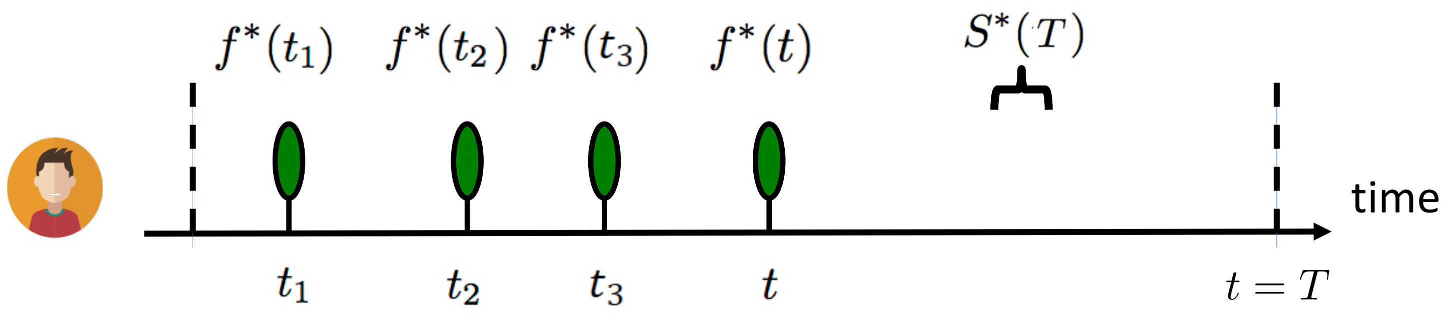


Intensity:
Probability between [t, t+dt) but not before t

$$\lambda^*(t)dt = \frac{f^*(t)dt}{S^*(t)} \geq 0 \quad \Rightarrow \quad \lambda^*(t)dt = \mathbb{E}[dN(t)|\mathcal{H}(t)]$$

Observation: $\lambda^*(t)$ **It is a rate = # of events / unit of time**

Advantages of intensity parametrization (I)



$$\lambda^*(t_1)\,\lambda^*(t_2)\,\lambda^*(t_3)\,\lambda^*(t)\;\exp\left(-\int_0^T \lambda^*(\tau)\,d\tau\right)$$

$\langle w, \phi^*(t_1)\rangle$

$\langle w, \phi^*(t_2)\rangle$

$\langle w, \phi^*(t_3)\rangle$

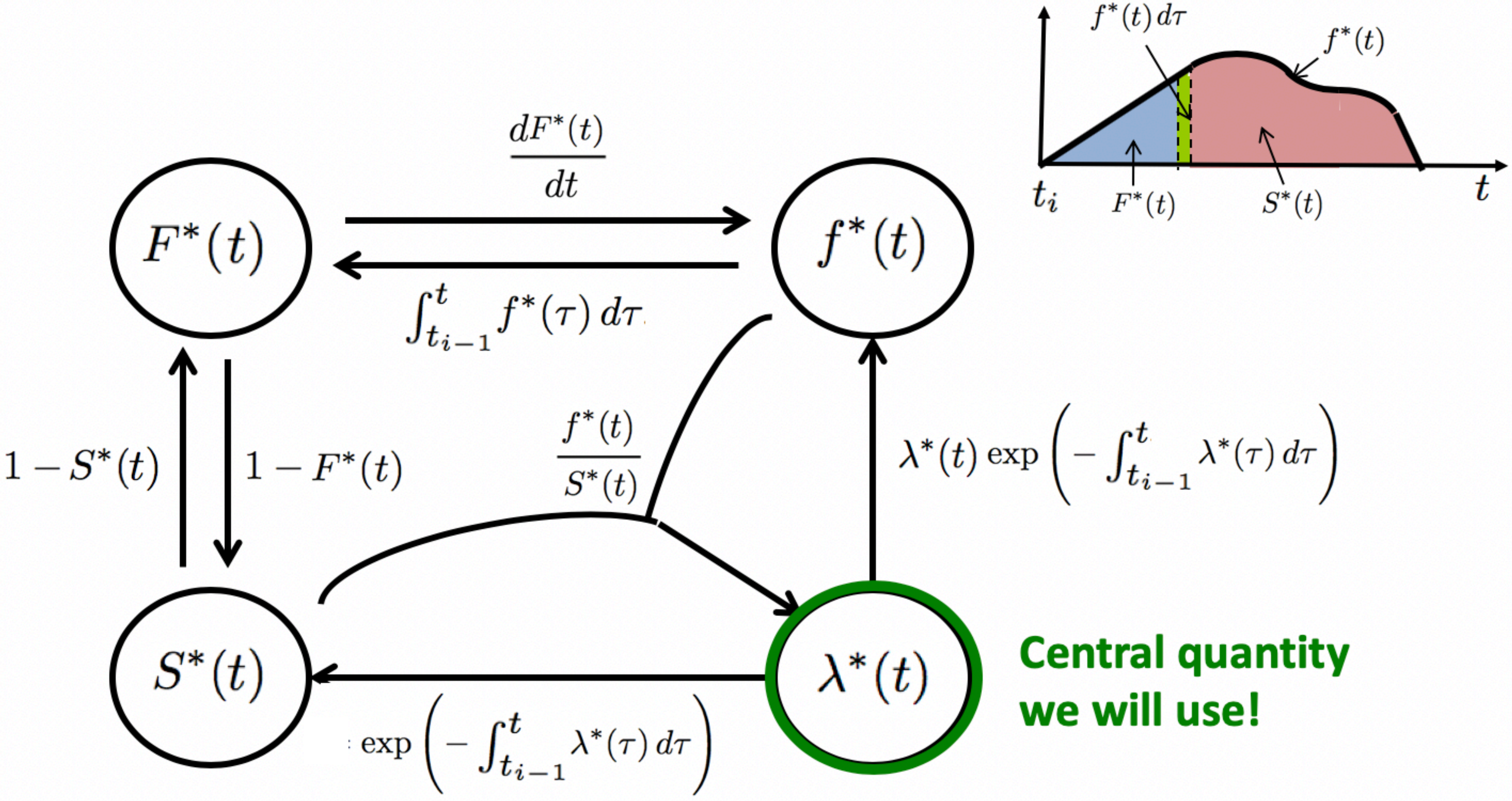
$\langle w, \phi^*(t)\rangle$

$\exp\left(-\int_0^T \langle w, \phi^*(\tau)\rangle d\tau\right)$

Suitable for model design and interpretable:

1. **Intensities only need to be nonnegative**
2. **Easy to combine timelines**

Relation between f*, F*, S*, λ*

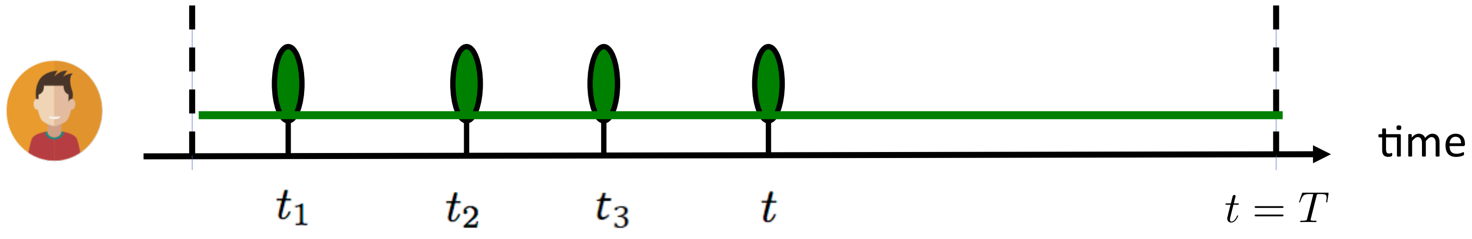


Central quantity we will use!

Representation:
Temporal Point Processes

1. Intensity function
2. Basic building blocks
3. Superposition
4. Marks and SDEs with jumps

Poisson process



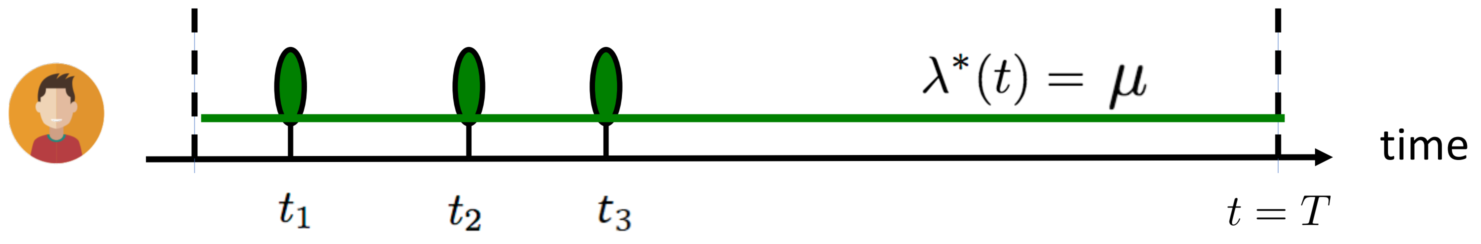
Intensity of a Poisson process

$$\lambda^*(t) = \mu$$

Observations:

1. **Intensity independent of history**
2. **Uniformly random occurrence**
3. **Time interval follows exponential distribution**

Fitting & sampling from a Poisson



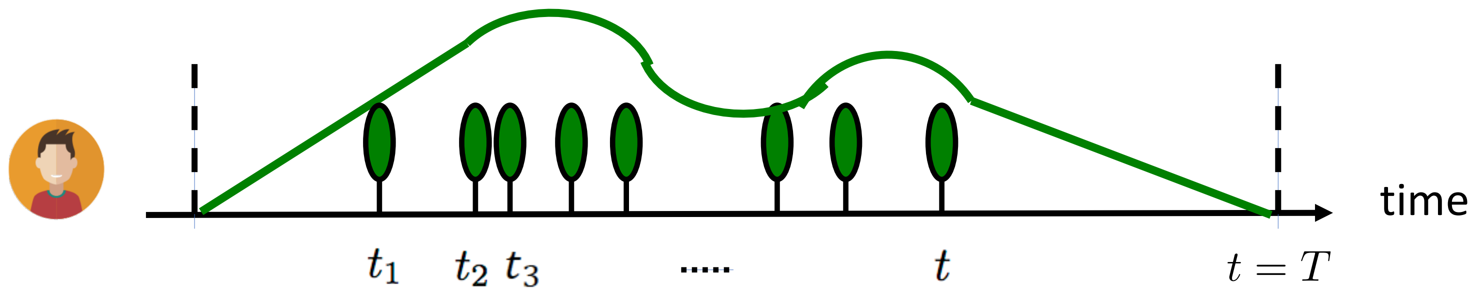
Fitting by maximum likelihood:

$$\mu^* = \underset{\mu}{\operatorname{argmax}} \; 3 \log \mu - \mu T = \frac{3}{T}$$

Sampling using inversion sampling:

$$t \sim \underbrace{\mu \exp\left(-\mu(t - t_3)\right)}_{f_t^*(t)} \quad \Rightarrow \quad t = \underbrace{-\frac{1}{\mu} \log(1 - \overset{\overset{Uniform(0,1)}{\downarrow}}{u}) + t_3}_{F_t^{-1}(u)}$$

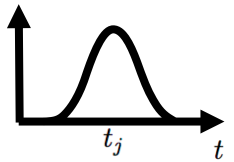
Inhomogeneous Poisson process



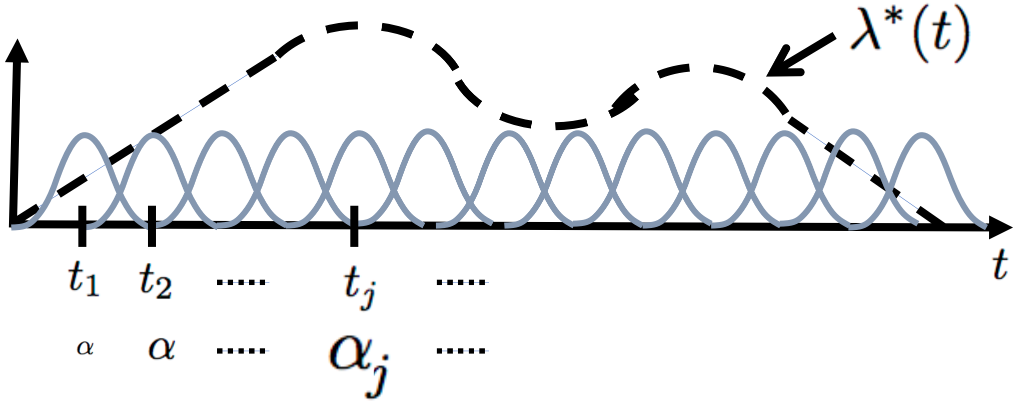
Intensity of an inhomogeneous Poisson process

$$\lambda^*(t) = g(t) \geqslant 0$$ **(Independent of history)**

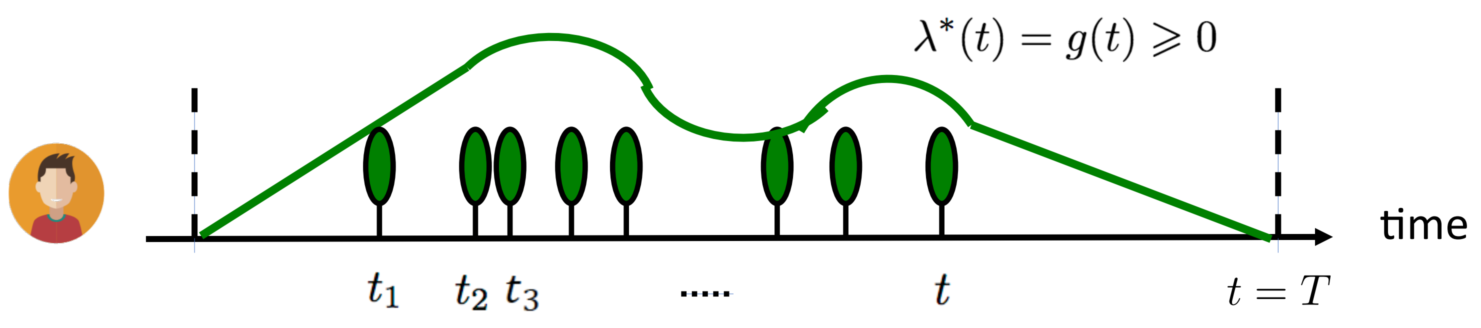
Example:



$$\lambda^*(t) = \sum_j \alpha_j \, k(t - t_j)$$



Fitting & sampling from inhomogeneous Poisson



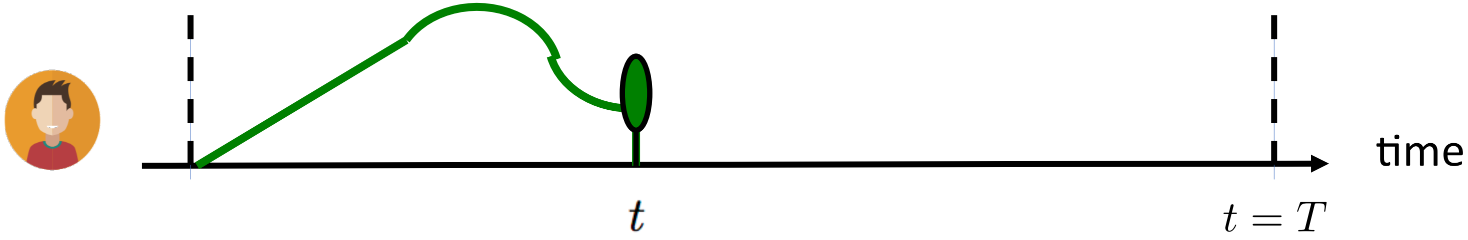
Fitting by maximum likelihood: $\underset{g(t)}{\text{maximize}} \sum_{i=1}^{n} \log g(t_i) - \int_0^T g(\tau)\, d\tau$

Sampling using thinning (reject. sampling) + inverse sampling:

1. **Sample** t **from Poisson process with intensity** μ **using inverse sampling**
2. **Generate** $u_2 \sim Uniform(0, 1)$
3. **Keep the sample if** $u_2 \leq g(t) / \mu$

} **Keep sample with prob.** $g(t)/\mu$

Terminating (or survival) process



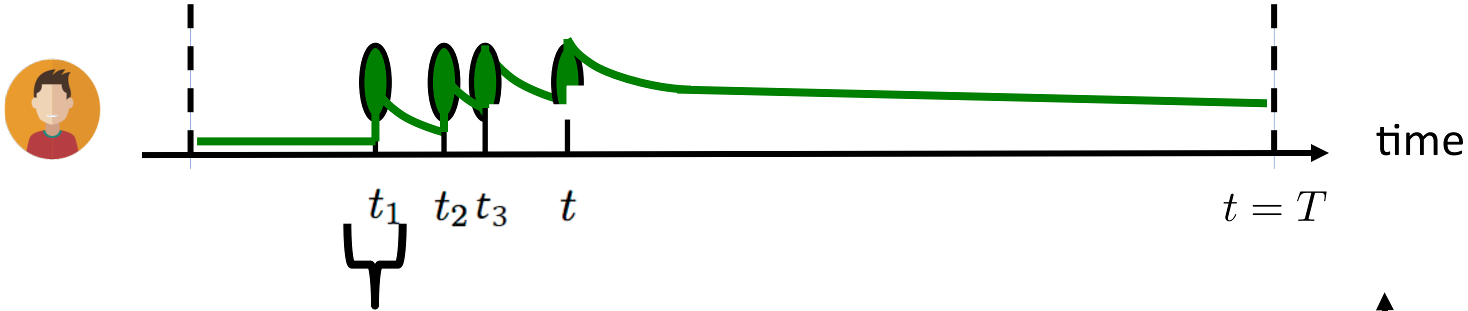
Intensity of a terminating (or survival) process

$$\lambda^*(t) = g^*(t)(1 - N(t)) \geqslant 0$$

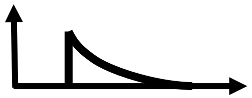
Observations:

1. **Limited number of occurrences**

Self-exciting (or Hawkes) process



History, $\mathcal{H}(t)$



Intensity of self-exciting (or Hawkes) process:

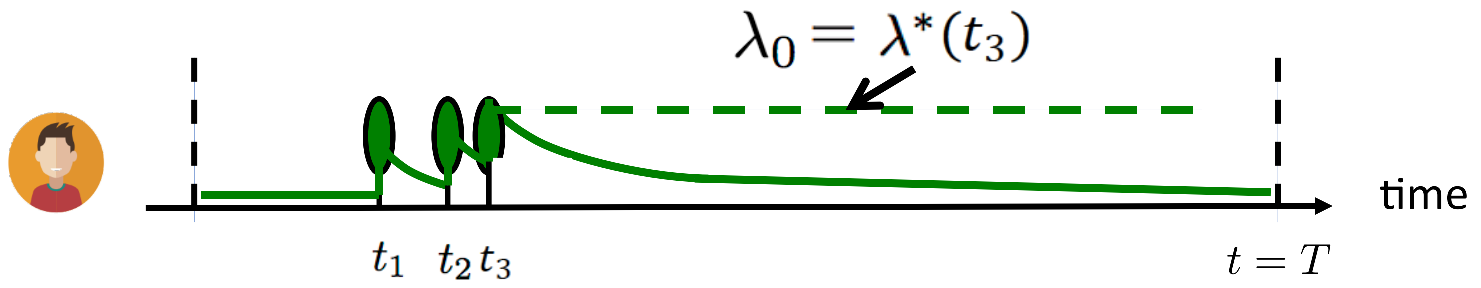
$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \overbrace{\kappa_\omega(t - t_i)}^{\text{Triggering kernel}}$$

$$= \mu + \alpha \kappa_\omega(t) \star dN(t)$$

Observations:

1. **Clustered (or bursty) occurrence of events**
2. **Intensity is stochastic and history dependent**

Fitting a Hawkes process from a recorded timeline



Fitting by maximum likelihood:

$$\underset{\mu,\alpha}{\text{maximize}} \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau$$

The max. likelihood is jointly convex in μ and α

Sampling using thinning (reject. sampling) + inverse sampling:

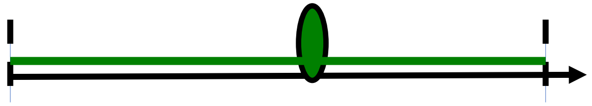
Key idea: the maximum of the intensity λ_0 changes over time

Summary

Building blocks to represent **different dynamic processes:**

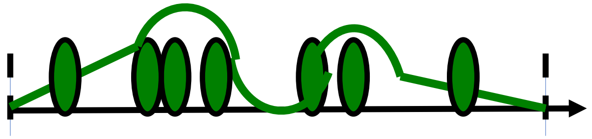
Poisson processes:

$$\lambda^*(t) = \lambda$$



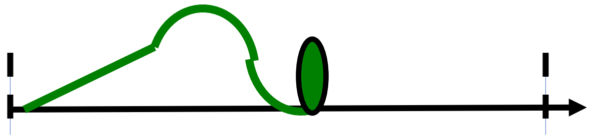
Inhomogeneous Poisson processes:

$$\lambda^*(t) = g(t)$$



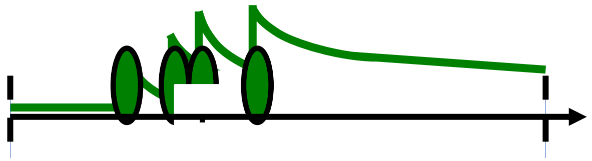
Terminating point processes:

$$\lambda^*(t) = g^*(t)(1 - N(t))$$



Self-exciting point processes:

$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i)$$

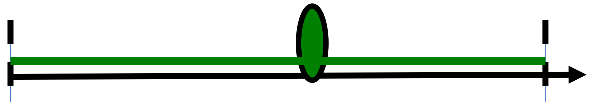


Summary

Building blocks to represent **different dynamic processes:**

Poisson processes:

$$\lambda^*(t) = \lambda$$



Inho

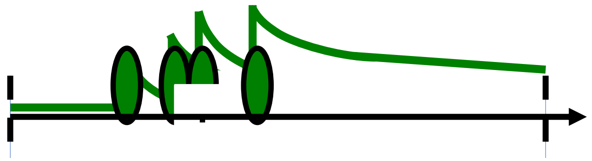
We know **how to fit** them
and **how to sample** from them

Term

$$\lambda^*(t) = g^*(t)(1 - N(t))$$

Self-exciting point processes:

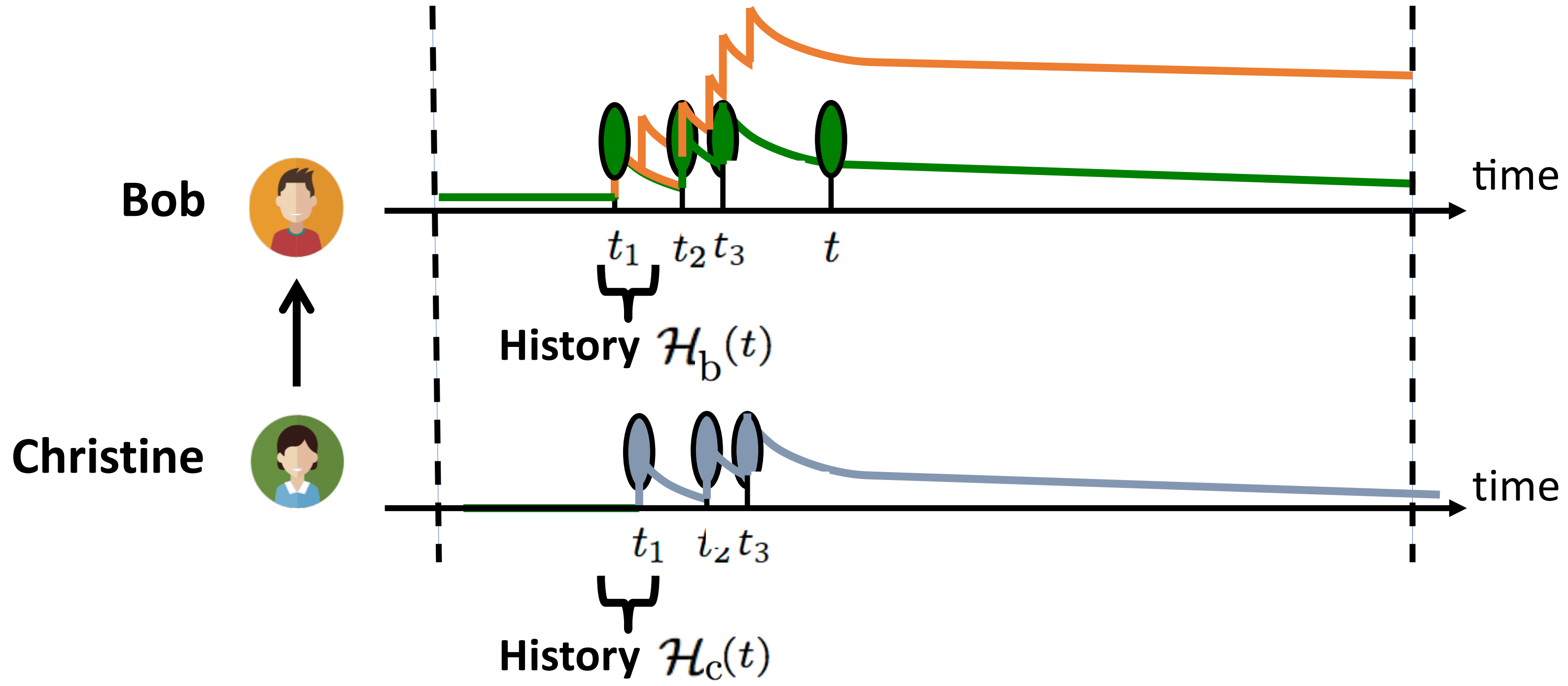
$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i)$$



Representation:
Temporal Point Processes

1. Intensity function
2. Basic building blocks
3. **Superposition**
4. Marks and SDEs with jumps

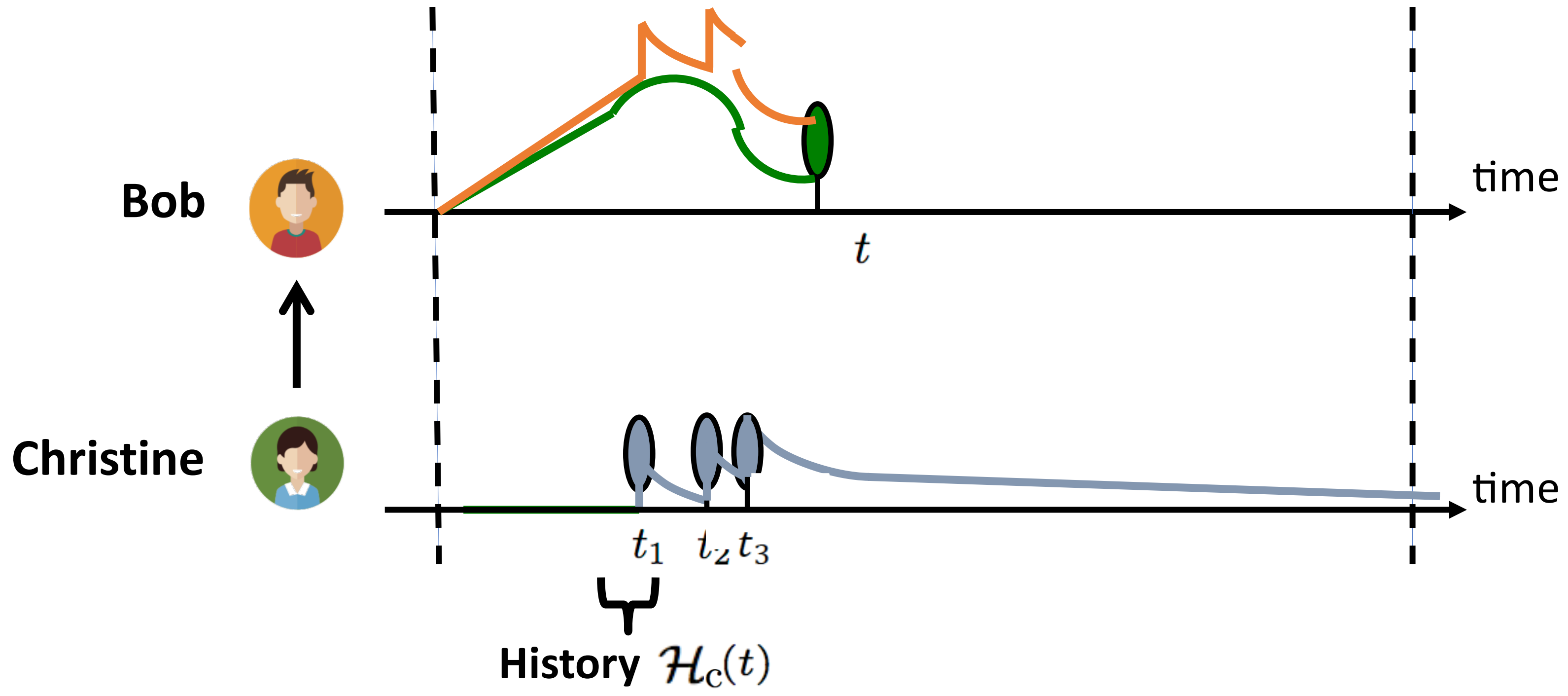
Mutually exciting process



Clustered occurrence affected by neighbors

$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}_b(t)} \kappa_\omega(t - t_i) + \beta \sum_{t_i \in \mathcal{H}_c(t)} \kappa_\omega(t - t_i)$$

Mutually exciting terminating process



Clustered occurrence affected by neighbors

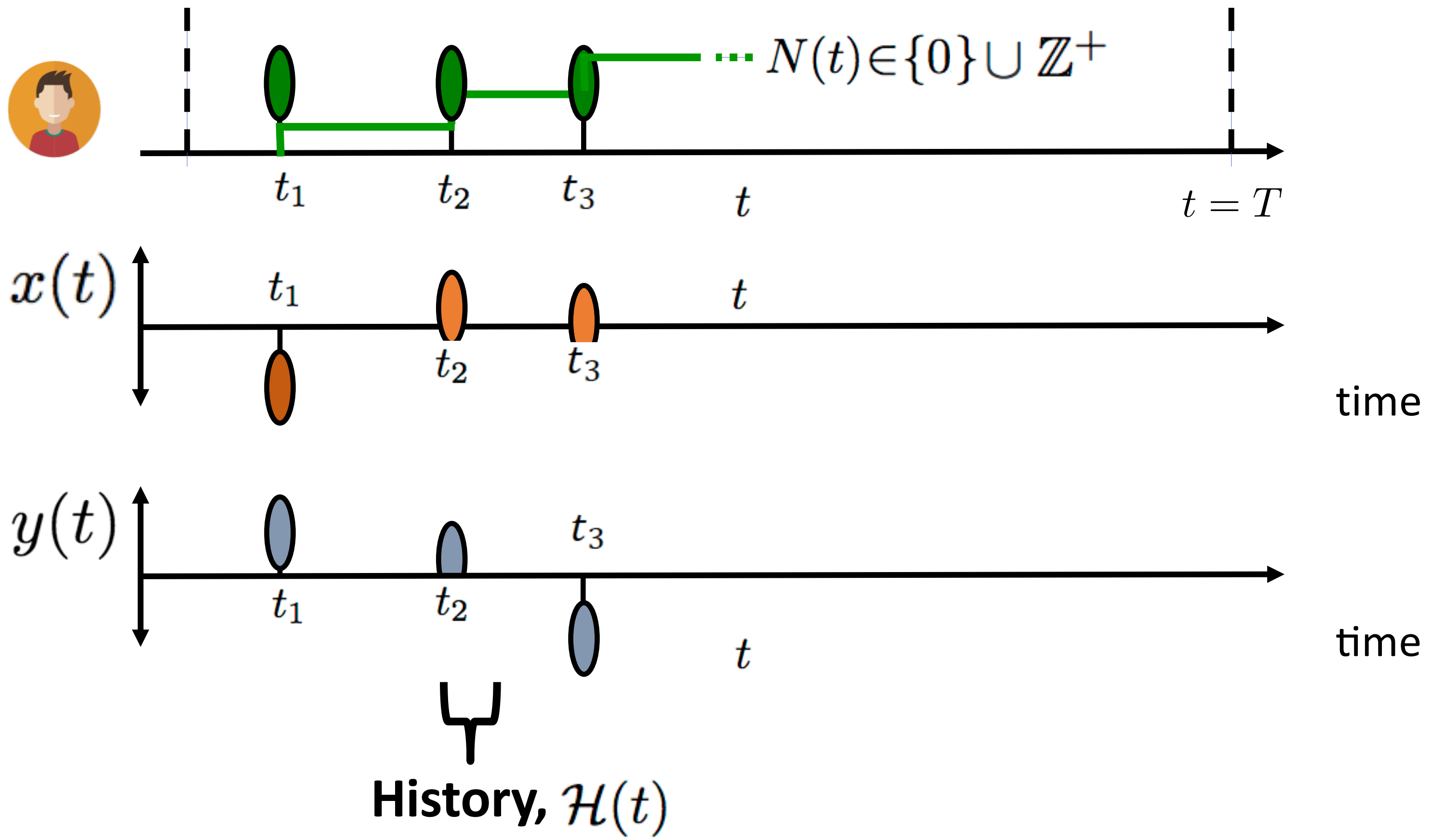
$$\lambda^*(t) = (1 - N(t))\Big(g(t) + \beta \sum_{t_i \in \mathcal{H}_c(t)} \kappa_\omega(t - t_i)\Big)$$

Representation:
Temporal Point Processes

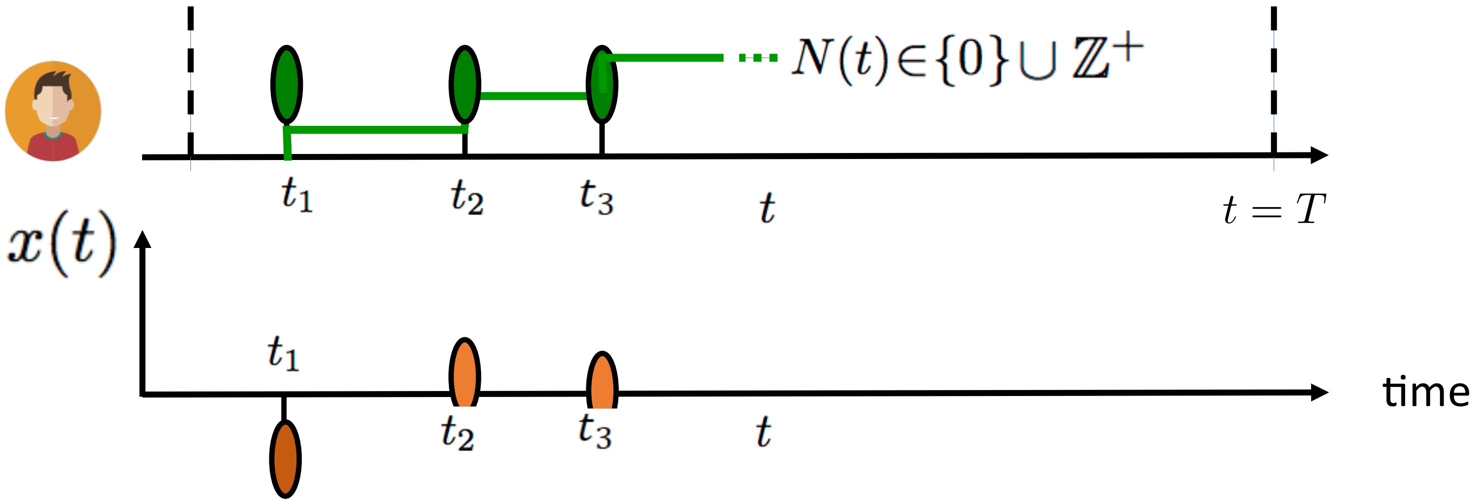
1. Intensity function
2. Basic building blocks
3. Superposition
4. **Marks and SDEs with jumps**

Marked temporal point processes

Marked temporal point process:
A random process whose realization consists of **discrete *marked* events localized in time**



Independent identically distributed marks



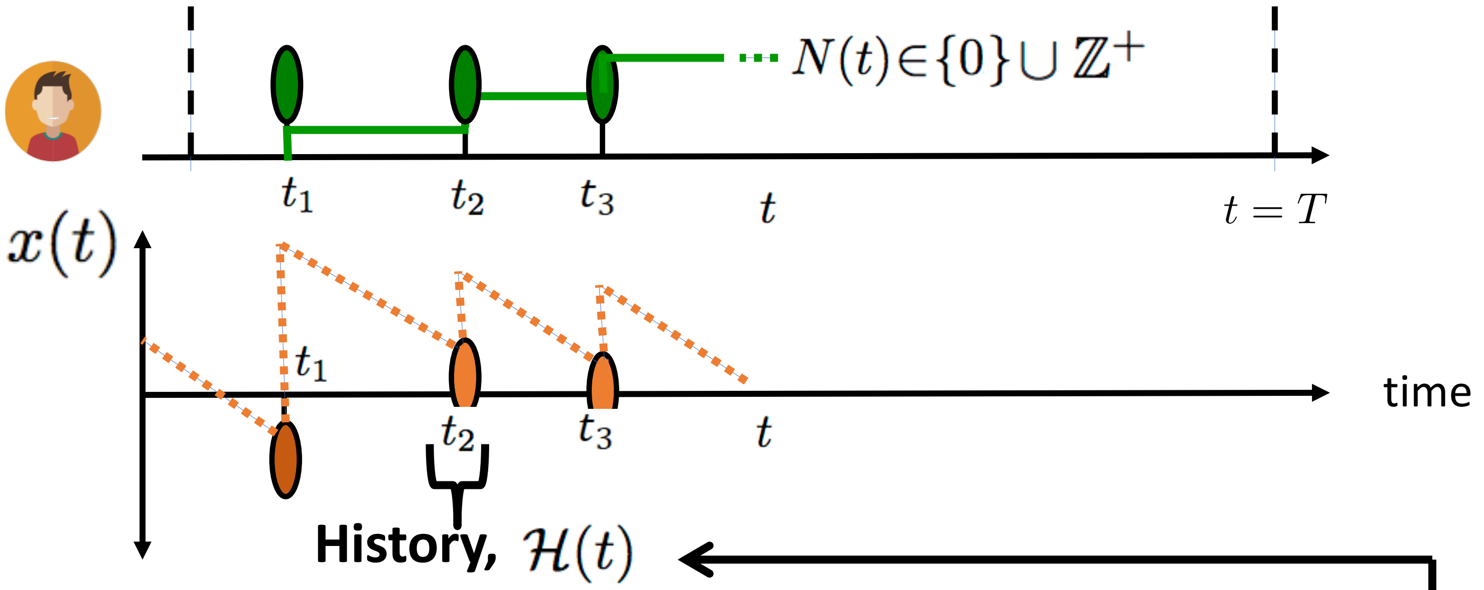
Distribution for the marks:

$$x^*(t_i) \sim p(x)$$

Observations:

1. **Marks independent of the temporal dynamics**
2. **Independent identically distributed (I.I.D.)**

Dependent marks: SDEs with jumps



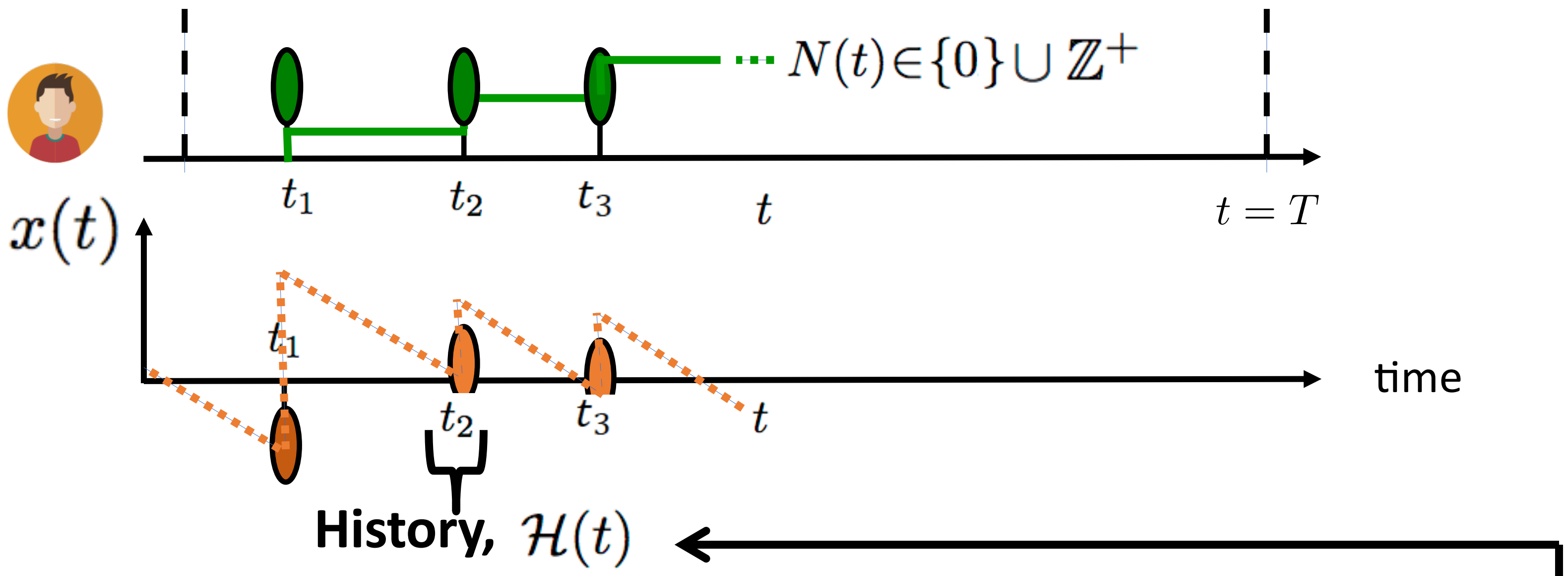
Marks given by stochastic differential equation with jumps:

$$x(t+dt) - x(t) = dx(t) = f(\underbrace{x(t), t}_{\textbf{Drift}})dt + h(\underbrace{x(t), t}_{\textbf{Event influence}})dN(t)$$

Observations:

1. **Marks dependent of the temporal dynamics**
2. **Defined for all values of t**

Dependent marks: distribution + SDE with jumps



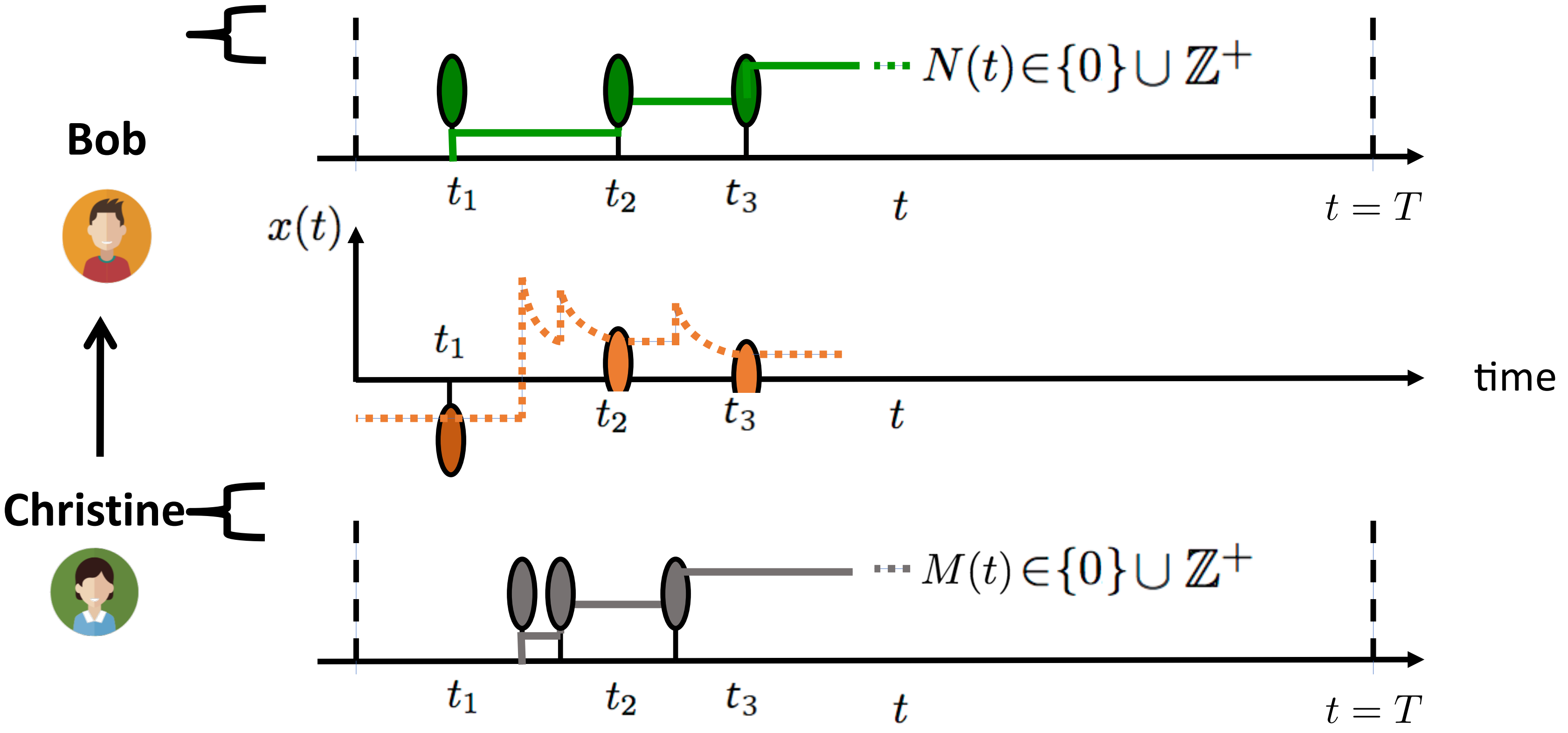
Distribution for the marks:

$$x^*(t_i) \sim p(x^* \mid x(t)) \Rightarrow dx(t) = \underbrace{f(x(t), t)}_{\text{Drift}}dt + \underbrace{h(x(t), t)}_{\text{Event influence}}dN(t)$$

Observations:

1. **Marks dependent on the temporal dynamics**
2. **Distribution represents additional source of uncertainty**

Mutually exciting + marks



Marks affected by neighbors

$$dx(t) = f(\underbrace{x(t), t}_{\text{Drift}})dt + g(\underbrace{x(t), t}_{\text{Neighbor influence}})dM(t)$$

Marked TPPs as stochastic dynamical systems

Example: Susceptible-Infected-Susceptible (SIS)

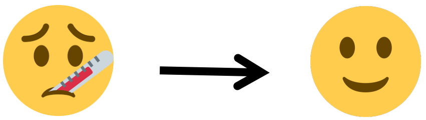
$X_i(t) = 0$ **Susceptible** $X_i(t) = 1$ **Infected** $X_i(t) = 0$ **Susceptible**

$$dX_i(t) = \underbrace{\overbrace{dY_i(t)}^{\text{SDE with jumps}}}_{\text{It gets infected}} - \underbrace{dW_i(t)}_{\text{It recovers}}$$

Infection rate

$$\mathbb{E}\left[dY_i(t)\right] = \lambda_{Y_i}(t)dt$$

$$\lambda_{Y_i}(t)dt = (1 - \overbrace{X_i(t)}^{\text{Node is susceptible}})\beta \underbrace{\sum_{j \in \mathcal{N}(i)}}_{\text{If friends are infected, higher infection rate}} X_j(t)dt$$



Recovery rate

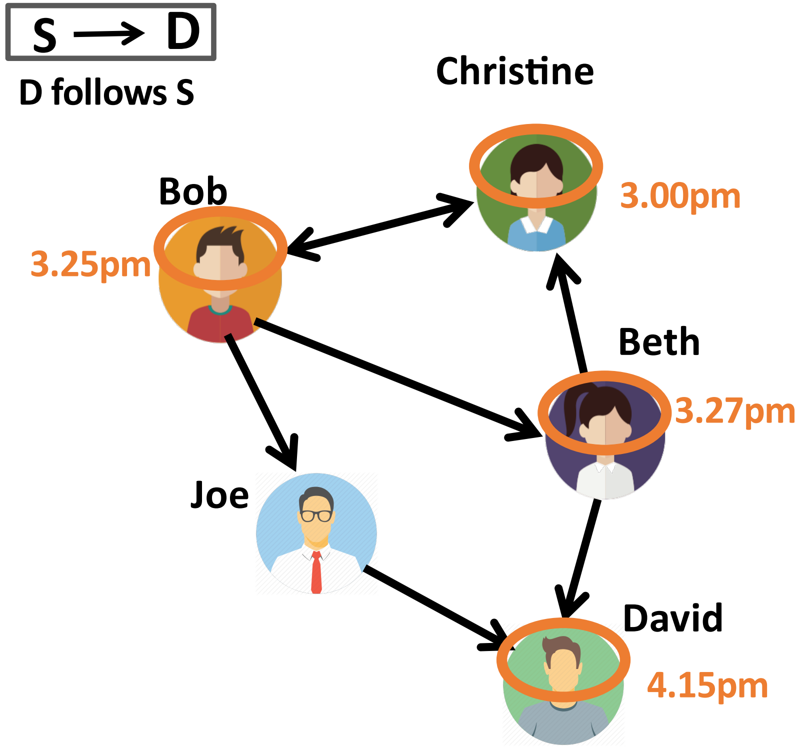
$$\mathbb{E}\left[dW_i(t)\right] = \lambda_{W_i}(t)dt$$

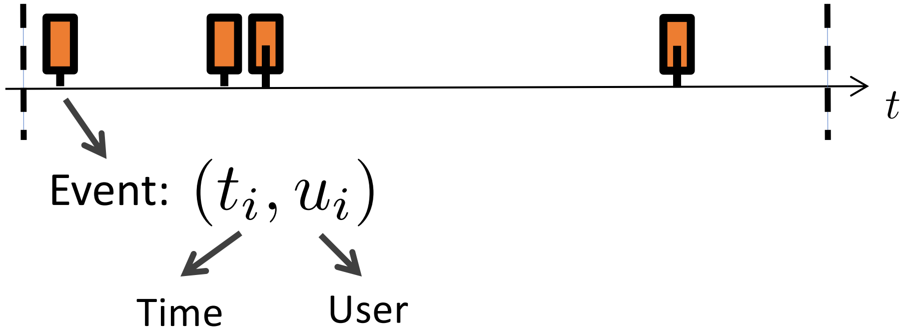
$$d\lambda_{W_i}(t) = \underbrace{\delta\, dY_i(t)}_{\text{Self-recovery rate when node gets infected}} - \overbrace{\lambda_{W_i}}^{\text{SDE with jumps}}\underbrace{(t)dW_i(t)}_{\text{If node recovers, rate to zero}} + \underbrace{\rho\, dN_i(t)}_{\text{Rate increases if node gets treated}}$$

Models & Inference

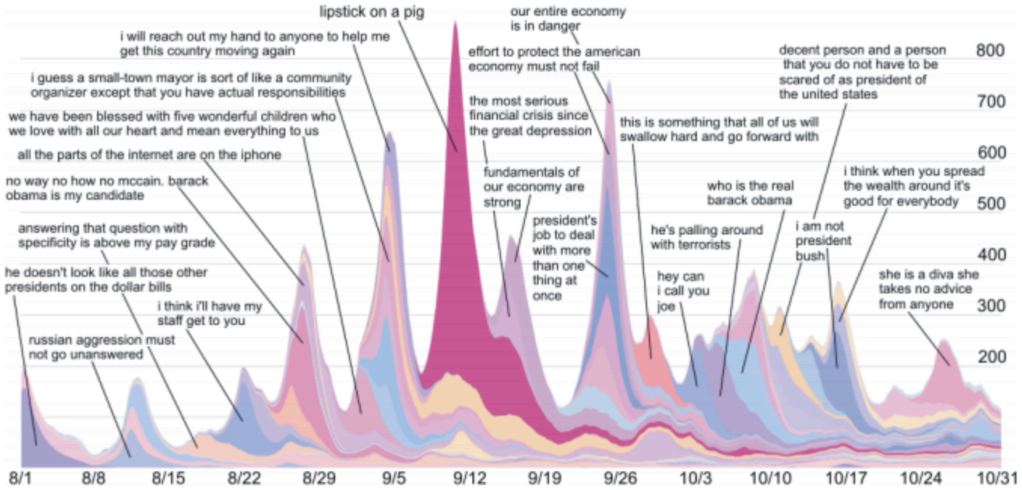
1. **Modeling event sequences**
2. Clustering event sequences
3. Capturing complex dynamics
4. Causal reasoning on event sequences

Event sequences as cascades



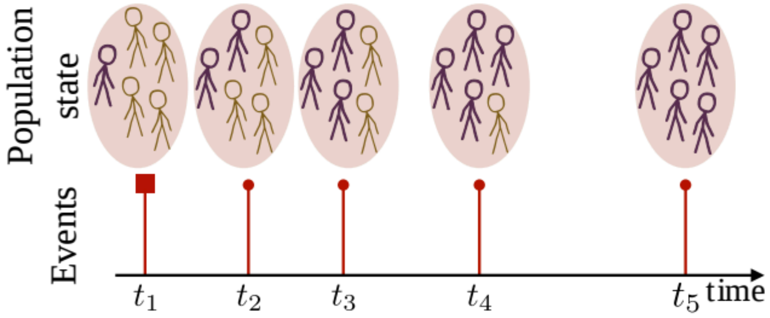


Information Diffusion



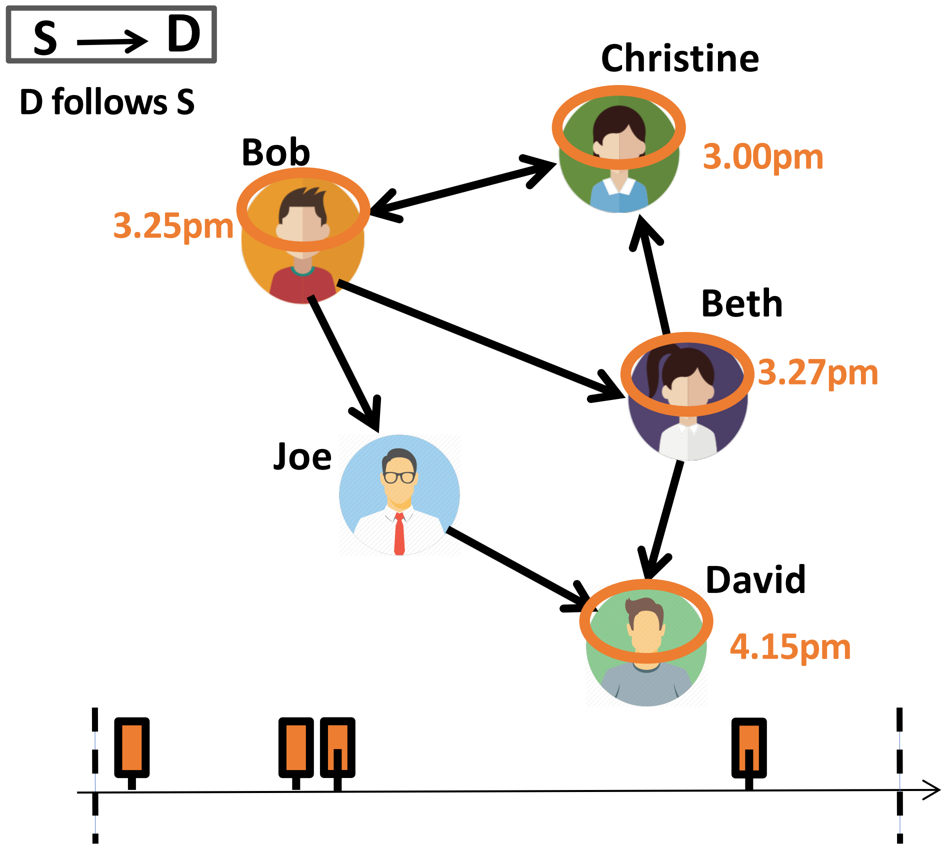
[Leskovec et al., 2009]

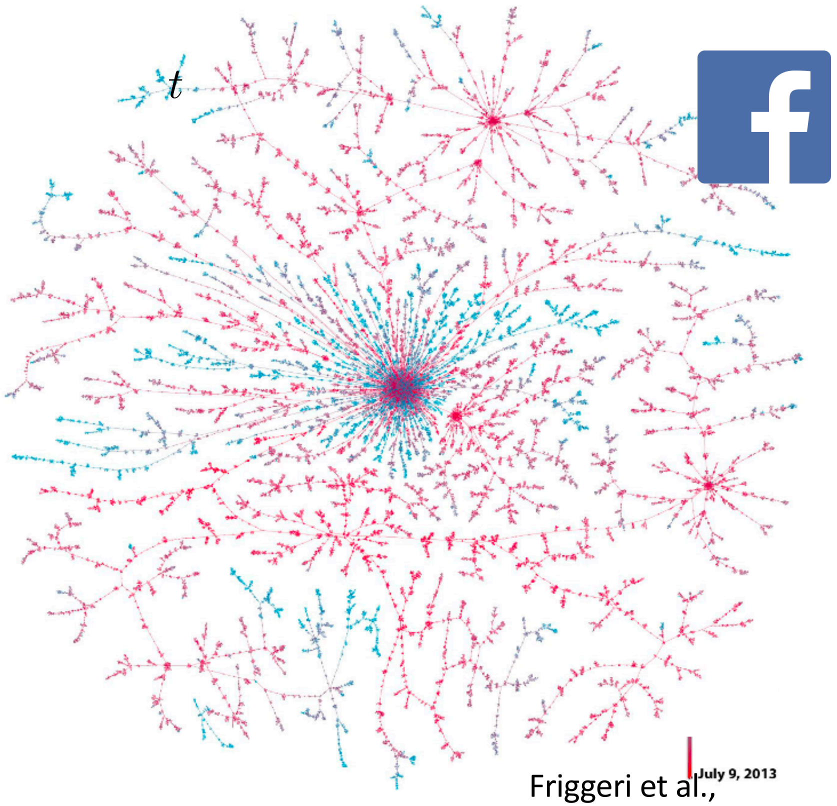
Disease Diffusion



[Rizoiu et al., 2018]

An example: idea adoption





Friggeri et al., 2014

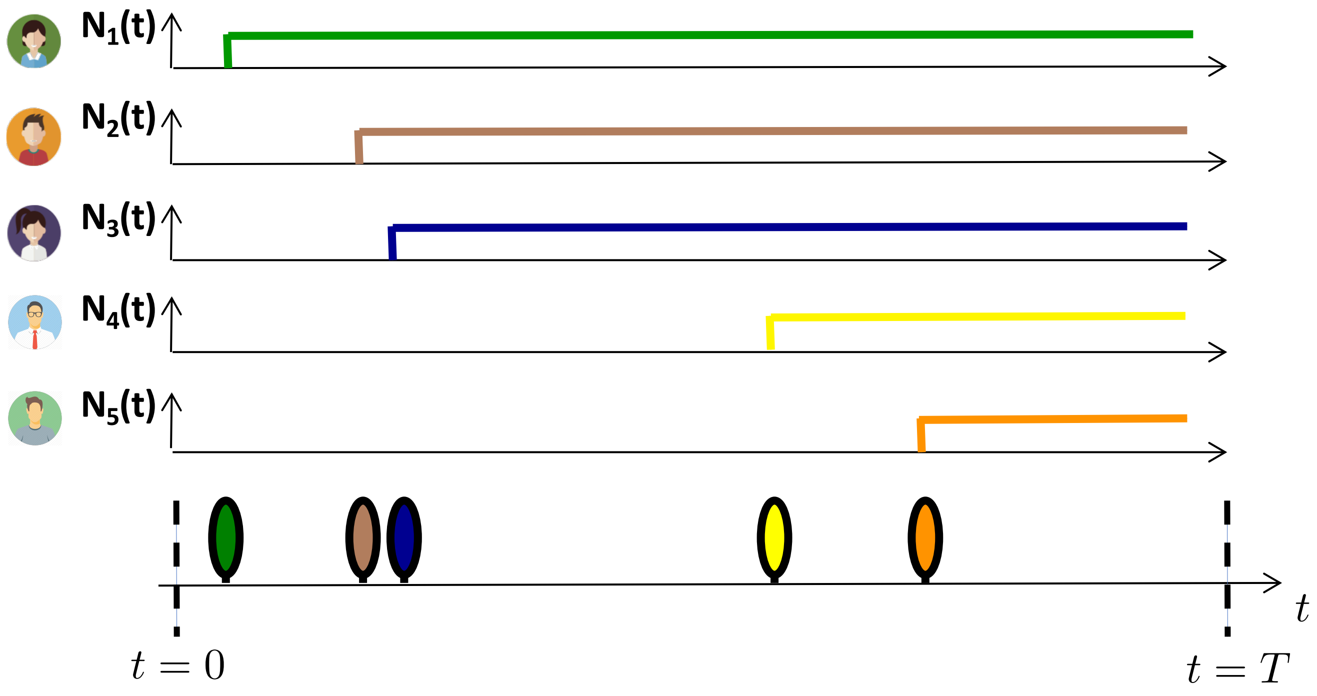
They can have an impact in the off-line world

theguardian

Click and elect: how fake news helped Donald Trump win a real election

Infection cascade representation

We represent an infection cascade using **terminating temporal point processes:**



Infection event:

$$(u_i, m_i, t_i)$$

User

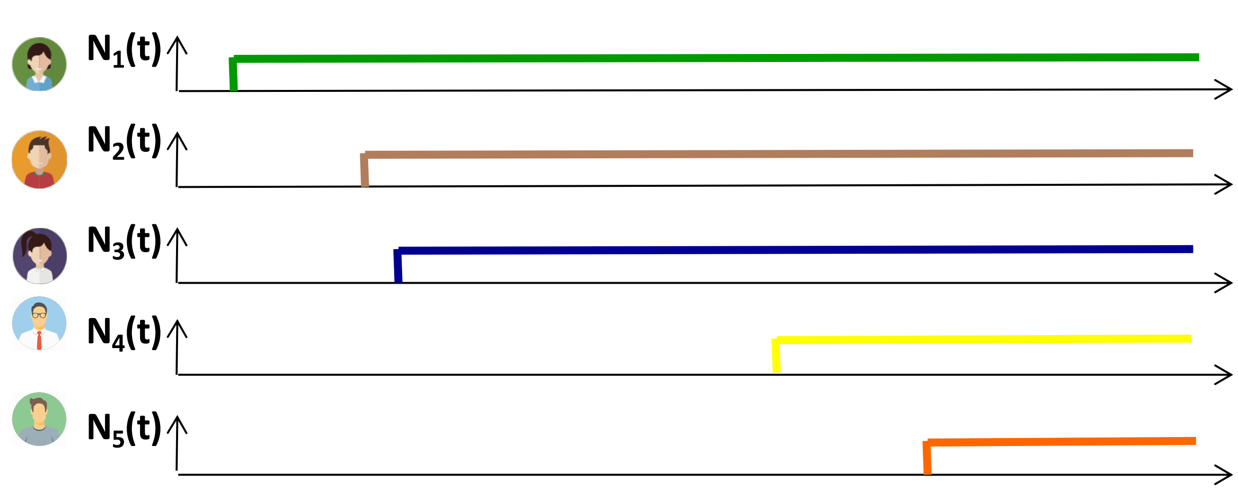
Cascade

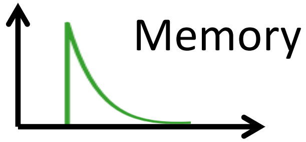
Time

Infection intensity

Source
(given, not modeled)

Follow-ups
(modeled)





$$\lambda_u^*(t) = (1 - N_u(t)) \sum_{v \in [m]} b_{vu} \sum_{e_i \in \mathcal{H}_v(t)} \kappa(t - t_i)$$

Users get infected only once

Influence from user v on user u

Previous infections of user v

[Gomez-Rodriguez et al., ICML 2011]

Model inference from multiple cascades

Conditional intensities

$$\lambda_u^*(t)$$

Diffusion log-likelihood

$$\mathfrak{L} = \sum_{u=1}^{n} \log \lambda_u^*(t_u) - \int_0^T \lambda_u^*(\tau)\, d\tau$$

Maximum likelihood approach to find model parameters!

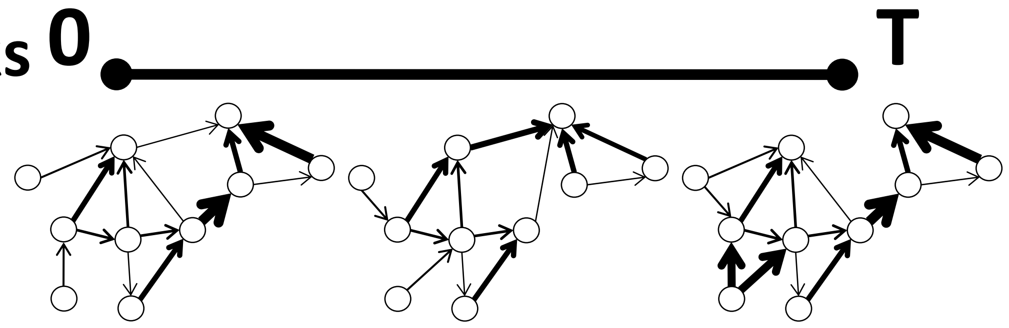
Sum up log-likelihoods of multiple cascades!

Theorem. For any choice of parametric memory, the **maximum likelihood** problem is **convex in B**.

[Gomez-Rodriguez et al., ICML 2011]

In some cases, influence change over time:

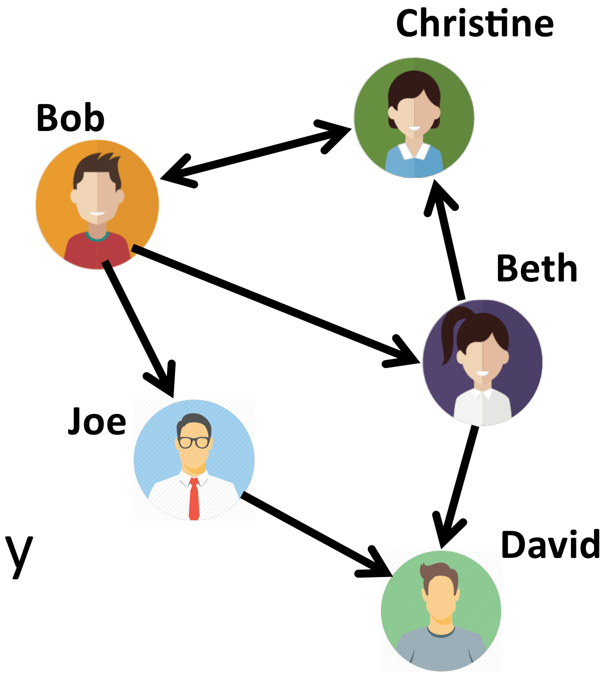
Propagation over networks with variable influence



[Gomez-Rodriguez et al., WSDM 2013]

Recurrent events: beyond cascades

Up to this point, each users is only infected once, and event sequences can be seen as cascades.



In general, users perform recurrent events over time. E.g., people repeatedly express their opinion online:

CNN
How social media is revolutionizing debates

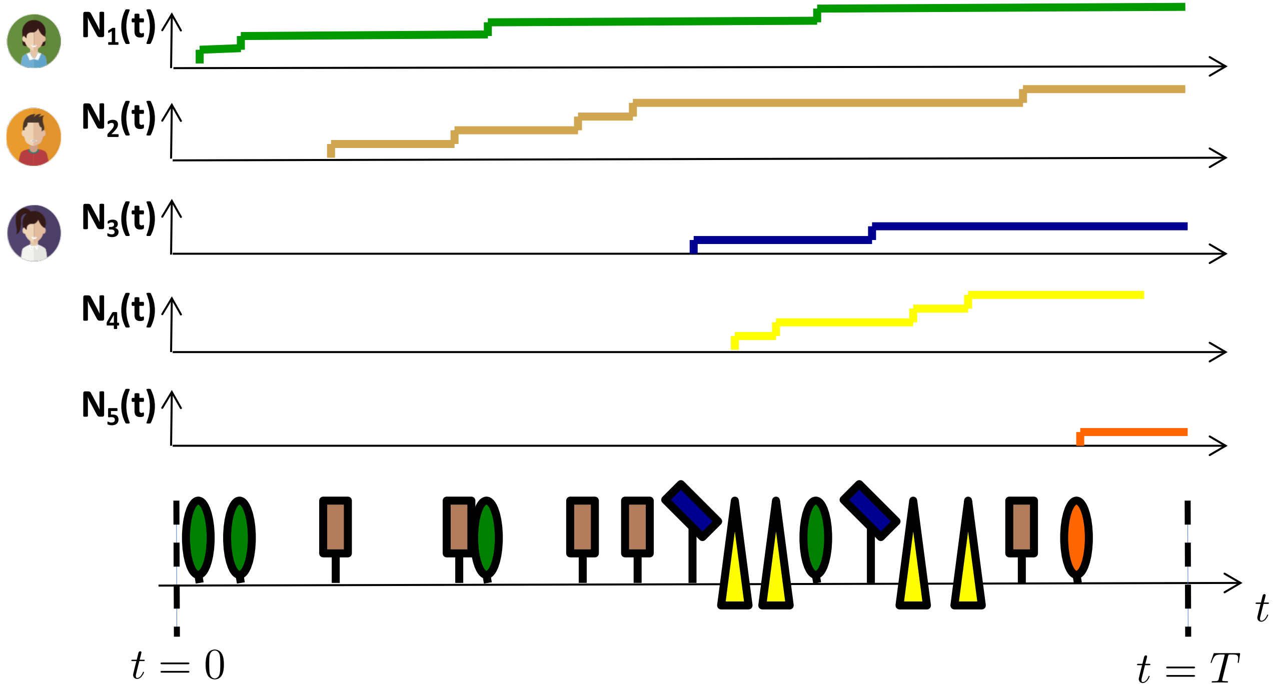
The New York Times
Social Media Are Giving a Voice to Taste Buds

TC
Twitter Unveils A New Set Of Brand-Centric Analytics

The New York Times
Campaigns Use Social Media to Lure Younger Voters

Recurrent events representation

We represent messages using **nonterminating temporal point processes**:

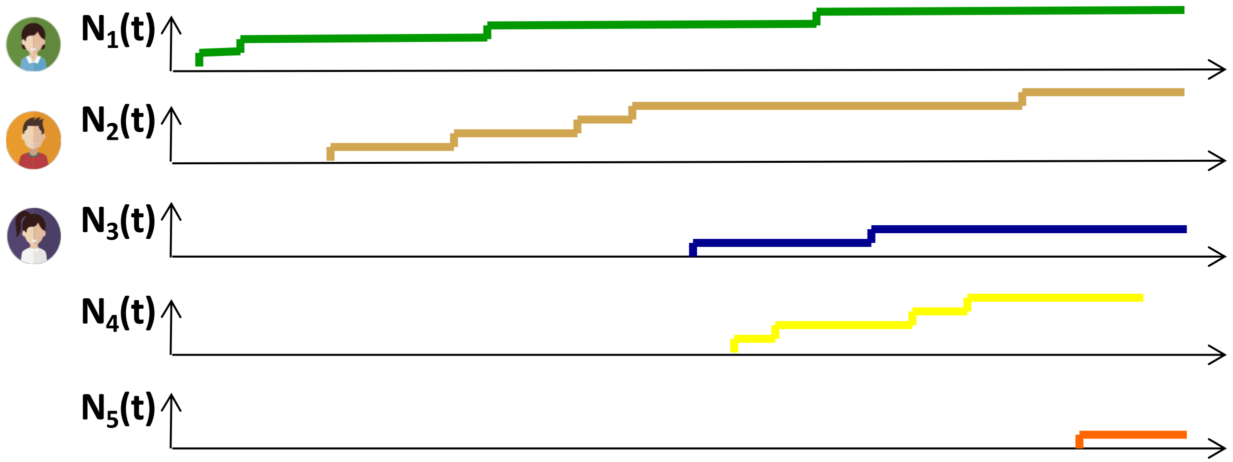


Recurrent event:

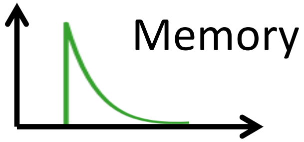
$$(u_i, t_i)$$

[Farajtabar et al., NIPS 2014]

Recurrent events intensity



Cascade sources!



$$\underbrace{\lambda_u^*(t)}_{\text{User's intensity}} = \underbrace{\mu_u}_{\text{Events on her own initiative}} + \sum_{v\in[m]} b_{vu} \underbrace{\sum_{e_i\in\mathcal{H}_v(t)}}_{\text{Previous messages by user v}} \kappa(t-t_i)$$

Influence from user v on user u

Hawkes process

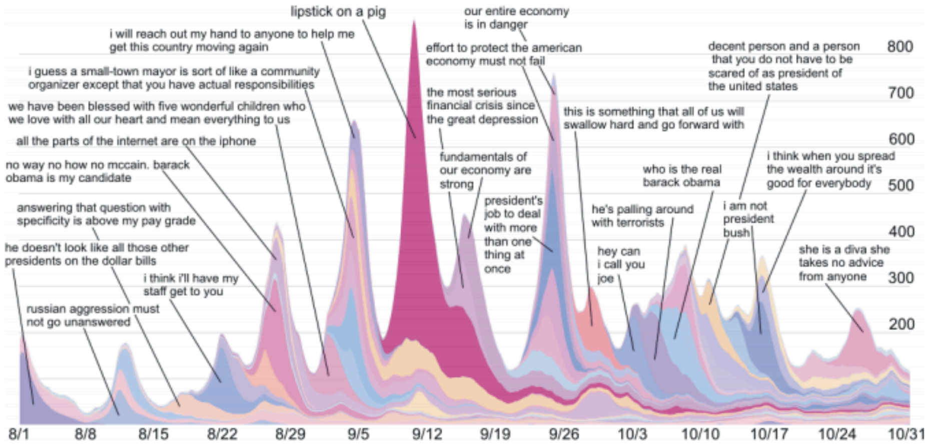
[De et al., NIPS 2016]

Models & Inference

1. Modeling event sequences
2. Clustering event sequences
3. Capturing complex dynamics
4. Causal reasoning on event sequences

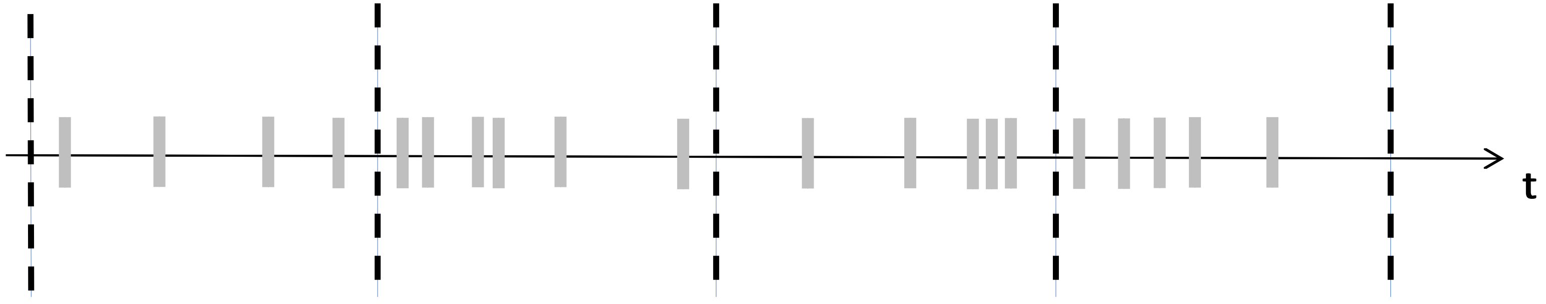
Event sequences

we have assumed the cascade (topic, etc.) that each event belongs to was known.

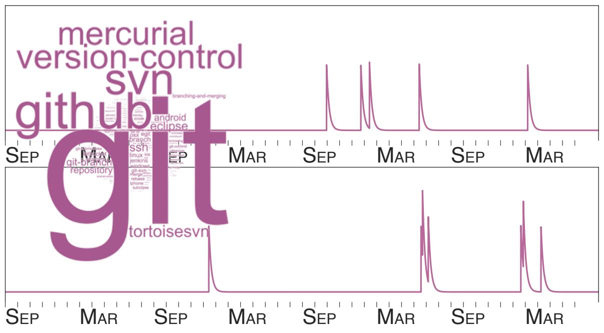


Often, the cluster (topic, etc.) that each event in a sequence belongs to is not known:

Assume the event **cluster to be hidden** and aim to automatically **learn the cluster assigments** from the data:

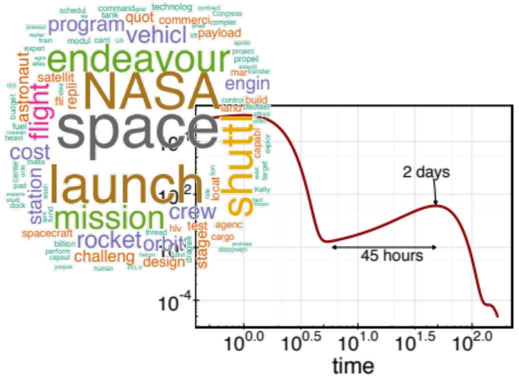


Bayesian methods to cluster event sequences in the context of:



Learning

Online News



Health care

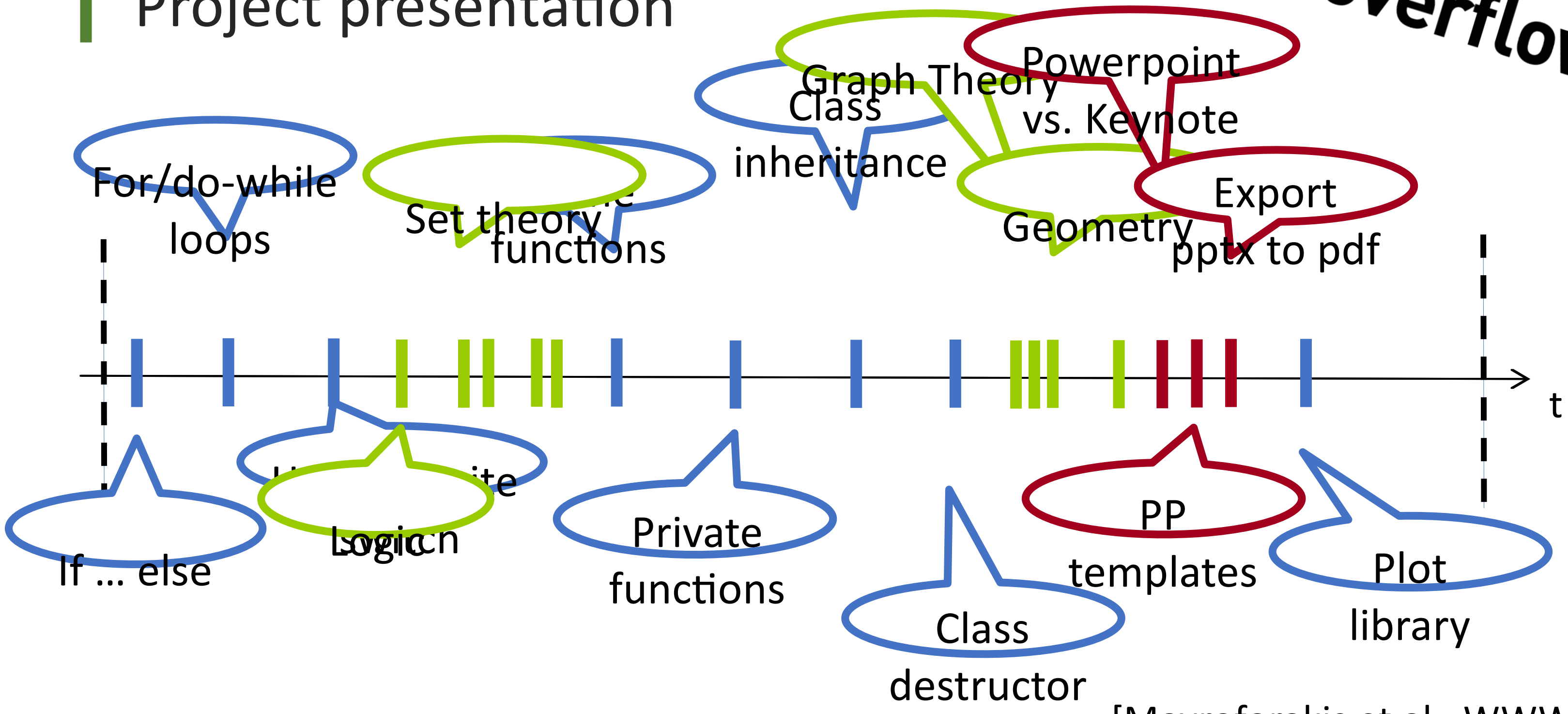
Method	DMHP
ICU Patient	**0.3778**
IPTV User	**0.2004**

[Du et al., 2015; Mavroforakis et al., 2017; Xu & Zha, 2017]

Hierarchical Dirichlet Hawkes process

1st year computer science student

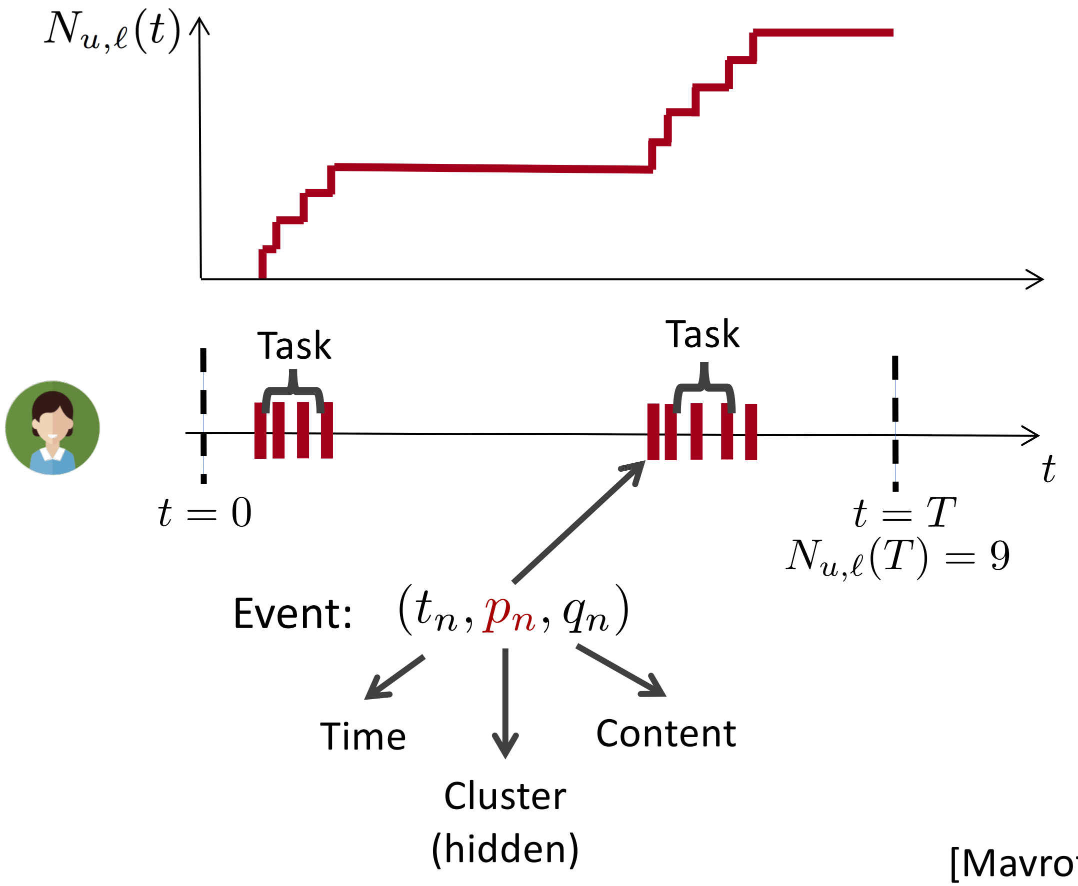
- Introduction to programming
- Discrete math
- Project presentation



[Mavroforakis et al., WWW 2017]

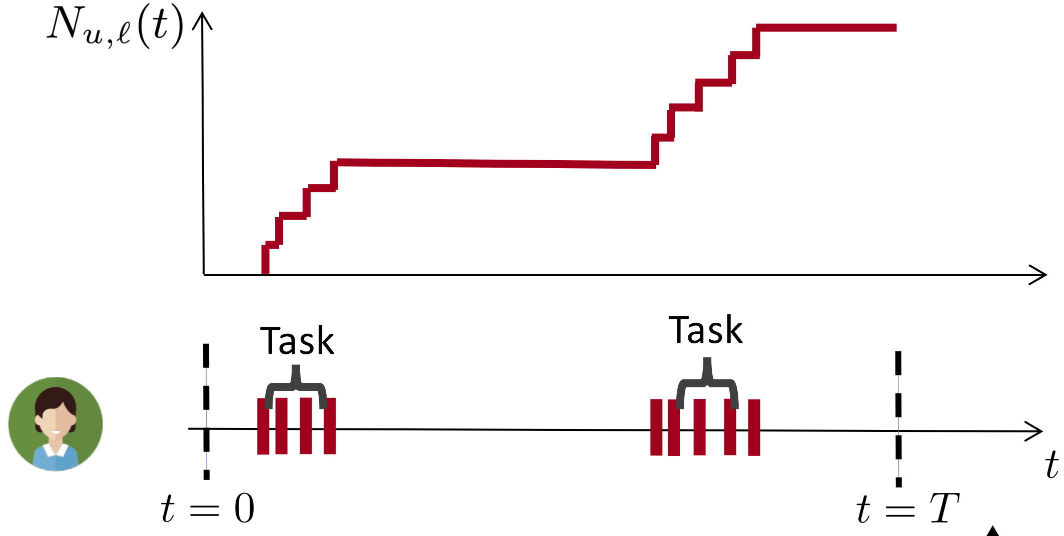
Events representation

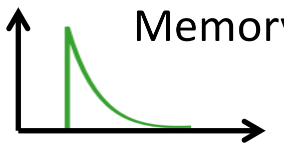
We represent the events using **marked temporal point processes**:



[Mavroforakis et al., WWW 2017]

Cluster intensity





Hawkes process

New cascade rate

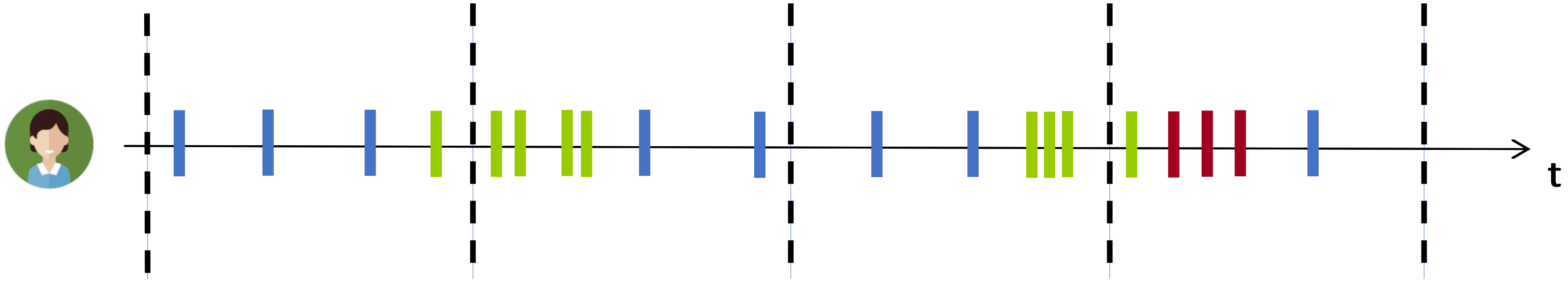
Cluster popularity

$$\underbrace{\lambda^*_{u,\ell}(t)}_{\substack{\textbf{Intensity}\\ \textbf{or rate}\\ \textbf{(events / hour)}}} = \underbrace{\mu_u \pi_\ell}_{\substack{\textbf{Own}\\ \textbf{initiative}}} + \underbrace{\sum_{j:t_j \in \mathcal{H}_{u,\ell}(t)} k_{\theta_\ell}(t - t_j)}_{\textbf{Follow-up}}$$

[Mavroforakis et al., WWW 2017]

User events intensity

Users adopt more than one cluster:



A user’s learning events as a multidimensional Hawkes:

Time, cluster

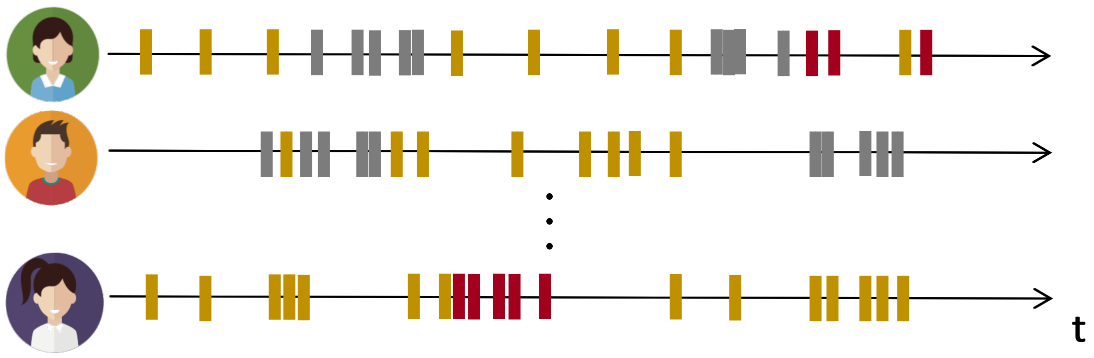
$$(t_n, p_n) \sim Hawkes\begin{pmatrix} \lambda^*_{u,1}(t) \\ \vdots \\ \lambda^*_{u,\infty}(t) \end{pmatrix}$$

Content $\rightarrow$ $= \boldsymbol{\omega}\ q_n \sim P(\cdot|\theta_{p_n})$ $\quad \omega_j \sim Multinomial(\boldsymbol{\theta}_p)$

[Mavroforakis et al., WWW 2017]

People share same clusters

Different users adopt same clusters



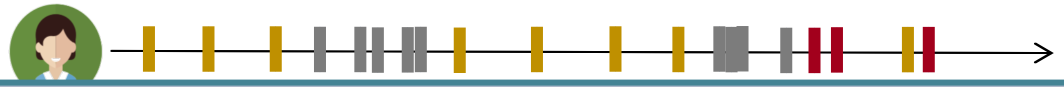
Cluster distribution from a **Dirichlet process:**

- Infinite # of clusters.
- Shared parameters across users.

[Mavroforakis et al., WWW 2017]

People share same clusters

Different users adopt same clusters



Efficient model inference using Sequential Montecarlo!

Clus

-

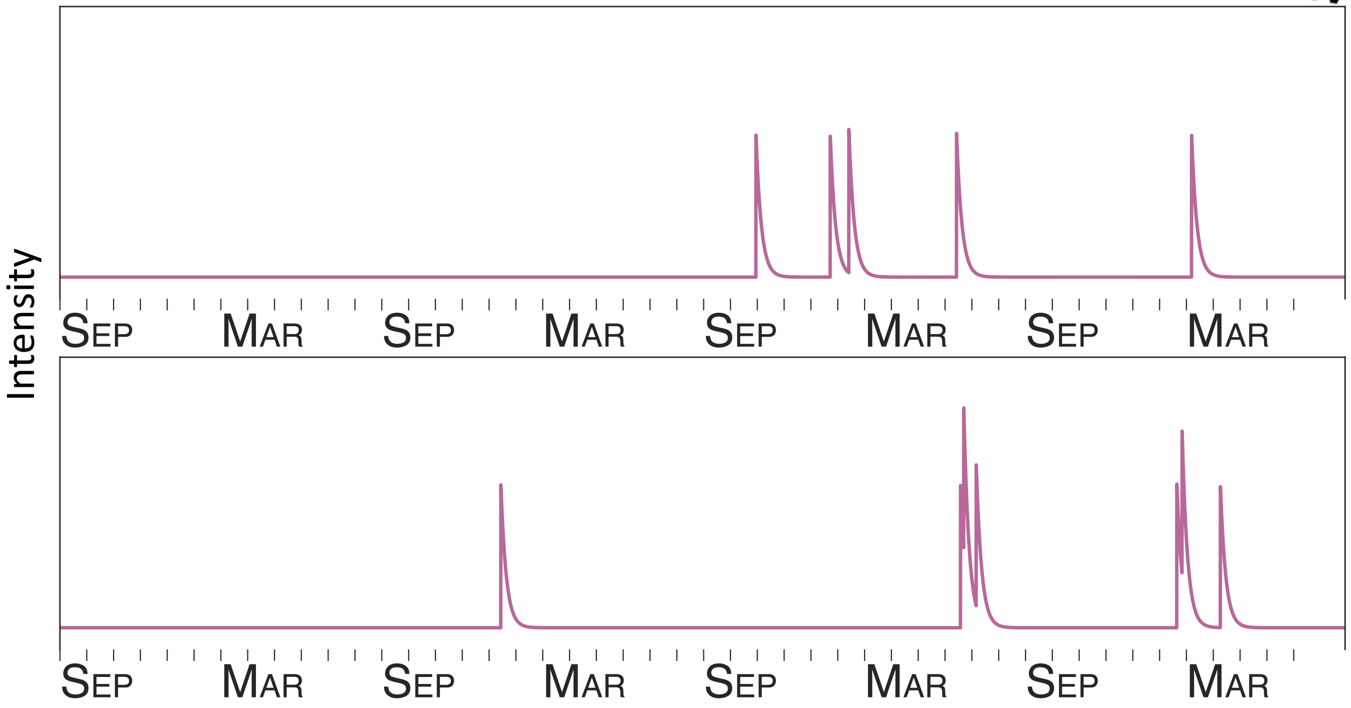
- Shared parameters across users.

[Mavroforakis et al., WWW 2017]

Version Control vs Machine Learning

Content

Intensities

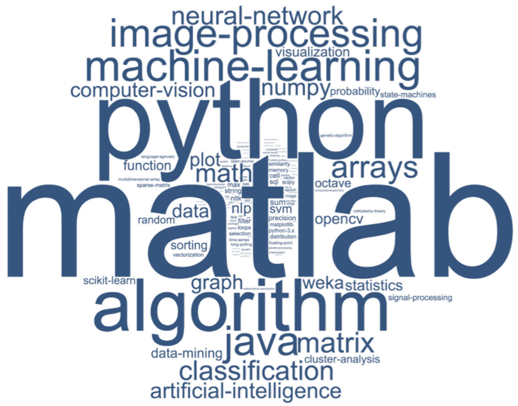


Version control tasks tend to be specific, quickly solved after performing few questions

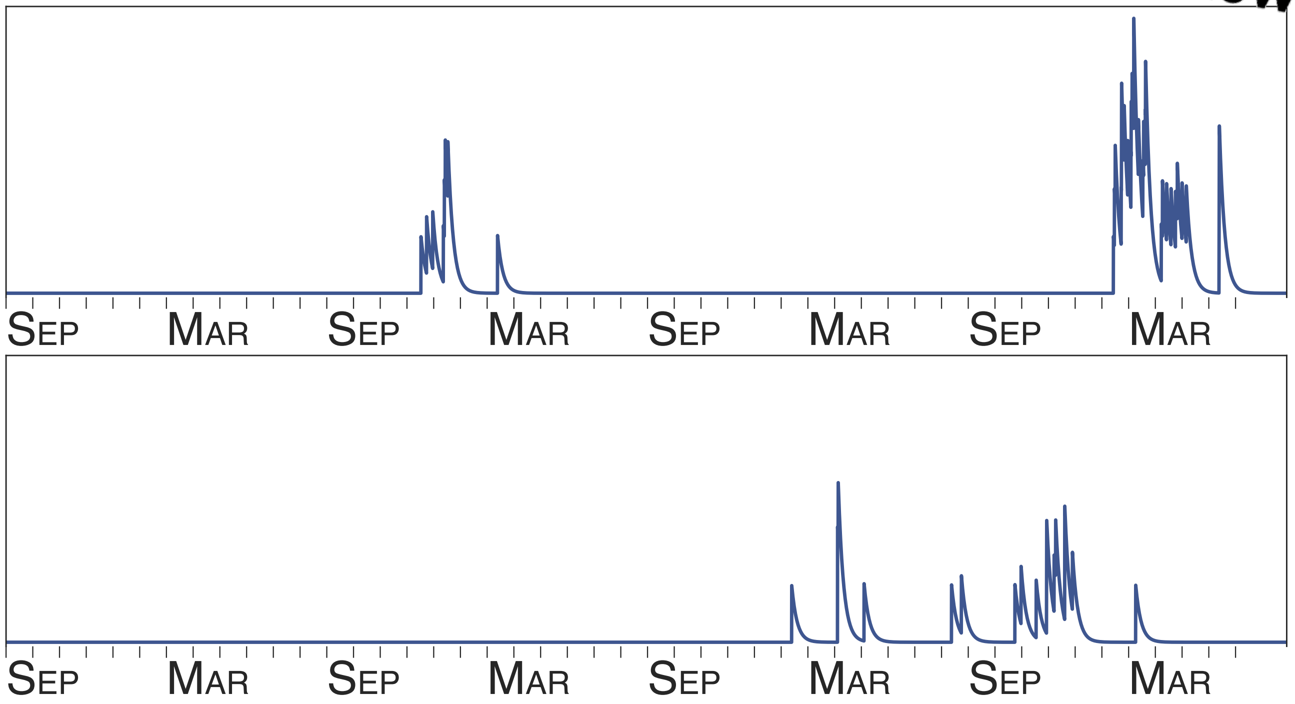
[Mavroforakis et al., WWW 2017]

Version Control vs Machine Learning

Content



Intensities



Machine learning tasks tend to be more complex and require asking more questions

[Mavroforakis et al., WWW 2017]

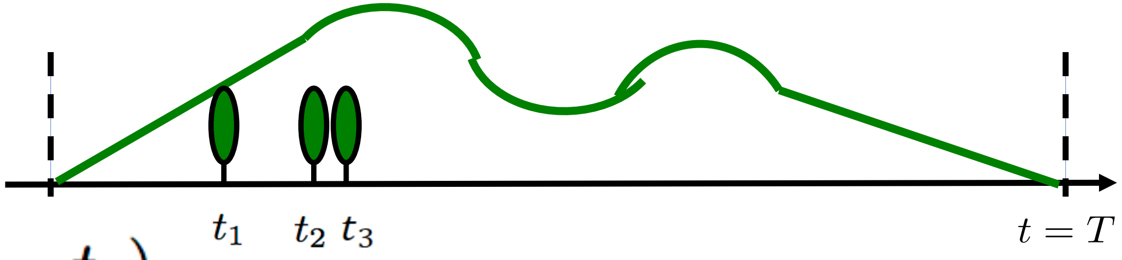
Models & Inference

1. Modeling event sequences
2. Clustering event sequences
3. **Capturing complex dynamics**
4. Causal reasoning on event sequences

RNN to Capture Complex Dynamics

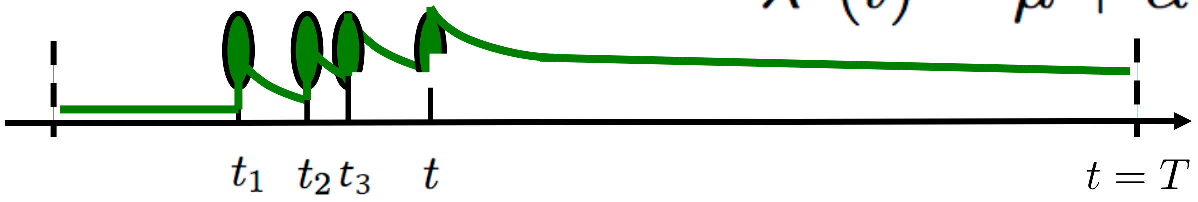
Up to now, we have focused on simple temporal dynamics (and intensity functions):

$$\lambda^*(t) = \mu$$



$$\lambda^*(t) = \sum_j \alpha_j \, k(t - t_j)$$

$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i)$$



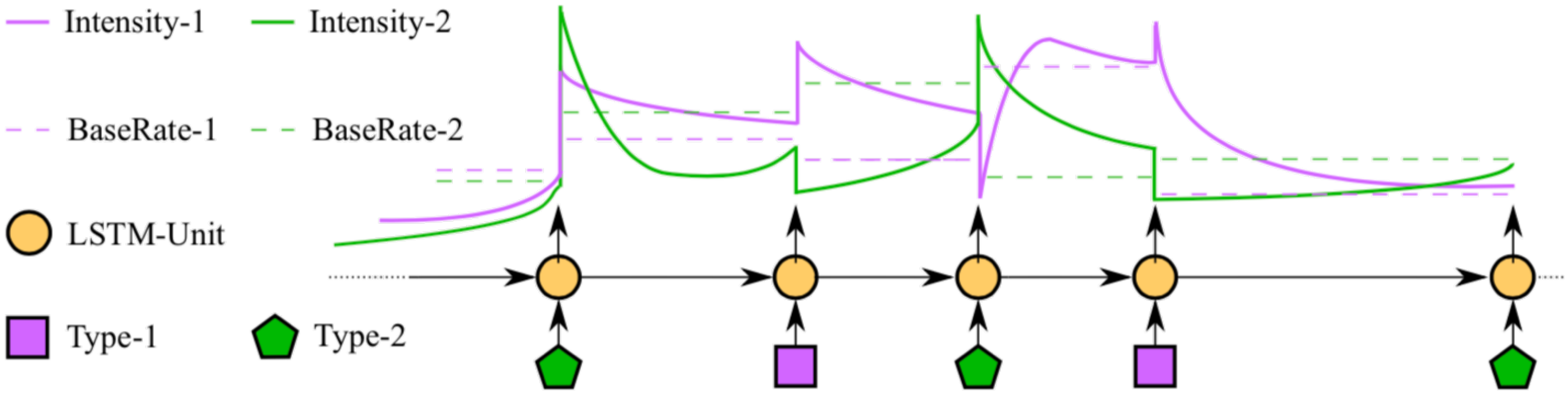
Recent works make use of RNNs to capture more complex dynamics

[Du et al., 2016; Dai et al., 2016; Mei & Eisner, 2017; Jing & Smola, 2017; Trivedi et al., 2017; Xiao et al., 2017a; 2018]

Neural Hawkes process

1) History effect does not need to be additive

2) Allows for complex memory effects
   (such as delays)



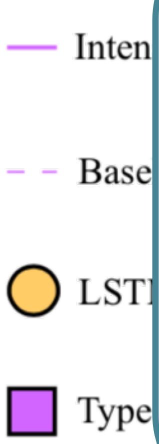
[Mei & Eisner, NIPS 2017]

Neural Hawkes process

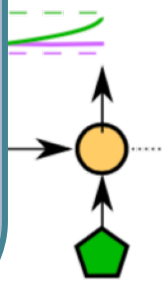
$$\lambda_u(t) = f_u(\mathbf{w}_u^\top \mathbf{h}(t))$$

Excitation & inhibition

$$\mathbf{h}(t) = \mathrm{RNN}(\overbrace{\mathcal{H}(t)}^{\textbf{Memory}})$$



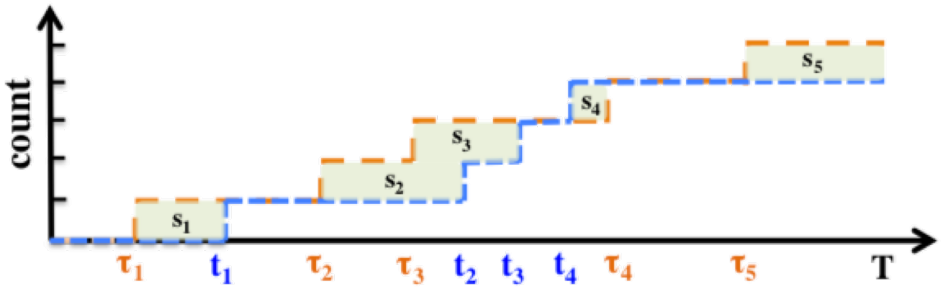
Parameter learning using stochastic gradient descent



[Mei & Eisner, NIPS 2017]

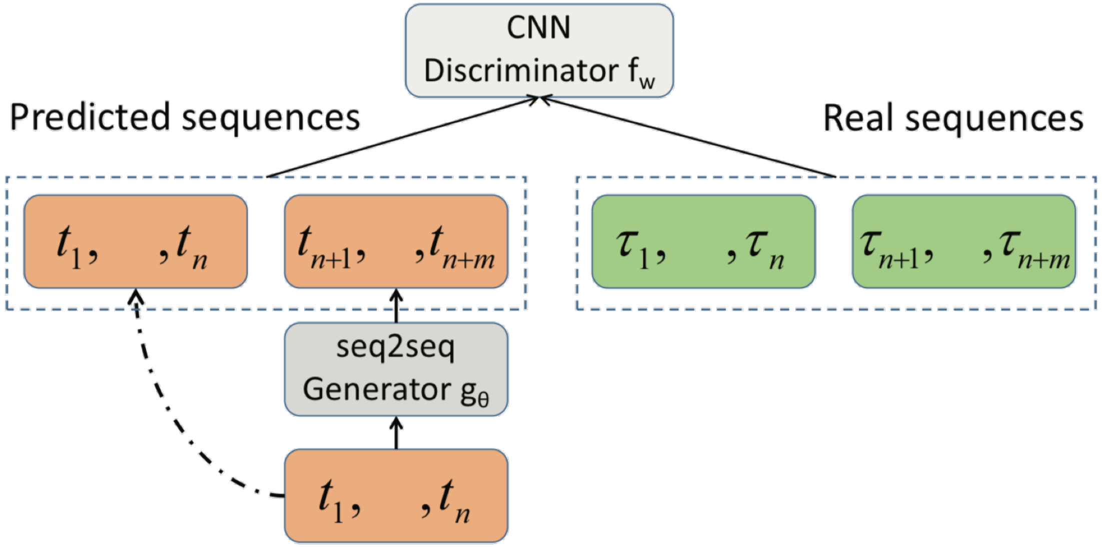
Applications (I): Predictive Models

Key idea: Intensity- and likelihood-free models



Wasserstein-Distance for Temporal Point Processes

GAN architecture

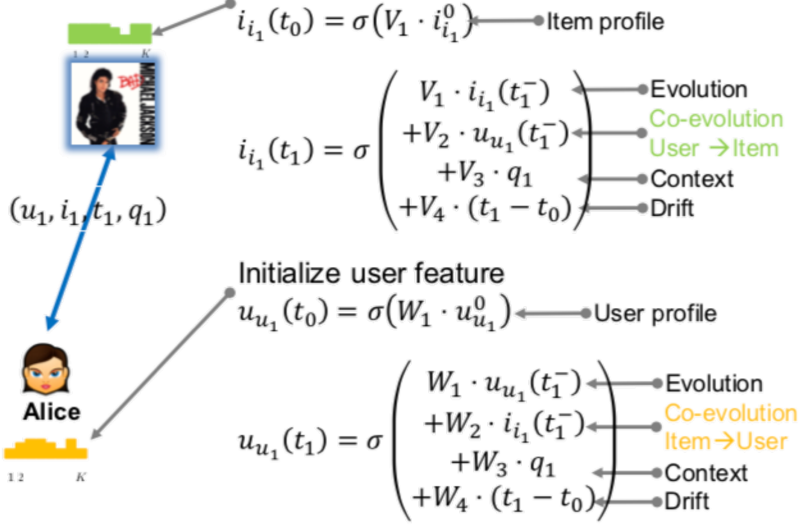


[Xiao et al., 2017 & 2018]

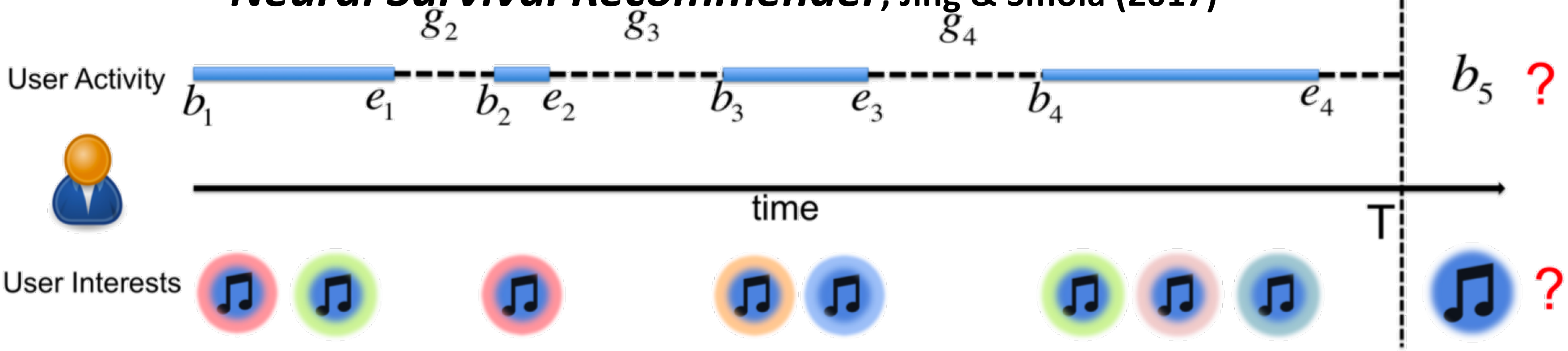
Applications (I): Predictive Models

Know-Evolve, Trivedi et al. (2017)

Coevolutionary Embedding, Dai et al. (2017)



Neural Survival Recommender, Jing & Smola (2017)



Models & Inference

1. Modeling event sequences
2. Clustering event sequences
3. Capturing complex dynamics
4. Causal reasoning on event sequences

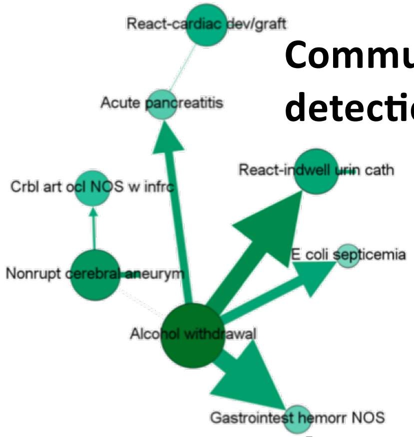
Temporal point processes beyond prediction

So far, we have focused on models that improve preditions:

Link prediction

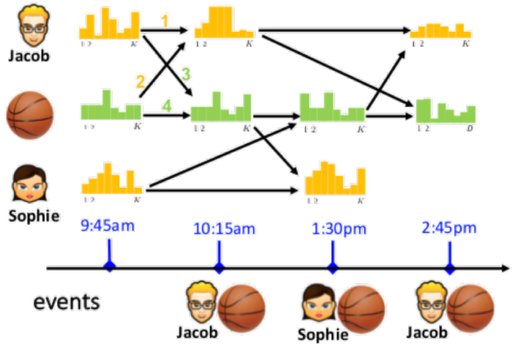
[Trivedi et al., 2017]

Community detection



[Xiao et al., 2017]

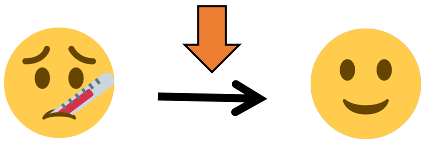
Recommendations



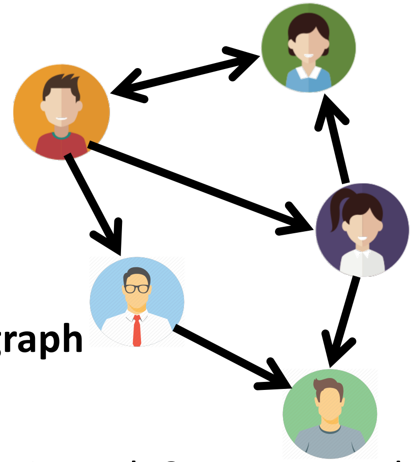
[Dai et al., 2017]

Recent works have focused on performing **causal inference using event sequences**:

Treatment effect



Granger causality graph



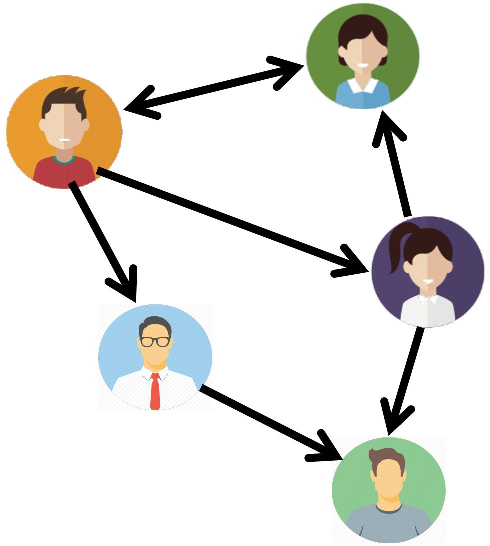
[Xu et al., 2016; Achab et al., 2017; Kuśmierczyk & Gomez-Rodriguez, 2018]

Uncovering Causality from Hawkes Processes

Multivariate Hawkes process:

$$N(t) = \sum_{u \in \mathcal{U}} N_u(t)$$

$$\lambda_u(t) = \mu_u + \sum_{v \in \mathcal{U}} \int_0^t k_{u,v}\underbrace{(t - t')}_{\textbf{Effect of v's past events on u}} dN_v(t')$$



Granger causality:

"X causes Y in the sense of Granger causality if forecasting future values of Y is more successful while taking X past values into account"

[Granger, 1969]

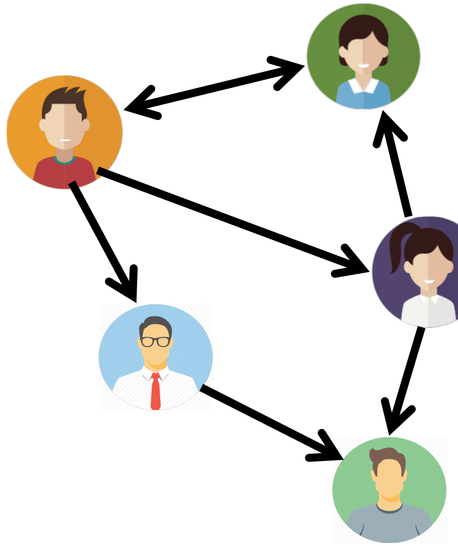
[Achab et al., ICML 2017]

Uncovering Causality from Hawkes Processes

Multivariate Hawkes process:

$$N(t) = \sum_{u \in \mathcal{U}} N_u(t)$$

$$\lambda_u(t) = \mu_u + \sum_{v \in \mathcal{U}} \int_0^t k_{u,v}\underbrace{(t - t')}_{\text{Effect of v's past events on u}} dN_v(t')$$



Granger causality on multivariate Hawkes processes:

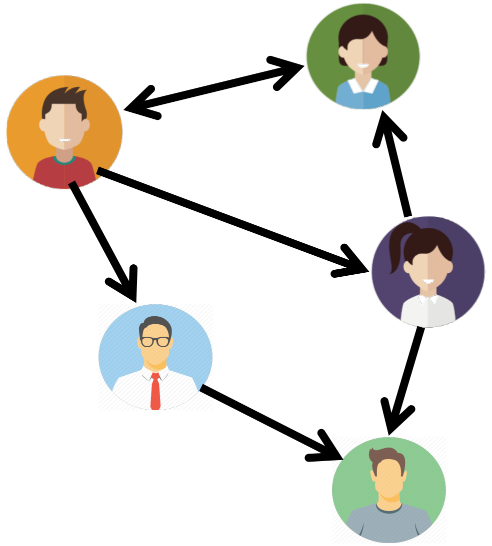
“ $N_v(t)$ does not Ganger-cause $N_u(t)$ w.r.t. $N(t)$ if and only if $k_{u,v}(\tau) = 0$ for $\tau \in \Re^+$ "

[Eichler et al., 2016]

[Achab et al., ICML 2017]

Uncovering Causality from Hawkes Processes

Goal is to estimate $G = [g_{uv}]$**, where:**



$$g_{uv} = \int_0^{+\infty} k_{u,v}(\tau) d\tau \geq 0 \text{ for all } u, v \in \mathcal{U}$$

Average total # of events of node u whose *direct* ancestor is an event by node v

Then, $G = [g_{uv}]$ **quantifies the** ***direct causal relationship*** **between nodes.**

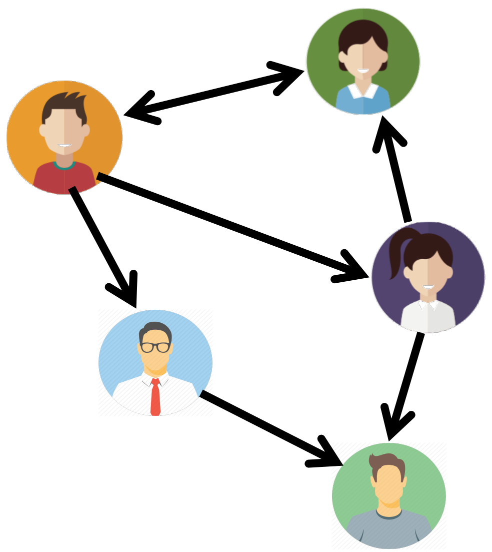
[Achab et al., ICML 2017]

Uncovering Causality from Hawkes Processes

Goal is to estimate $G = [g_{uv}]$**, where:**

$$g_{uv} = \int_0^{+\infty} k_{u,v}(\tau) d\tau \geq 0 \text{ for all } u, v \in \mathcal{U}$$

Average total # of events of node u whose *direct* ancestor is an event by node v



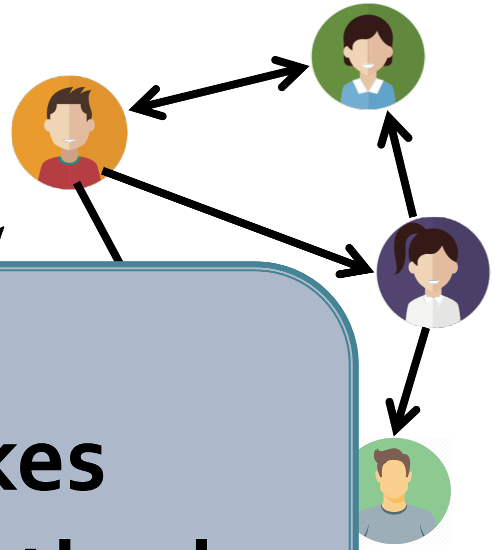
Then, $G = [g_{uv}]$ **quantifies the *direct causal relationship* between nodes.**

Key idea: Estimate G using the cumulants the dN(t) of the Hawkes process.

[Achab et al., ICML 2017]

Uncovering Causality from Hawkes Processes

Goal is to estimate $G = [g_{uv}]$, where:



$$g_{uv} = \int^{+\infty} k \quad (\tau) d\tau \geq 0 \text{ for all } u, v \in \mathcal{U}$$

Non parametric Hawkes cumulant estimation method

(with TensorFlow implementation)

The ... ***hip***

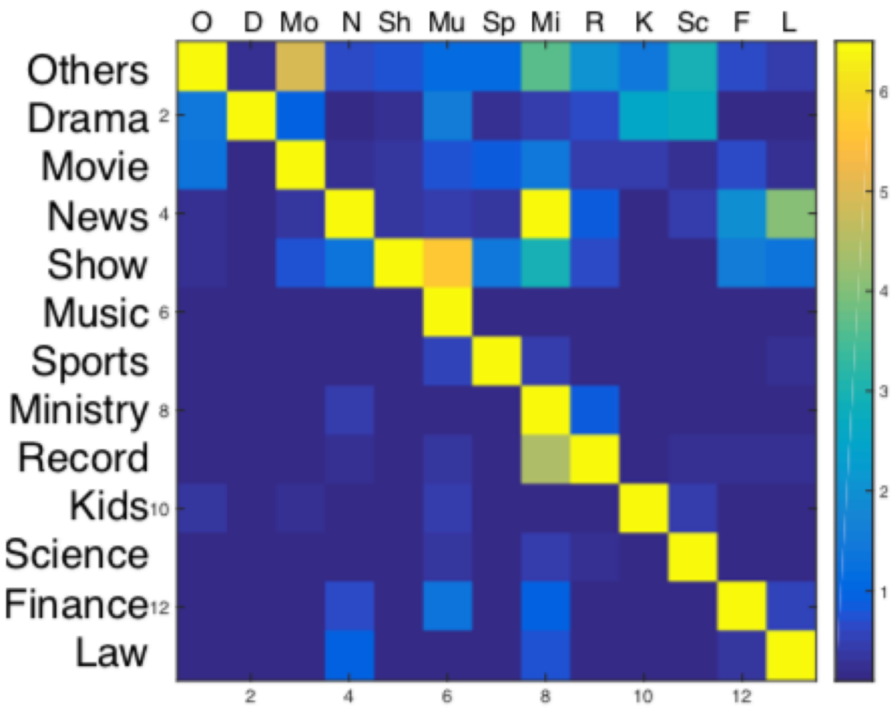
bet

Key idea: **Estimate G using the cumulants the dN(t) of the Hawkes process.**

[Achab et al., ICML 2017]

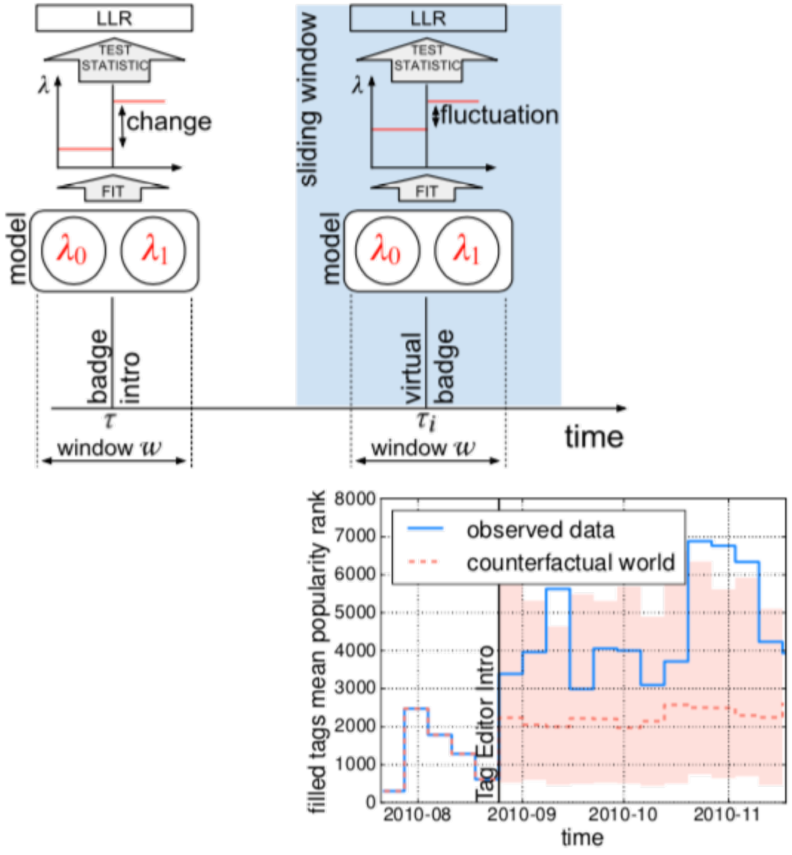
Causal reasoning: Applications

Infectivity matrix estimation



[Xu et al., 2016]

Effect of Badges



Tag wiki rank over time

[Kuśmierczyk & Gomez-Rodriguez, 2018]