Neural Networks for point processes

Alexey Zaytsev

Assistant professor, Skoltech

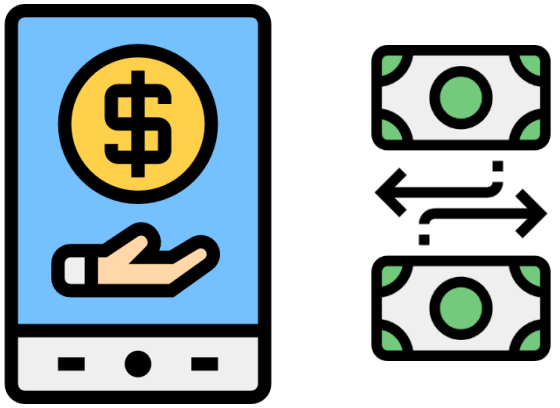
22th of December

Based on ICML Tutorial, July 2018 by I. Valera & M.G. Rodriguez
Slides/references: http://learning.mpi-sws.org/tpp-icml18

Skoltech

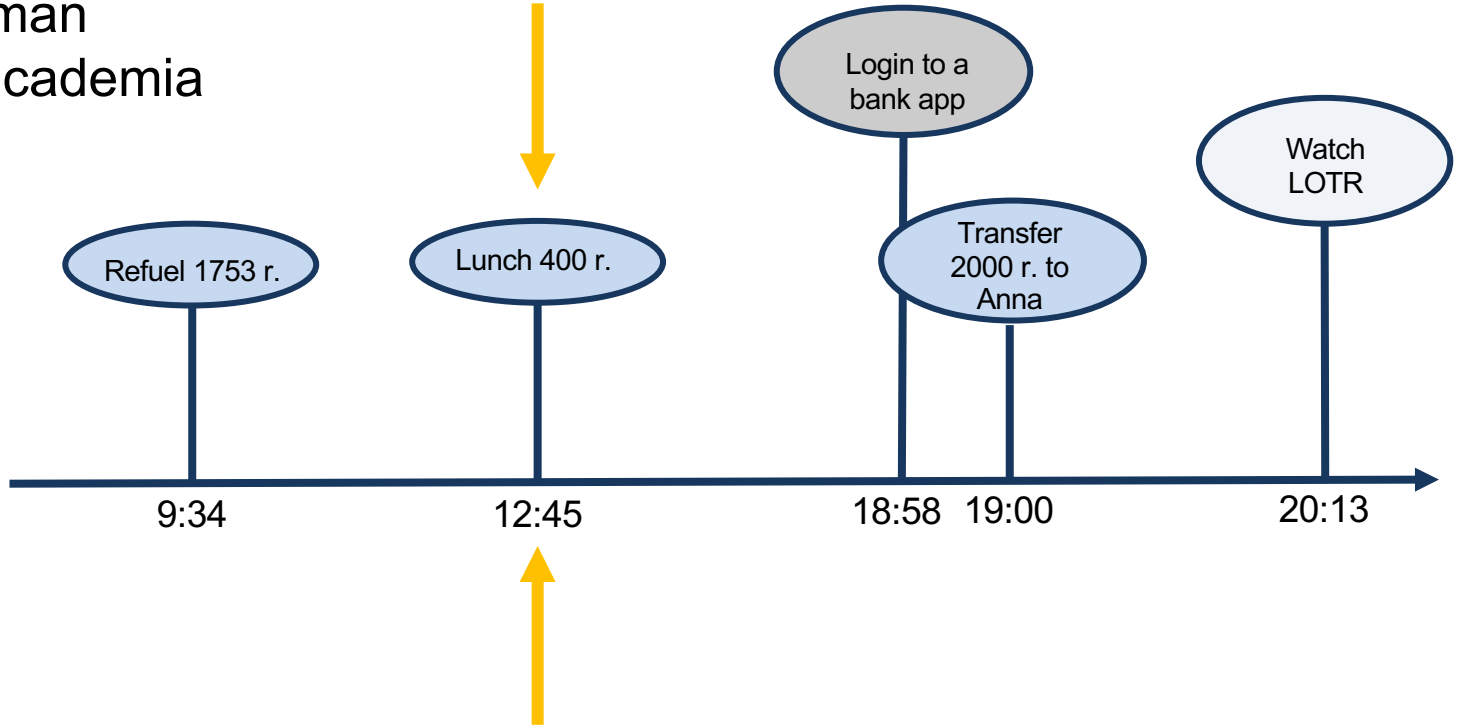
Temporal Point Processes (TPPs): applied problems

Example: financial transactions are event sequences



Alex, 33, man
works in academia

Event features:
sum, geolocation

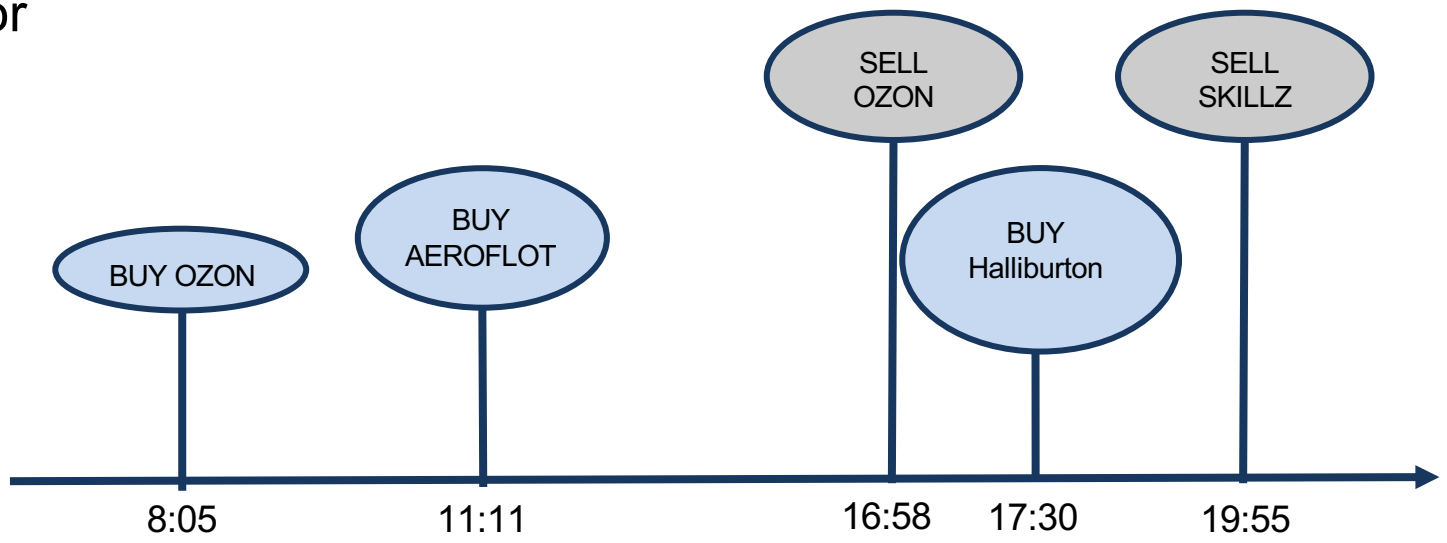


Event type + event time

Skoltech

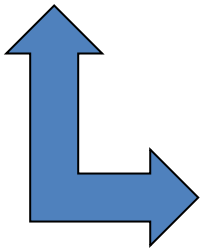
Example: operations in markets are event sequences

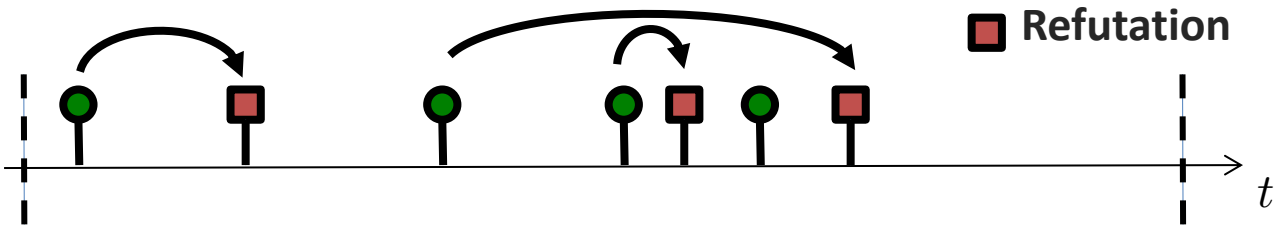
An investor

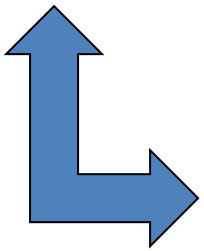


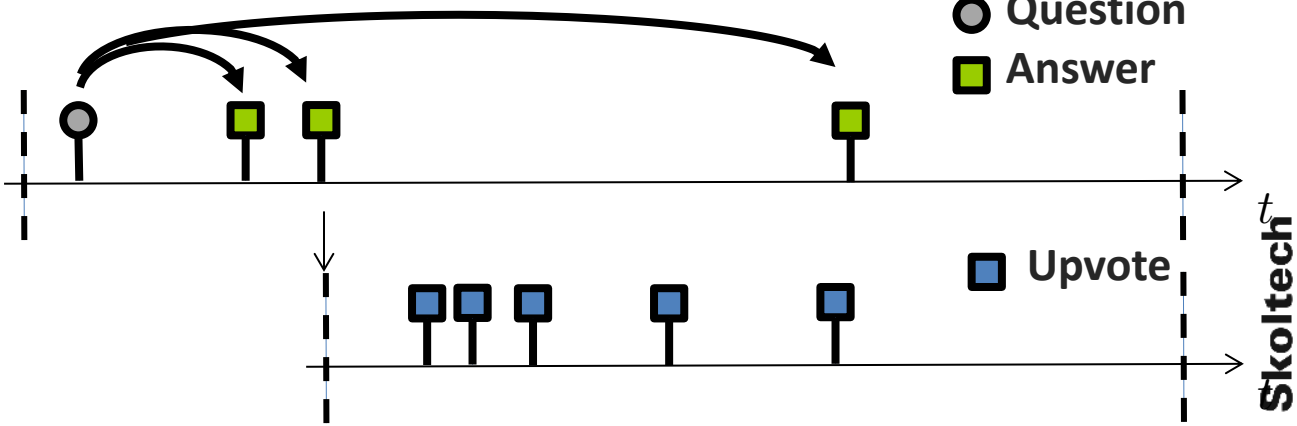
Skoltech

More complex example: response history





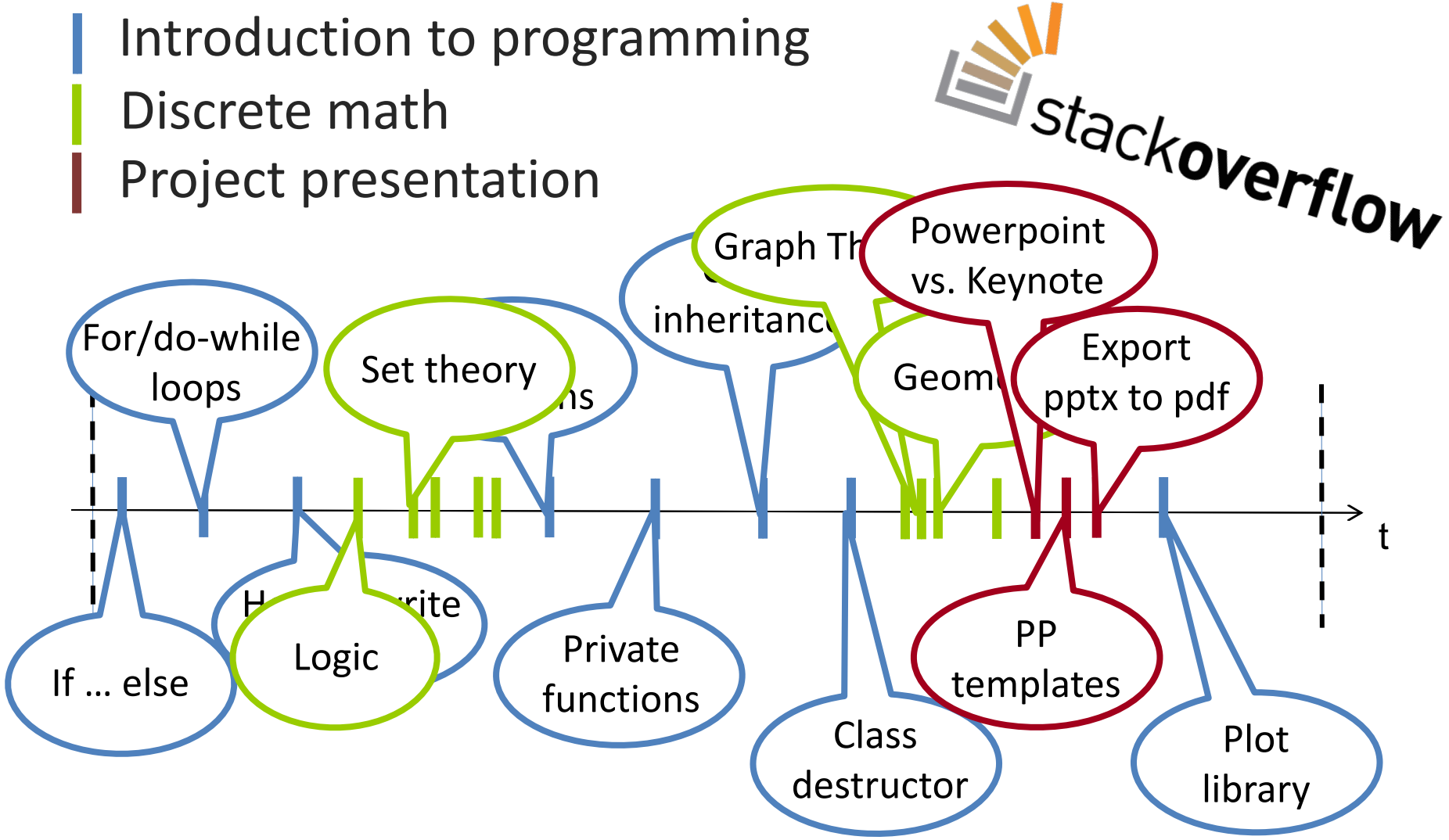




Skoltech

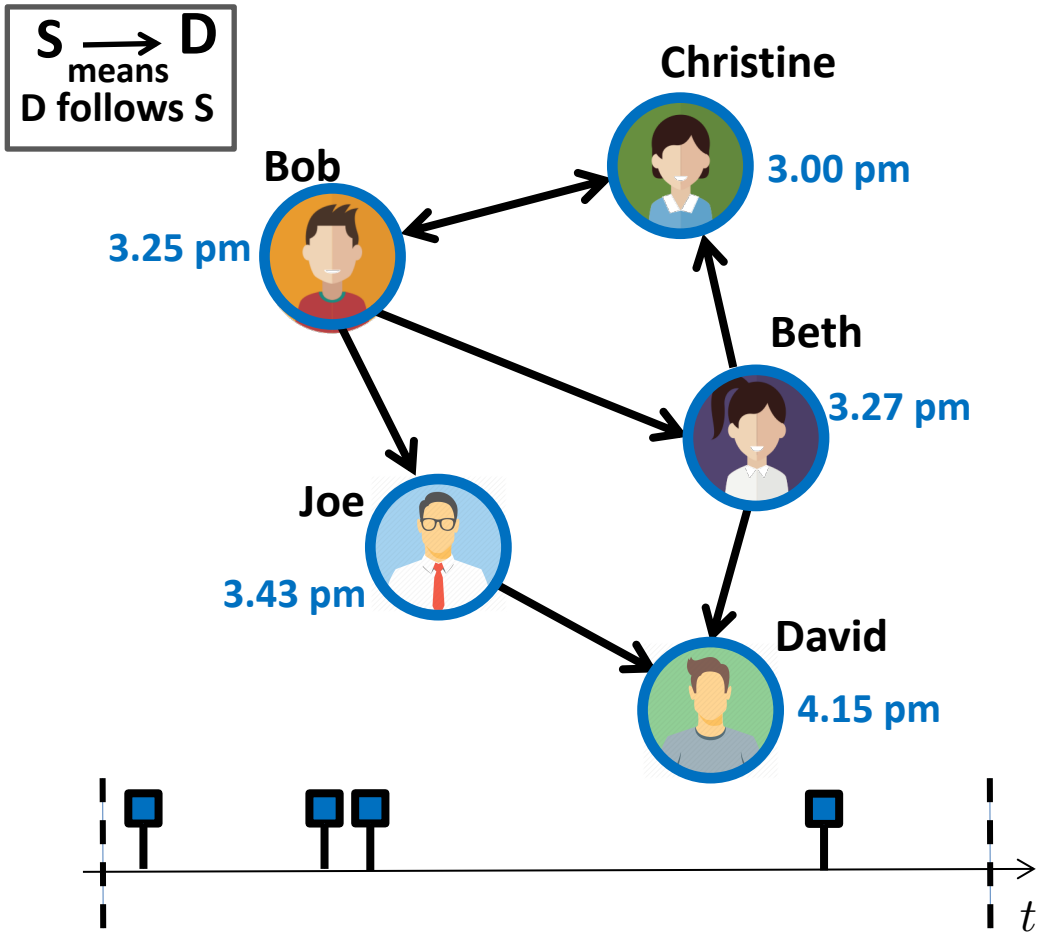
Example: development

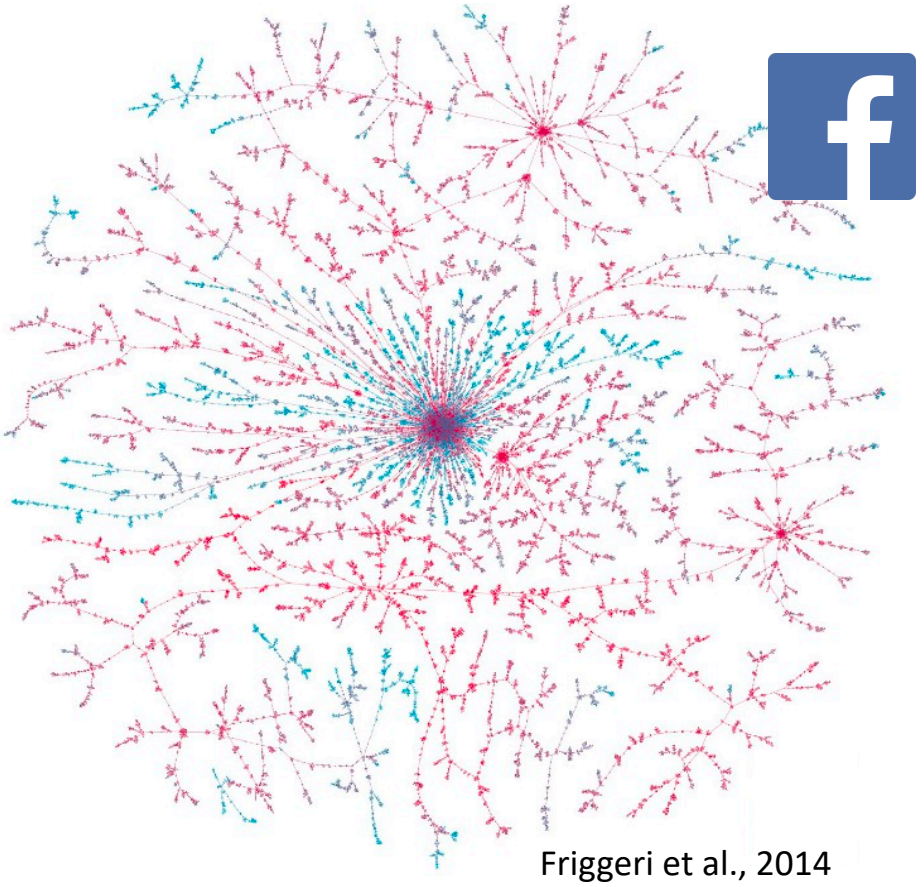
1st year computer science student



Skoltech

Example: Information propagation in graphs



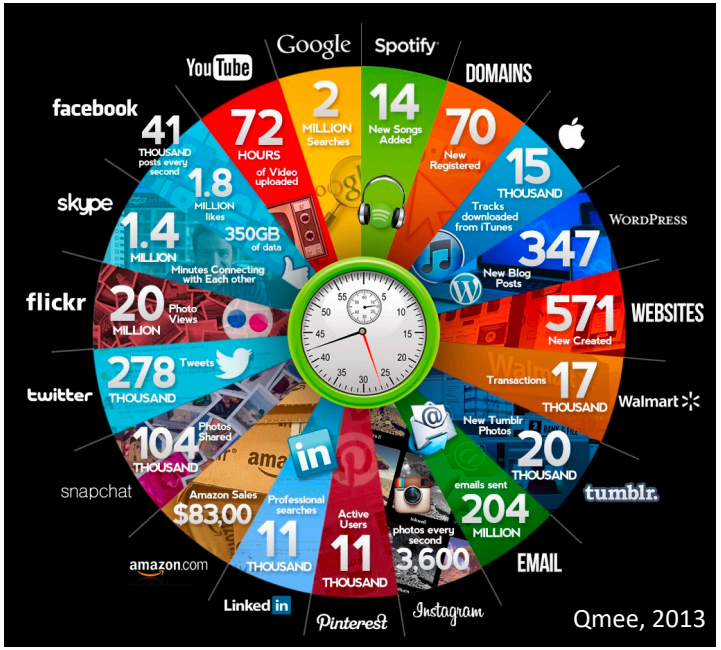


Friggeri et al., 2014

These cascades can deliver valuable insights

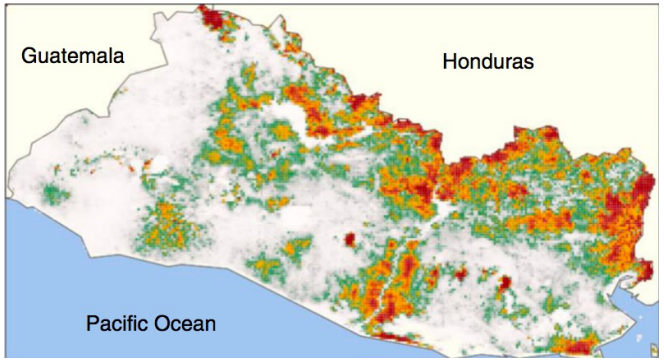
Skoltech

Many discrete *events* in continuous time



Clickstreams: Online actions of a user

Financial trading



Disease dynamics

Financial transactions

Skoltech

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

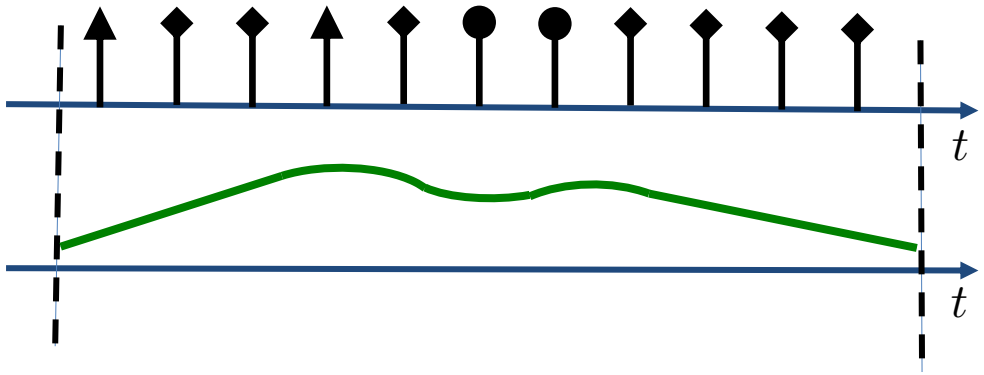
Ride-sharing requests

A user's reputation in Quora

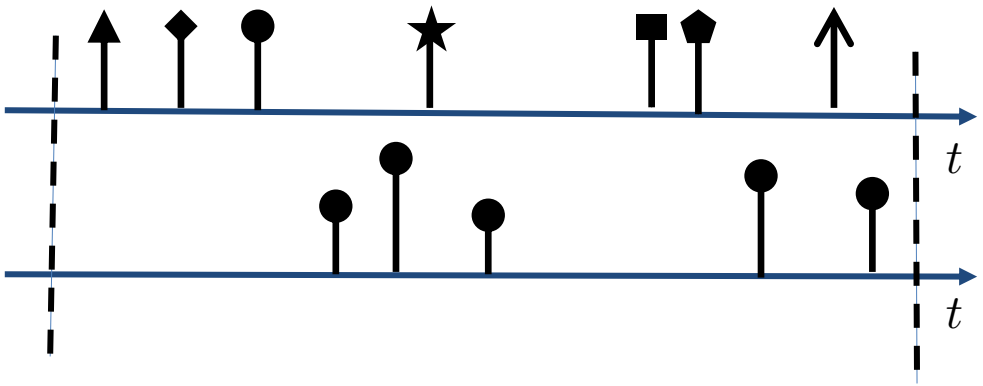
FAST — SLOW

…in a wide range of temporal scales.

Aren't these event traces just time series?

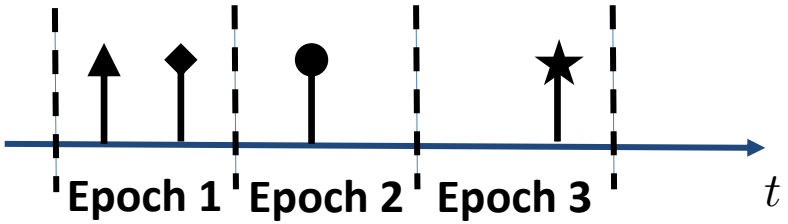


Discrete and continuous times series



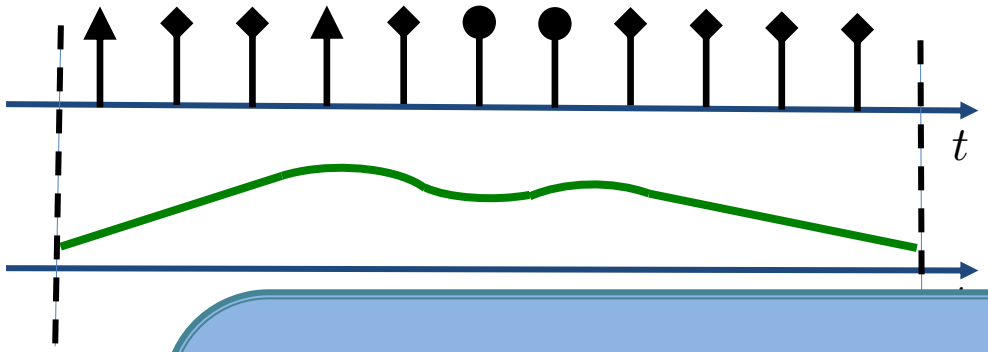
Discrete events in continuous time

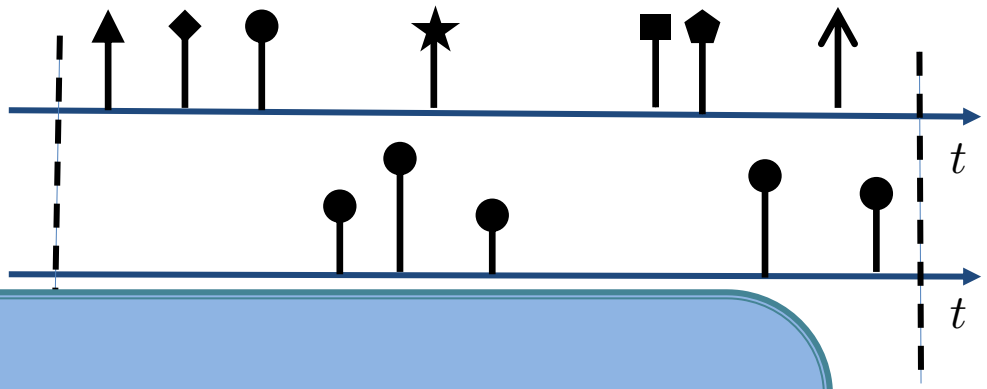
What about aggregating events in *epochs*?



- **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?





Dis

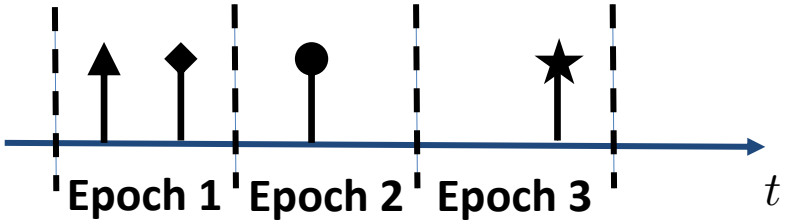
W

epoch?

What if no event in one epoch?

What about time-related queries?

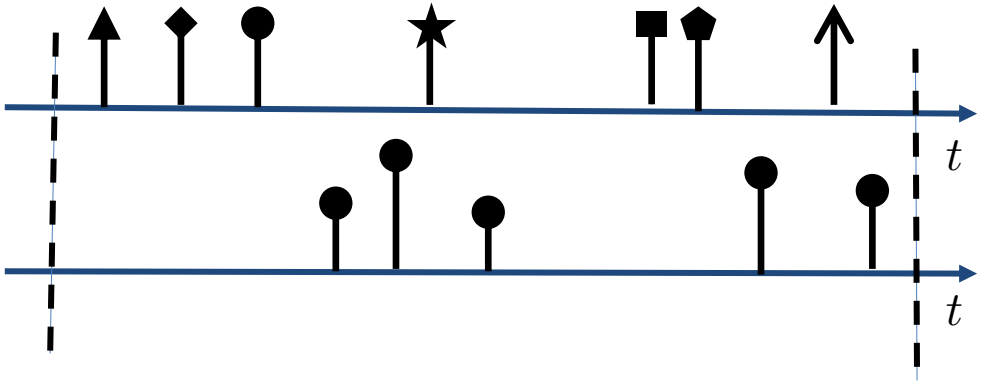
The framework of
temporal point processes
provides a ***native representation***



Skoltech

Problems for event sequences we'll solve!

- Compact description of data: models
- Forecasting/Prediction: distribution for the next event time; event type
- Interpretation: what causes what?
- Control: what should we do?
- Hypothesis testing
- Simulation

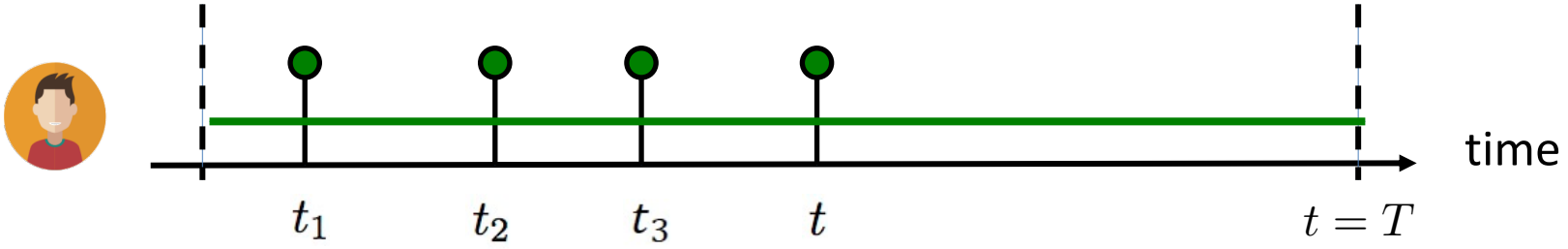


Point process: generates discrete events in continuous time

Poisson process example and estimation for it; basic terms

Poisson process

Coarse approximation of many real-life processes



Intensity of a Poisson process is constant:

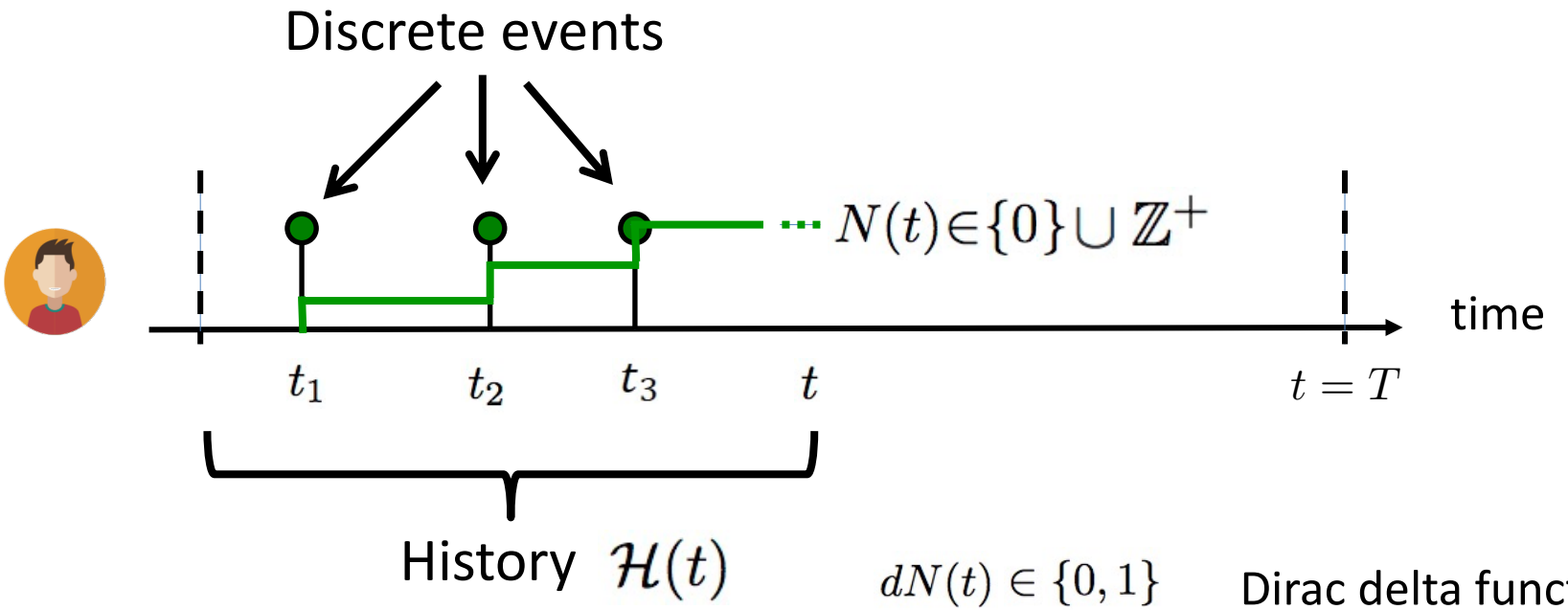
$$\lambda^*(t) = \mu$$

Observations:

1. Intensity independent of history
2. Uniformly random occurrence
3. Time between events follows exponential distribution

Defintion: Temporal point processes

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

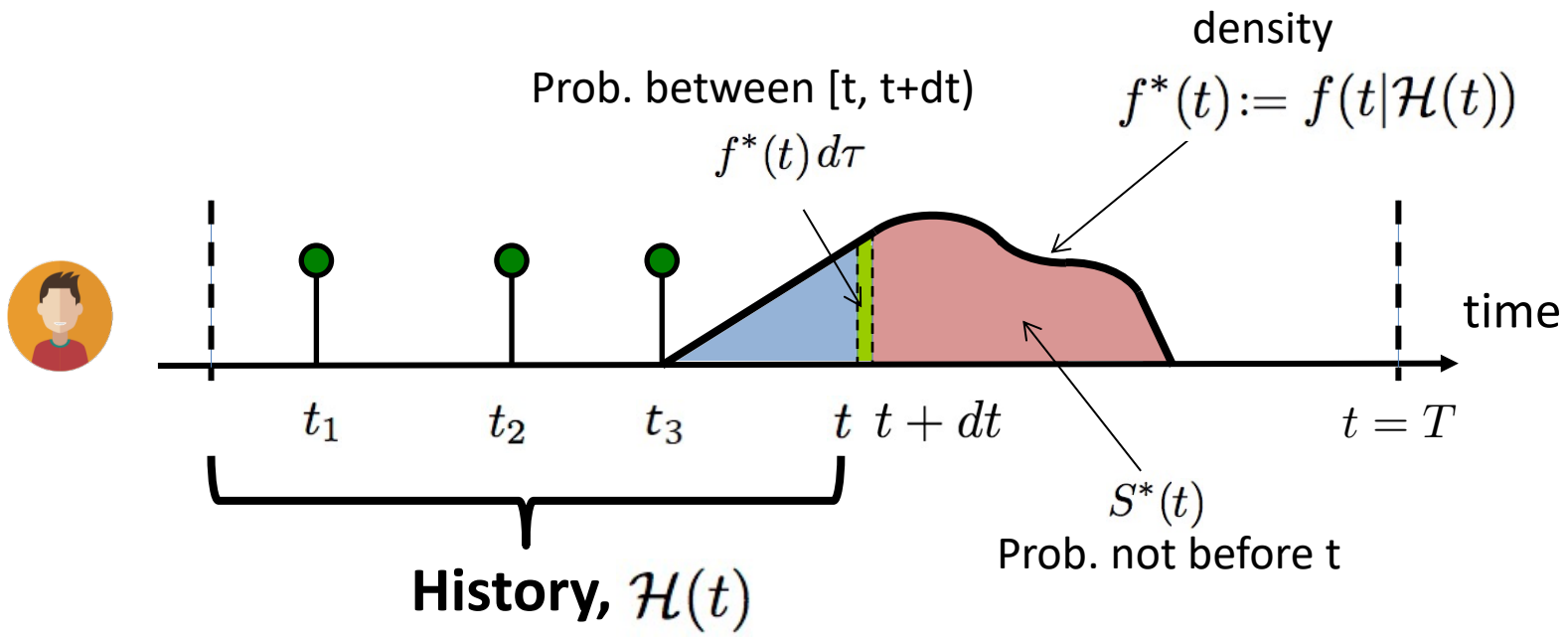


$dN(t) \in \{0, 1\}$

Dirac delta function

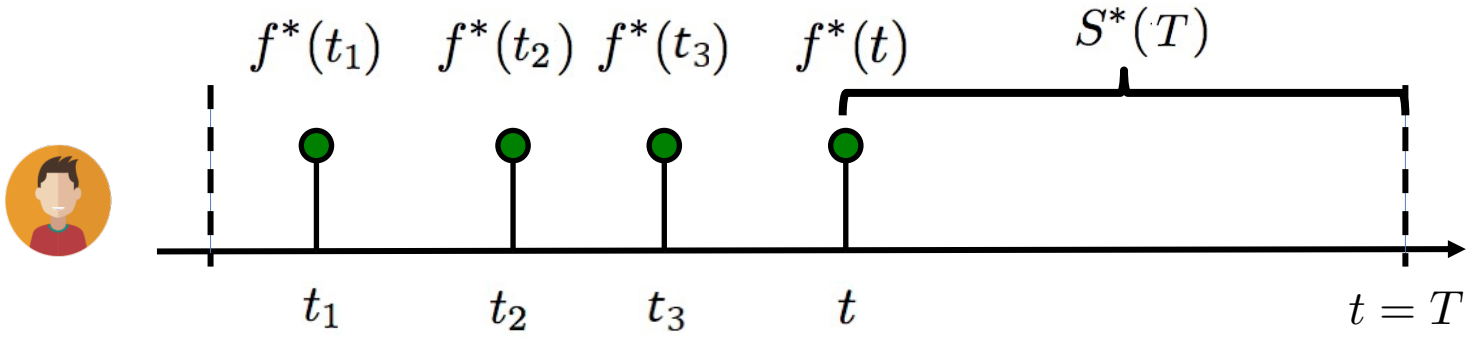
Formally: $N(t) = \int_0^t dN(s)$ ⇒ $dN(t) = \sum_{t_i \in \mathcal{H}} \delta(t - t_i)\, dt$

Distribution we are looking for the next event time



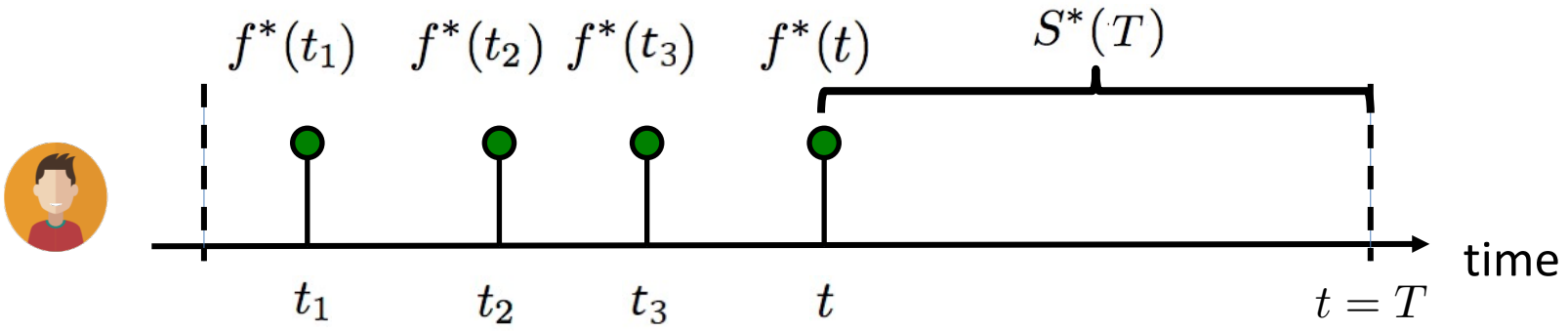
Skoltech

We can write and optimize a likelihood function



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

Density parametrization is hard



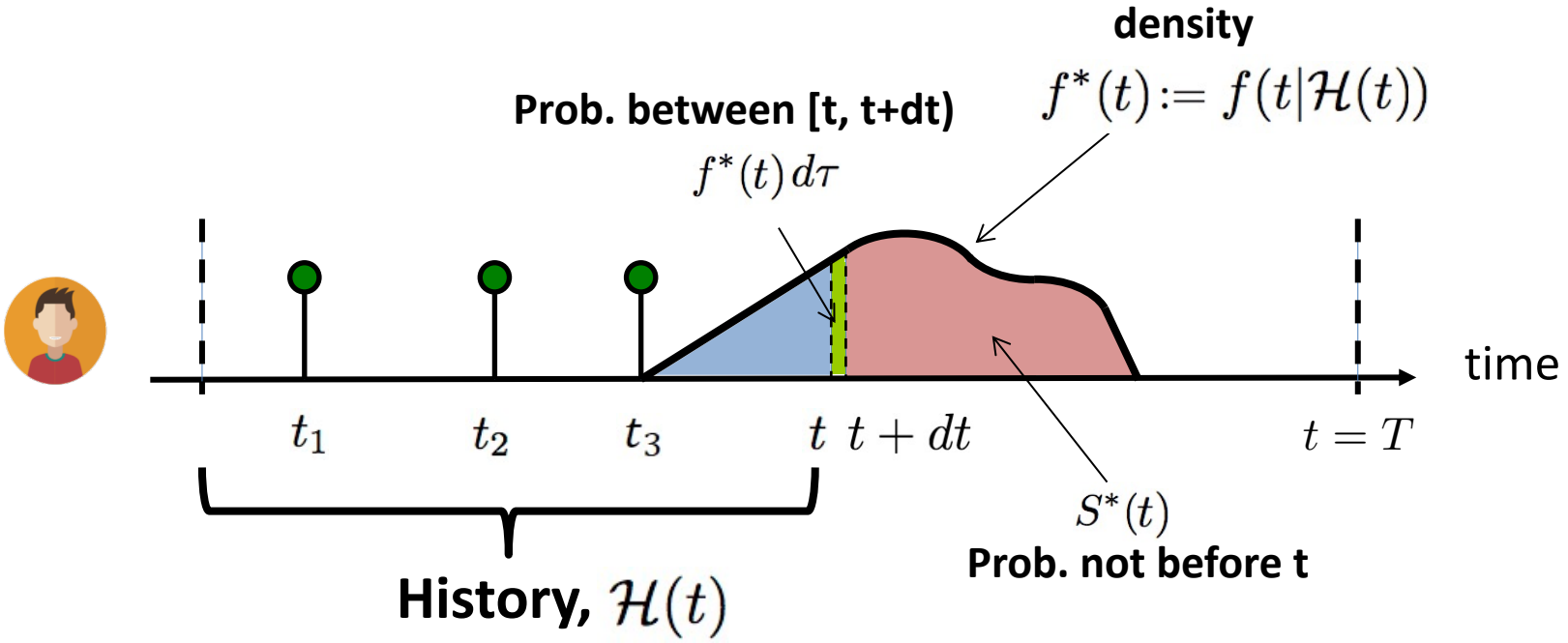
$$f^*(t_1)\ f^*(t_2)\ f^*(t_3)\ f^*(t)\ S^*(T)$$

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

It is <u>difficult for model design and interpretability</u>:

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

Intensity function is an alternative



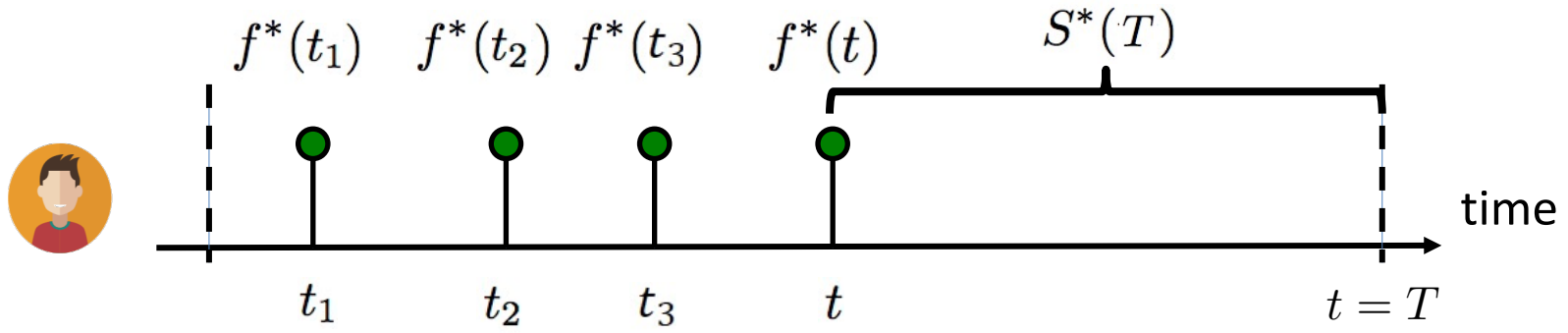
Intensity:

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

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

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

Likelihood for intensity



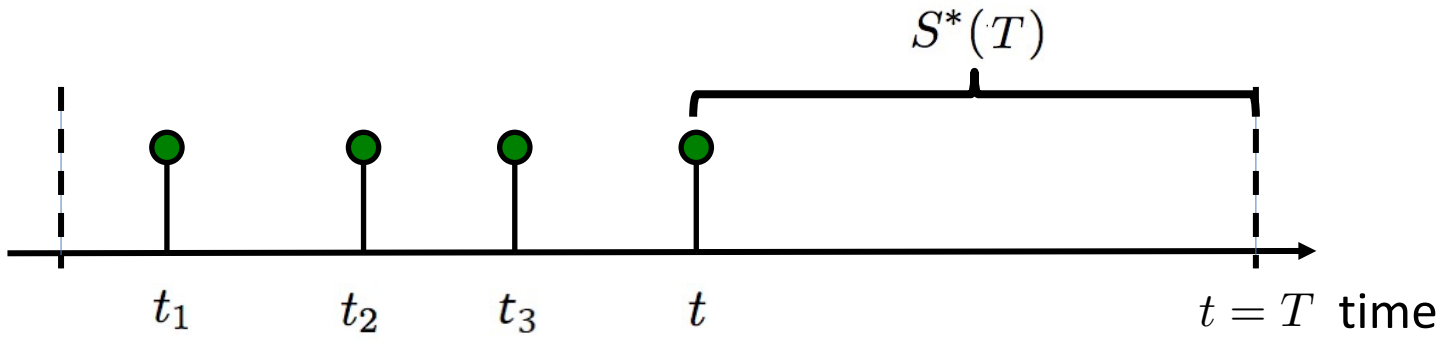
$$\lambda^*(t_1)\,\lambda^*(t_2)\,\lambda^*(t_3)\,\lambda^*(t)\,\exp\left(-\int_0^T \lambda^*(\tau)\,d\tau\right)$$

$\langle w, \phi^*(t_1)\rangle$ $\langle w, \phi^*(t_2)\rangle$ $\langle w, \phi^*(t_3)\rangle$ $\langle w, \phi^*(t)\rangle$ $\exp\left(-\int_0^T \langle w, \phi^*(\tau)\rangle d\tau\right)$

Suitable for model design and interpretable:

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

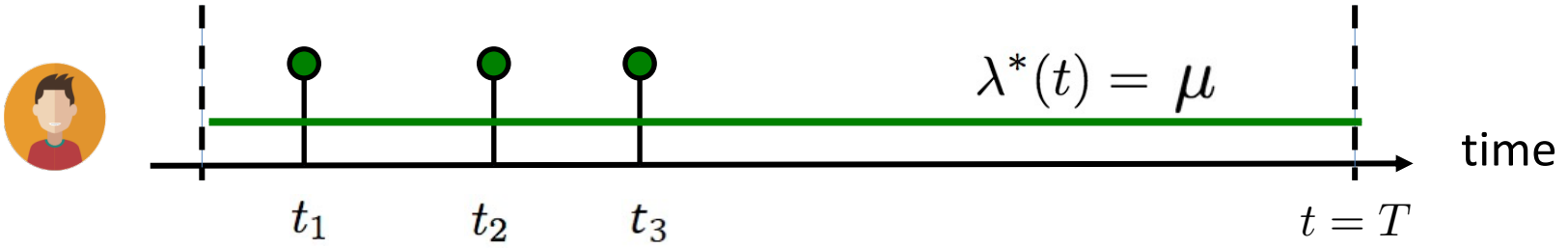
Log likelihood via intensity and density



$$L = f(t_1|\mathcal{H}_0) f(t_2|\mathcal{H}_{t_1}) \cdots f(t_n|\mathcal{H}_{t_{n-1}})(1 - F(T|\mathcal{H}_{t_n}))$$

$$\begin{aligned}
L &= \left(\prod_{i=1}^{n} f(t_i|\mathcal{H}_{t_{i-1}}) \right) \frac{f(T|\mathcal{H}_{t_n})}{\lambda^*(T)} \\
&= \left(\prod_{i=1}^{n} \lambda^*(t_i) \exp\left(- \int_{t_{i-1}}^{t_i} \lambda^*(s) \mathrm{d}s \right) \right) \exp\left(- \int_{t_n}^{T} \lambda^*(s) \mathrm{d}s \right) \\
&= \left(\prod_{i=1}^{n} \lambda^*(t_i) \right) \exp\left(- \int_{0}^{T} \lambda^*(s) \mathrm{d}s \right),
\end{aligned}$$

Fitting & sampling for a Poisson process



Fitting by maximum likelihood:

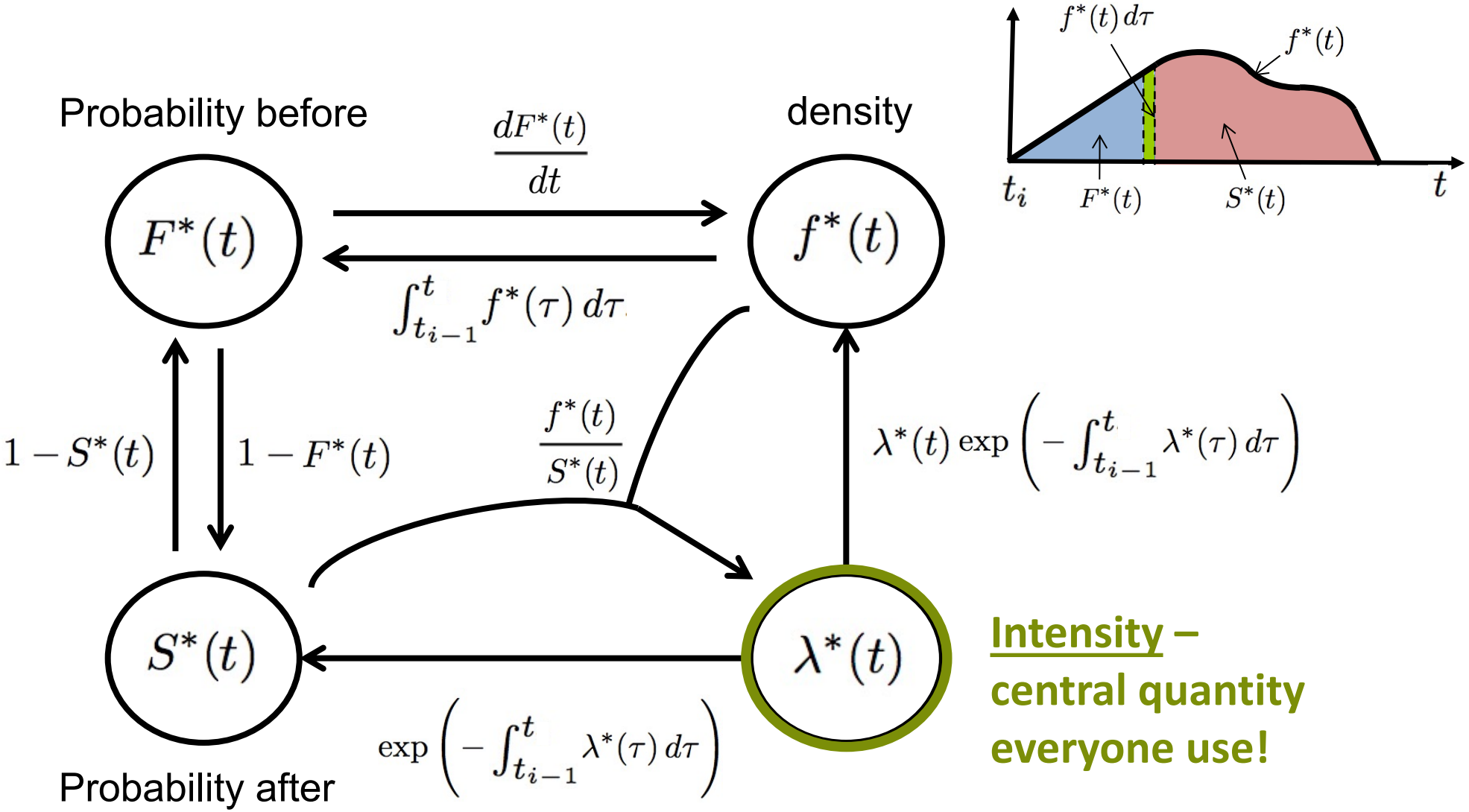
$$\mu^* = \underset{\mu}{\operatorname{argmax}} \; 3 \log \mu \; - \; \mu \, T \; = \; \frac{3}{T}$$

Sampling using inversion sampling:

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

$Uniform(0,1)$

Relation between f*, F*, S*, λ*



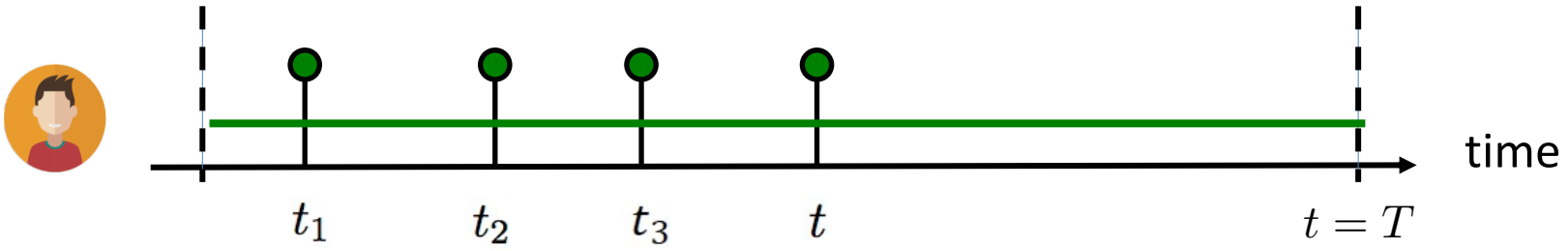
Intensity – central quantity everyone use!

Skoltech

Other models for temporal point processes

Poisson process

Coarse approximation of many real-life processes



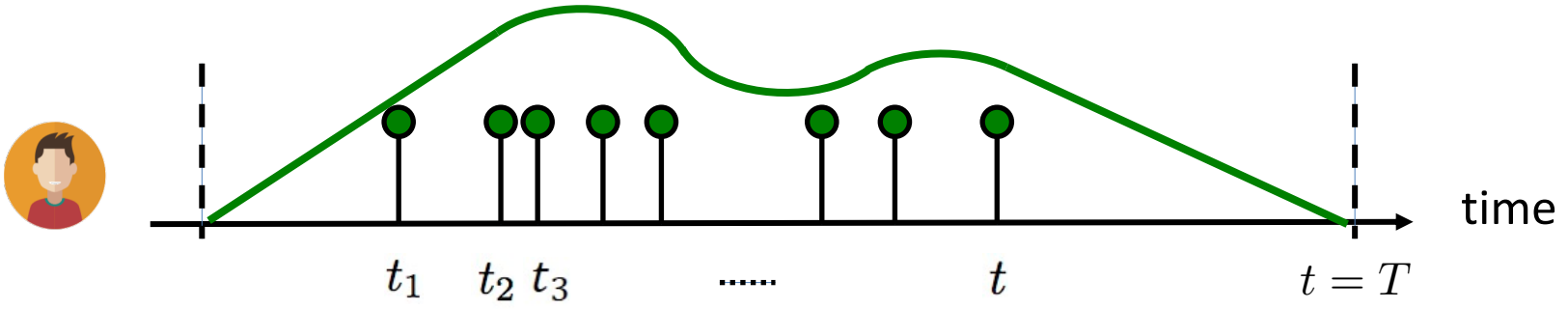
Intensity of a Poisson process

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

Observations:

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

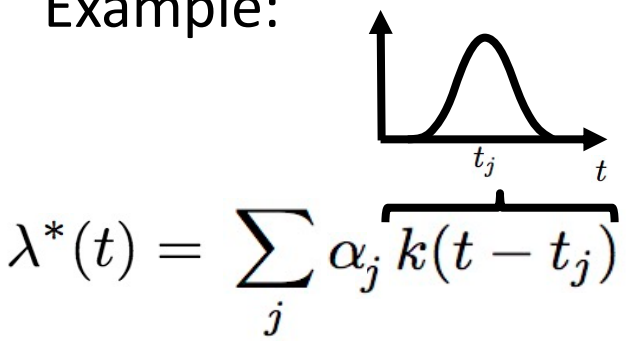
Inhomogeneous Poisson process

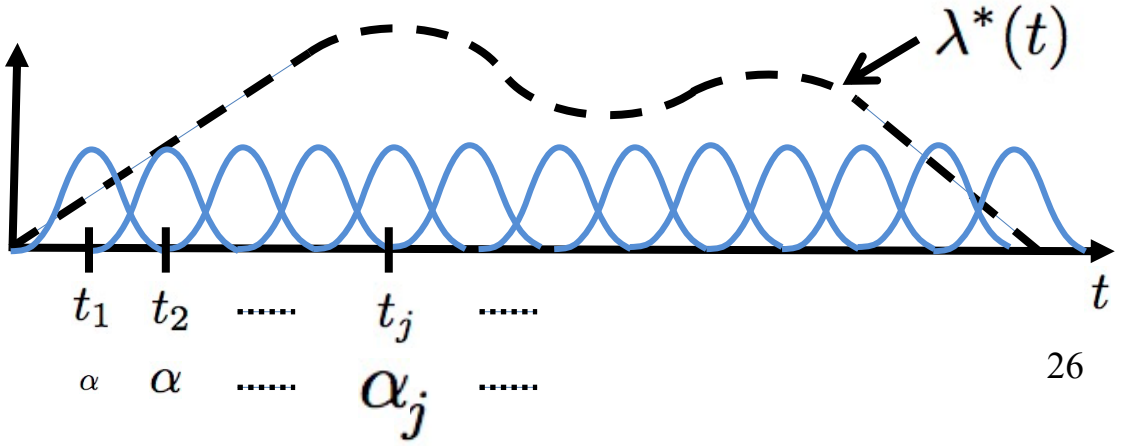


Intensity of an inhomogeneous Poisson process

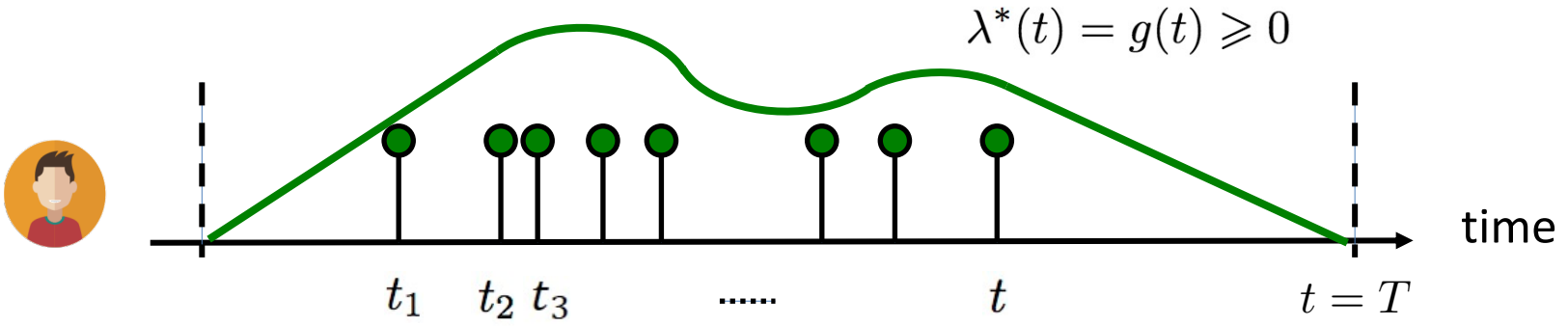
$$\lambda^*(t) = g(t) \geqslant 0$$ – Independent of history

Example:





Fitting from inhomogeneous Poisson



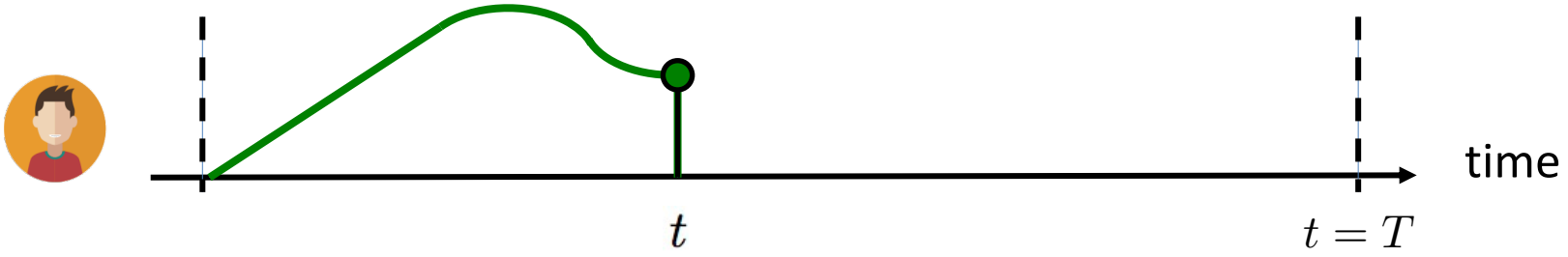
Fitting by maximum log-likelihood:

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

Idea: we have additional features, so we can use a generalized linear model for it

Intensity is $g(t) = g(\boldsymbol{x}_t) = \exp(\boldsymbol{x}_t^T w)$

Terminating (or survival) process



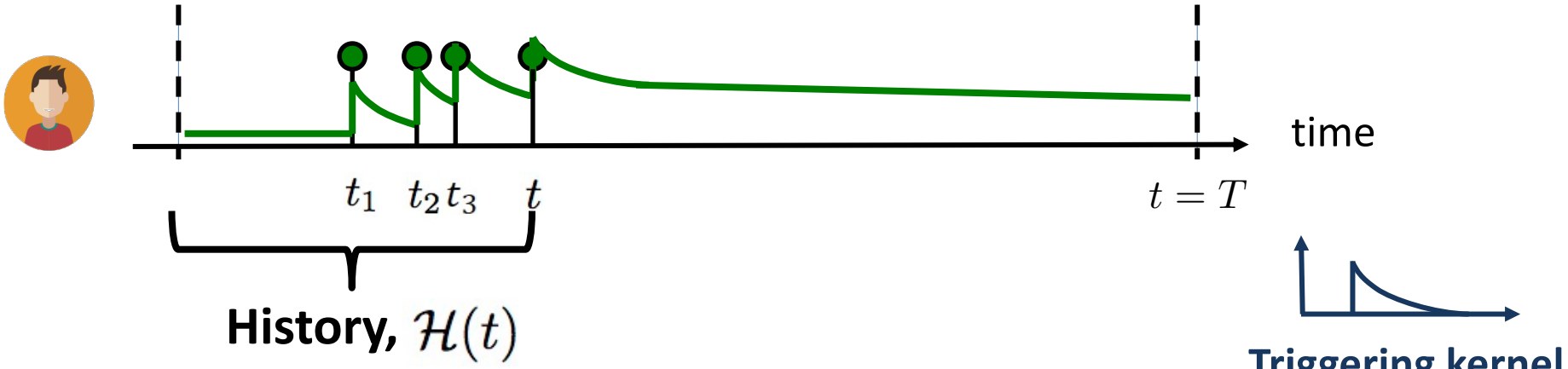
Intensity of a terminating (or survival) process

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

Observations:

1. Limited number of occurrences
2. Hazard function in actuarial science

Self-exciting Hawkes process



Intensity of self-exciting (or Hawkes) process:

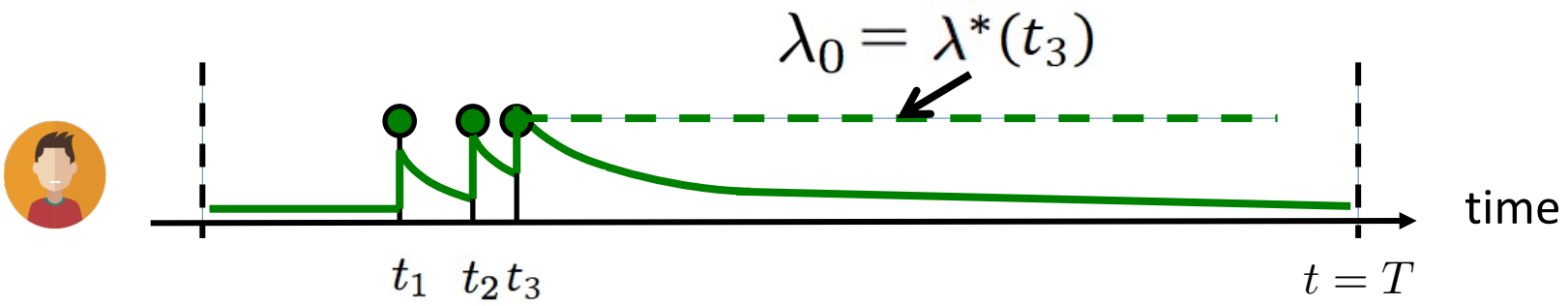
$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i)$$
$$= \mu + \alpha \kappa_\omega(t) \star dN(t)$$

Observations:

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

Hawkes, Alan G. "Spectra of some self-exciting and mutually exciting point processes." *Biometrika* 58.1 (1971): 83-90.

Fitting a Hawkes process from a recorded timeline



Fitting by maximum likelihood:

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

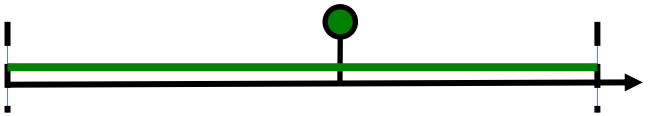
The max. likelihood is jointly convex in μ and α

Summary

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

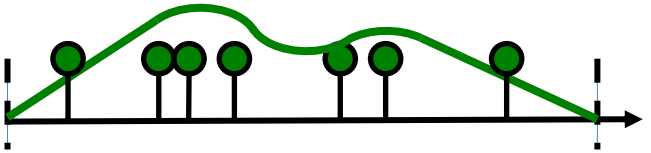
Poisson processes:

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



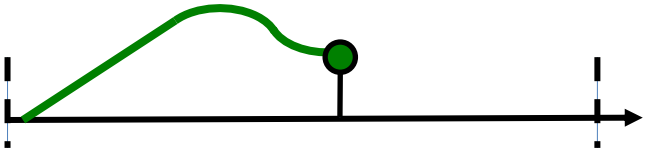
Inhomogeneous Poisson processes:

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



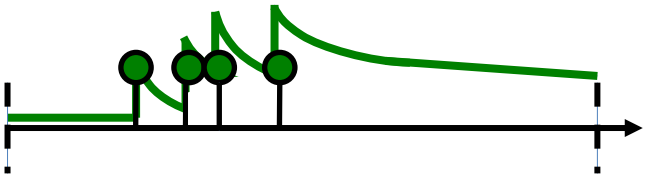
Terminating point processes:

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



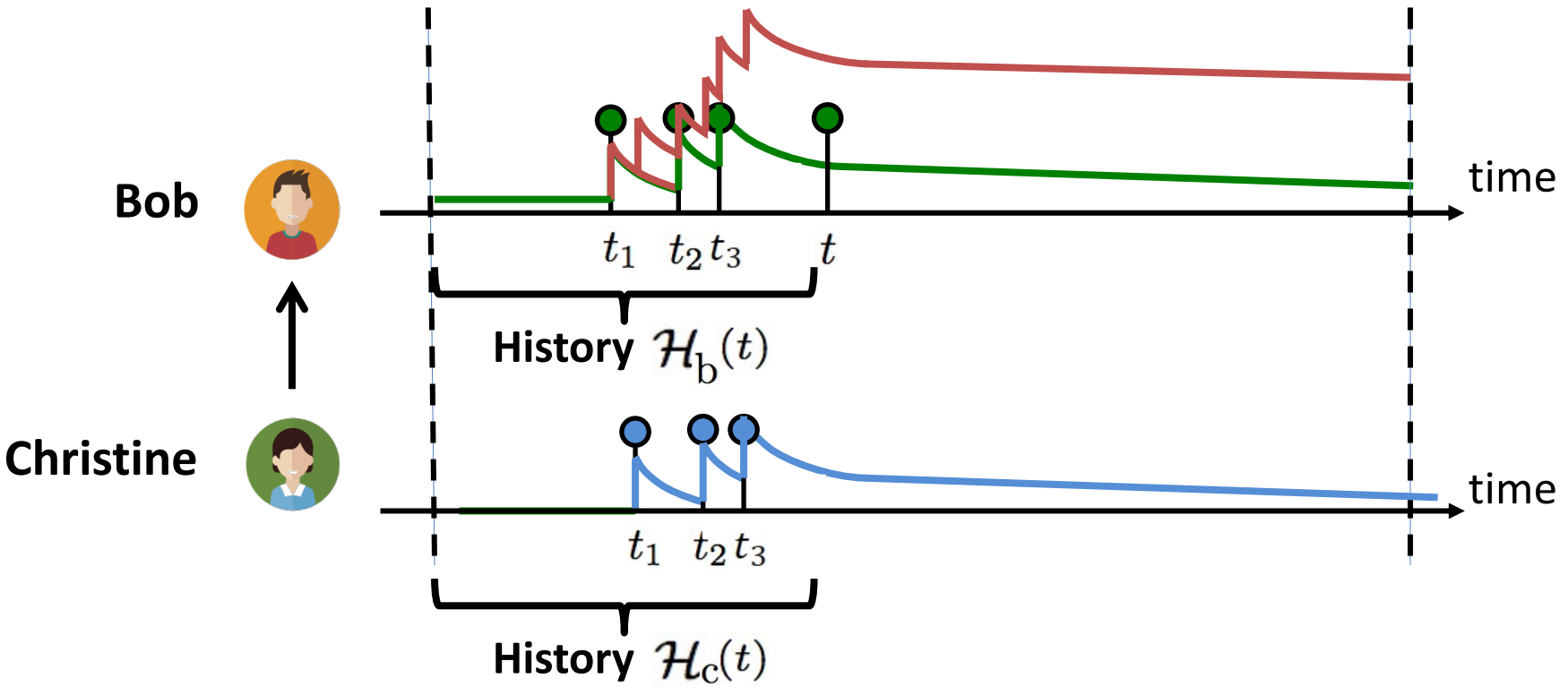
Self-exciting point processes:

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



Mutually excited point processes

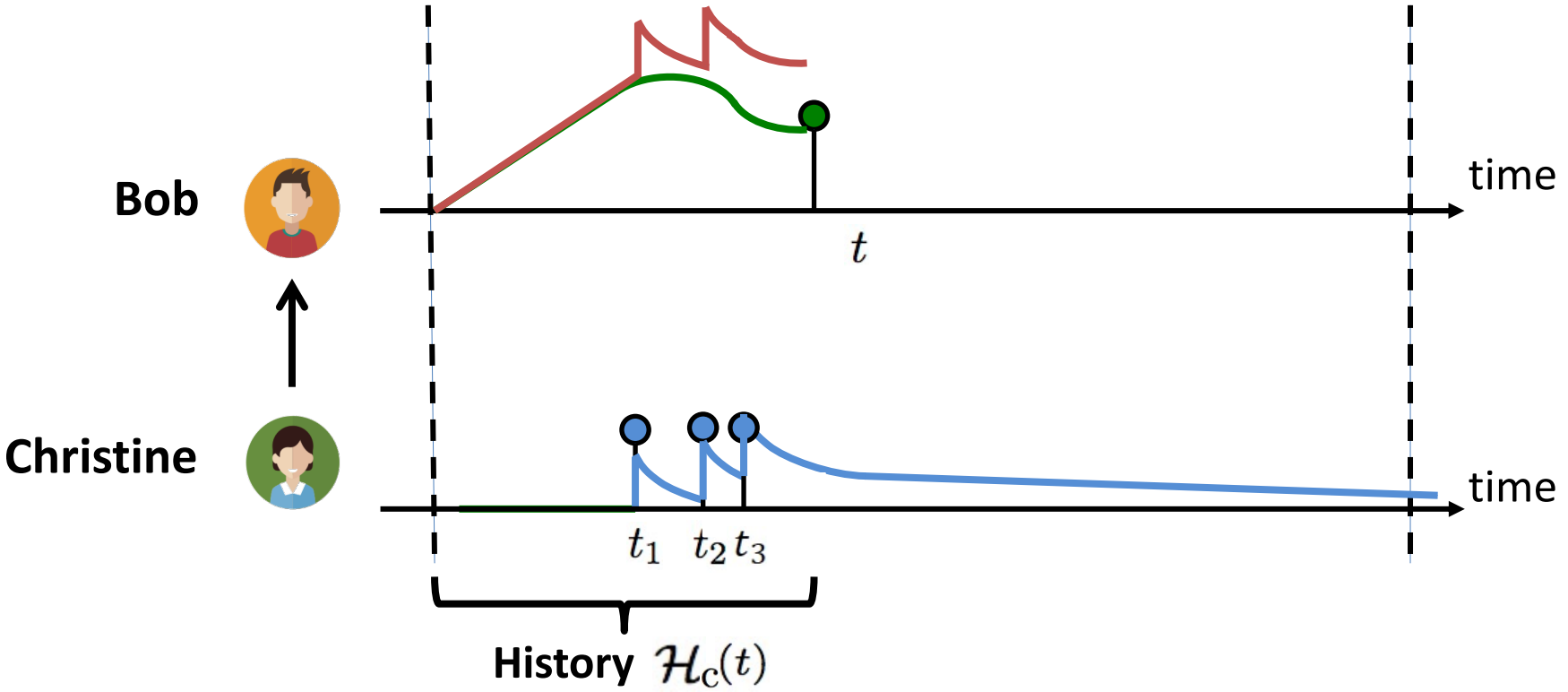
Mutually exciting process



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

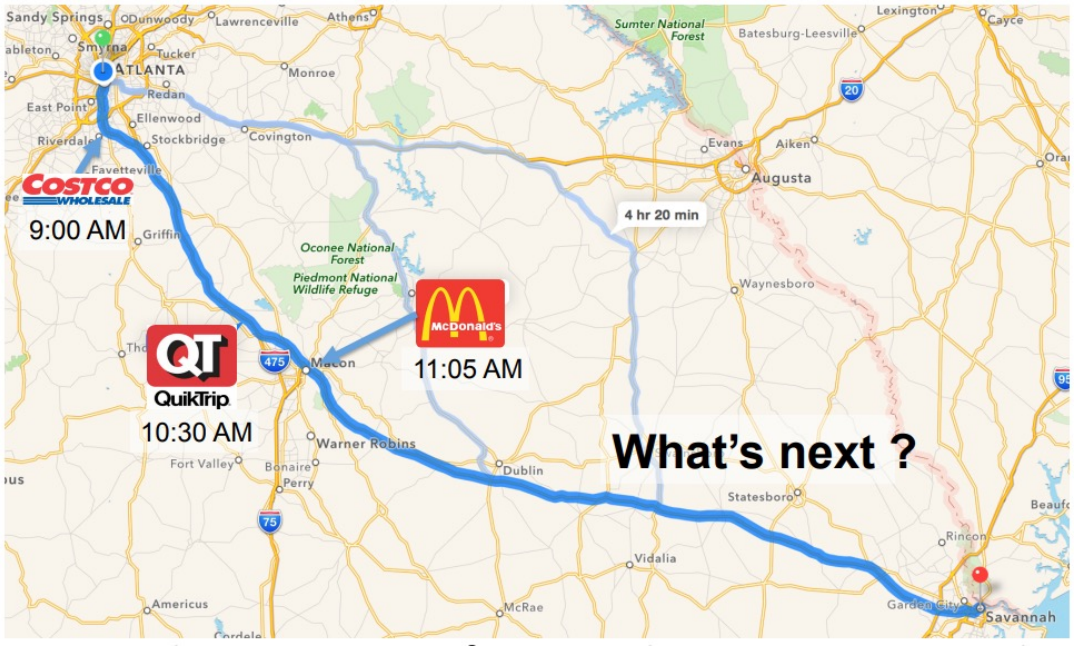
$$\lambda^*(t) = (1 - N(t))\Big(g(t) + \beta \sum_{t_i \in \mathcal{H}_c(t)} \kappa_\omega(t - t_i)\Big)$$

Marked temporal point processes

Marked temporal point processes

Marked temporal point process:

A random process whose realization consists of discrete *marked* events localized in time



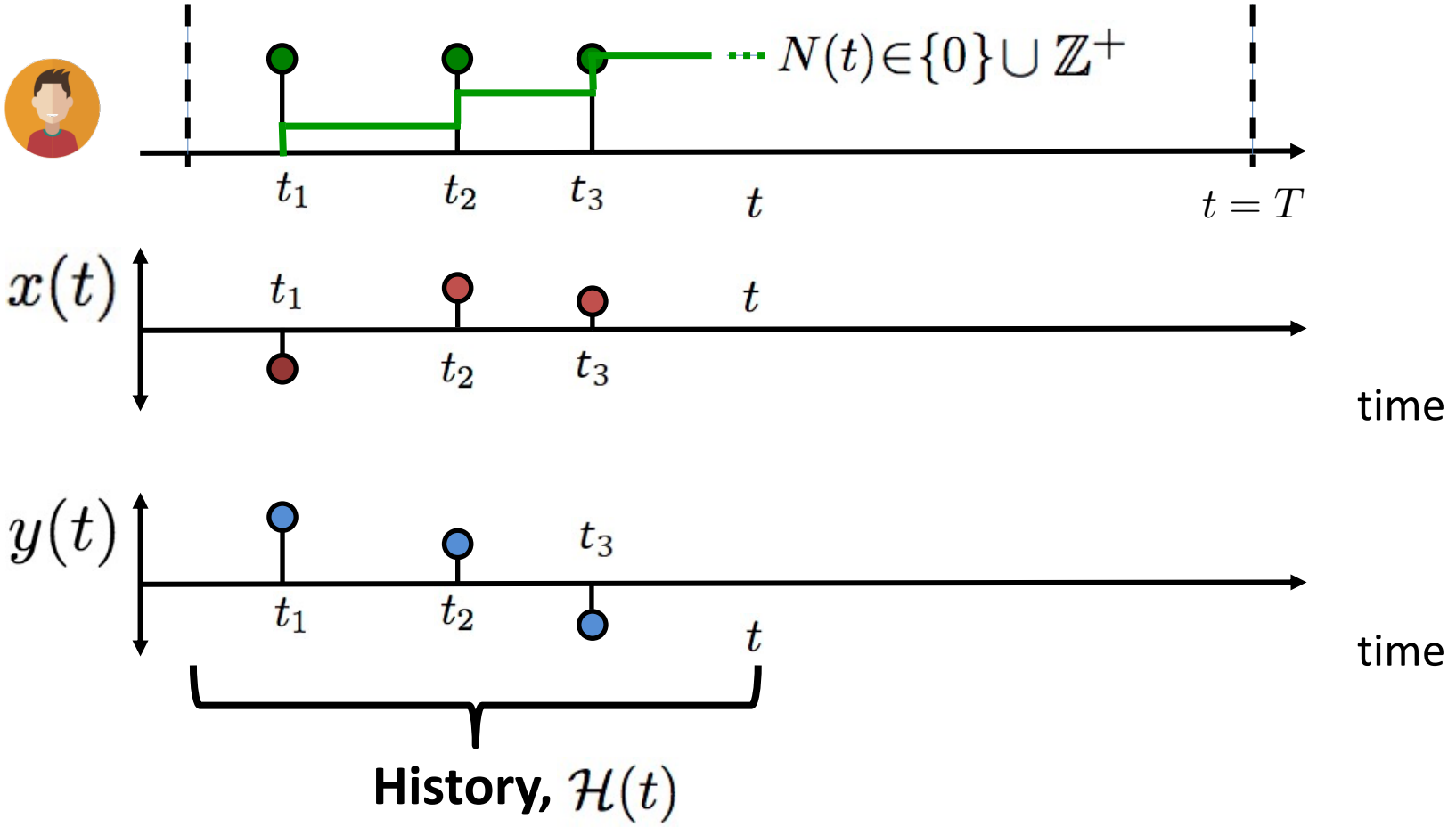
Given the trace of past locations and time, can we predict the location and time of the next stop?

Figure source: Du, Nan, et al. "Recurrent marked temporal point processes: Embedding event history to vector." KDD. 2016.

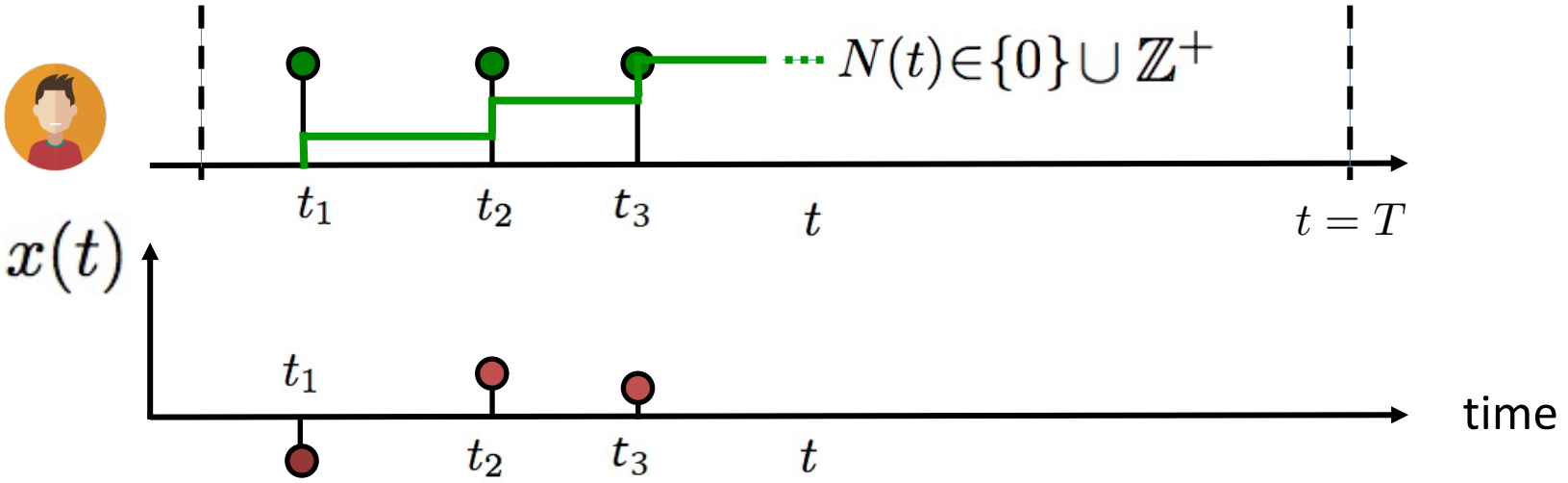
Marked temporal point processes

Marked temporal point process:

A random process whose realization consists of discrete *marked* events localized in time



Independent identically distributed marks



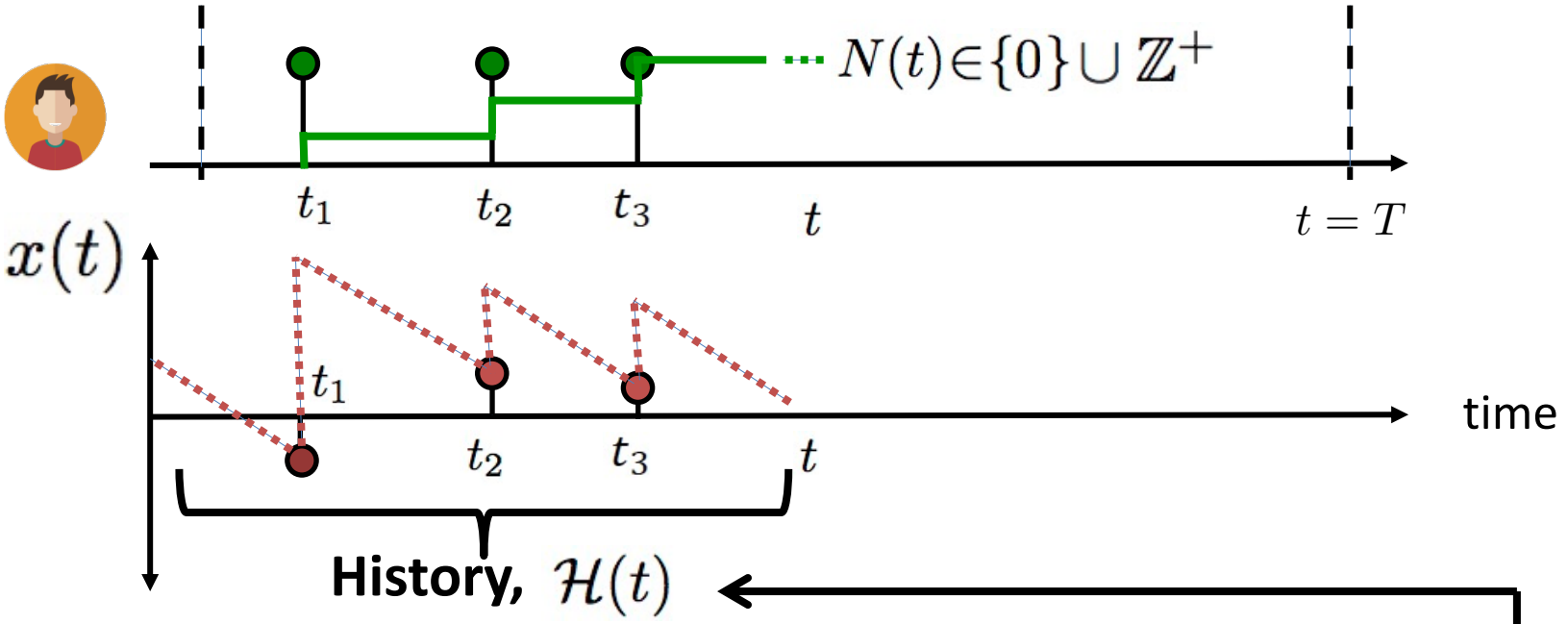
Distribution for the marks:

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

Observations:

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

Dependent marks: SDEs with jumps



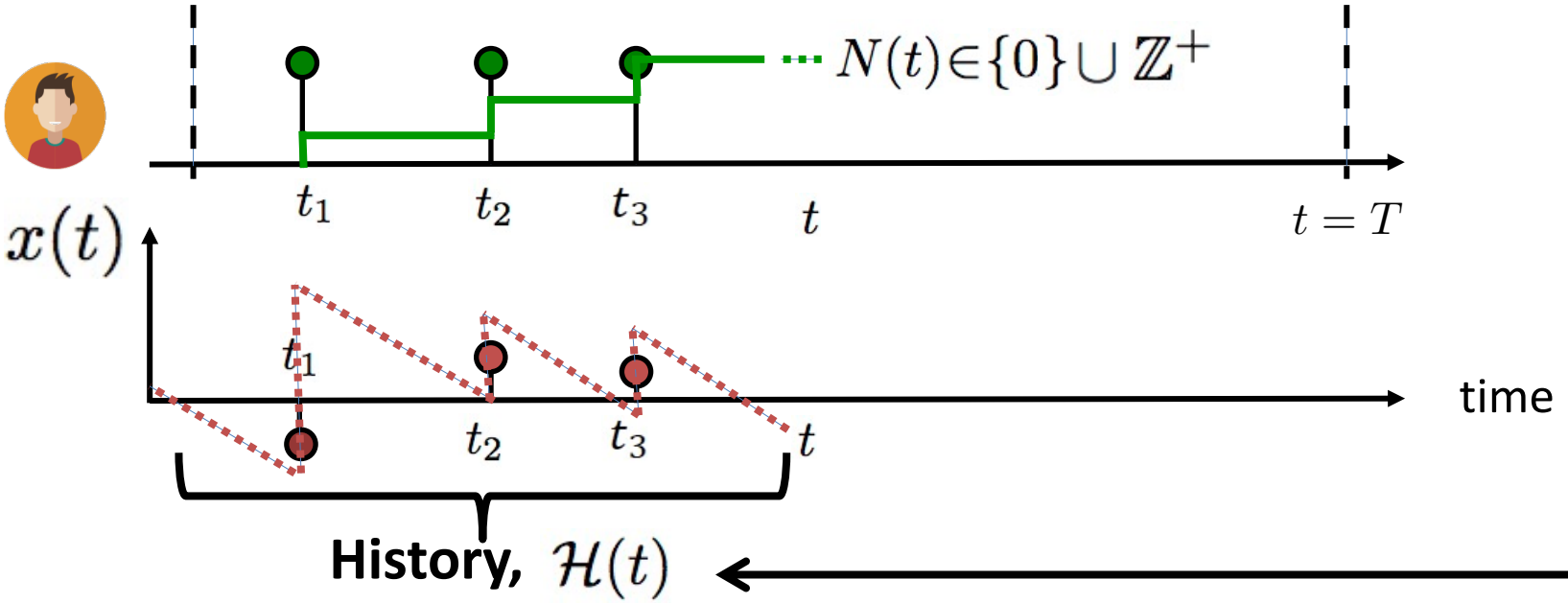
Marks given by stochastic differential equation with jumps:

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

Observations:

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

Dependent marks: distribution + SDE with jumps



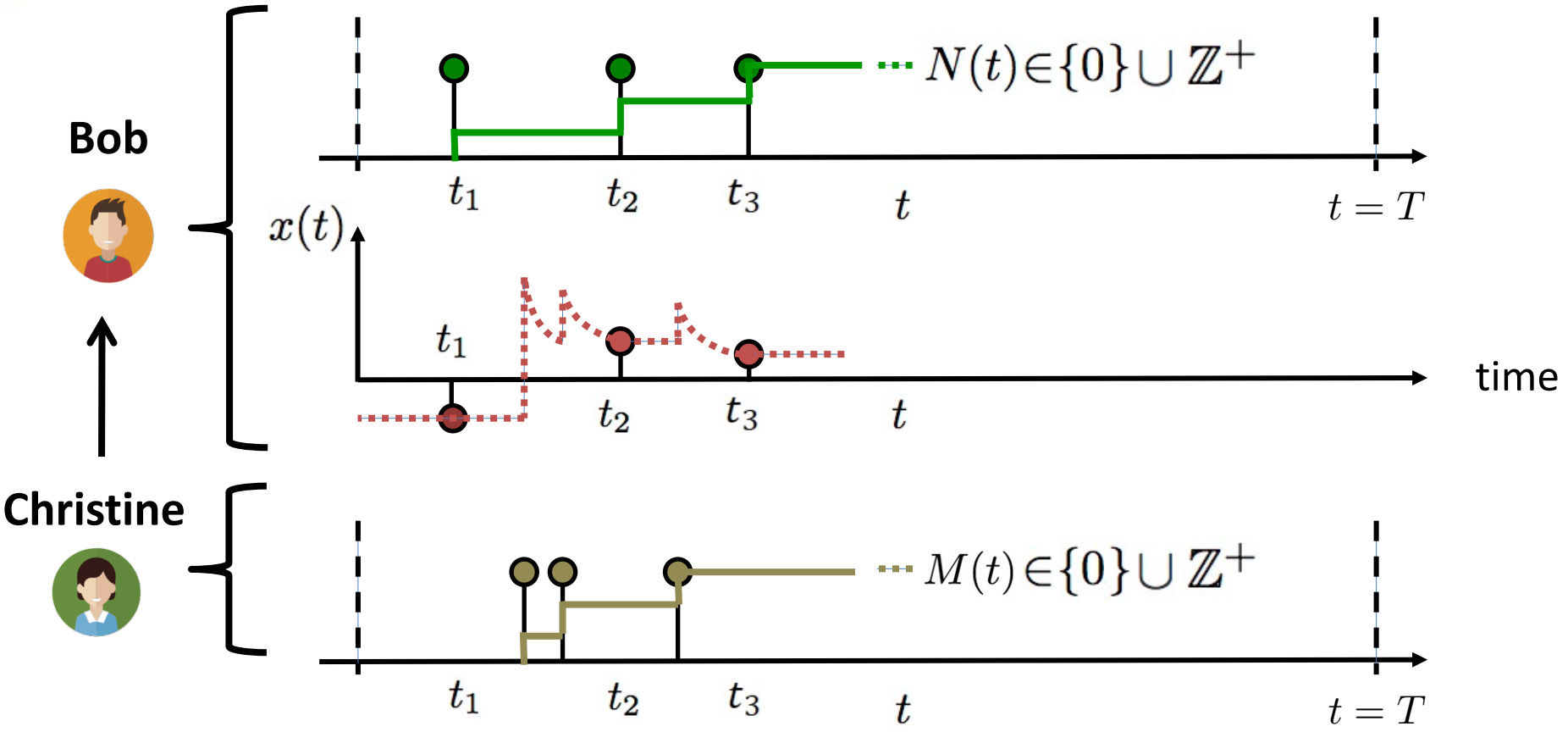
Distribution for the marks:

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

Observations:

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

Mutually exciting + marks



Marks affected by neighbors

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

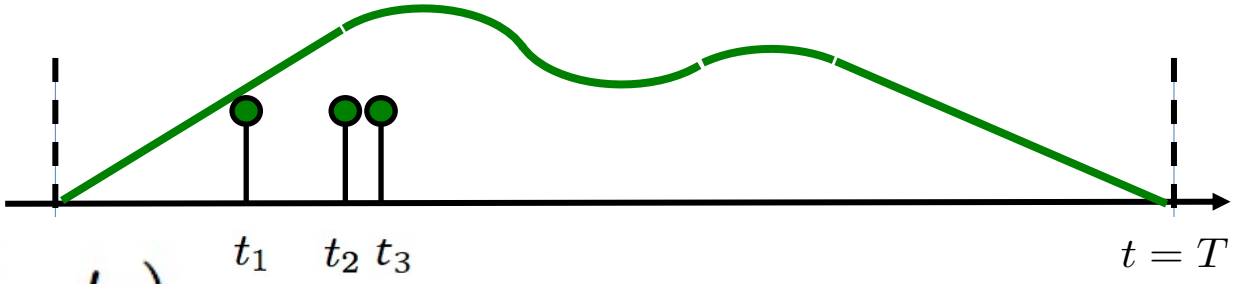
Models & Inference: Neural networks for the win

Skoltech

Hawkes process

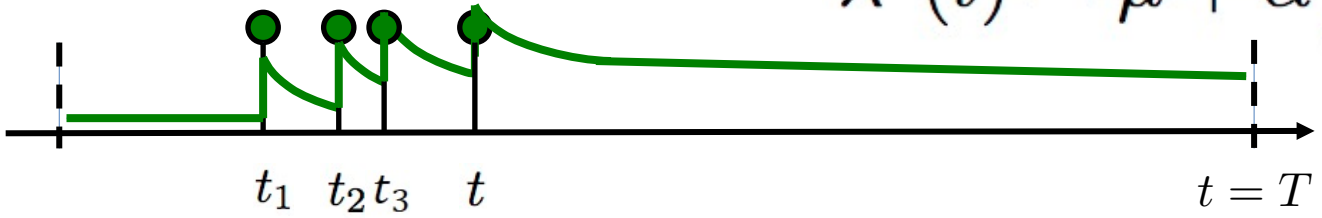
We focused on simple temporal dynamics and corresponding intensity functions:

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



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

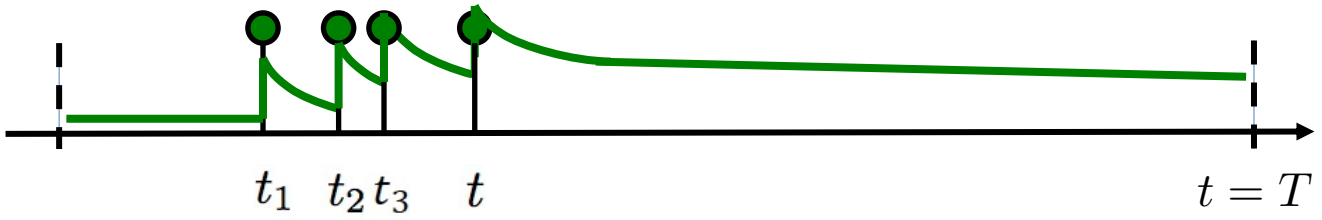
$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i)$$



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

Triggering Kernels for Hawkes point process

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



Triggering kernel is a linear sum of kernels

$$g_{uu'}(t) = \sum_{d=1}^{D} a_{uu'}^d g_d(t),$$

Loss function that penalizies non-smooth kernels

$$\mathcal{L}_\alpha(\Theta) = -\mathcal{L}(\Theta) + \alpha \left(\sum_d \mathcal{R}(g_d) + \sum_{u,u',d} (a_{uu'}^d)^2 \right).$$

Zhou, Ke, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional Hawkes processes. *ICML*, 2013.

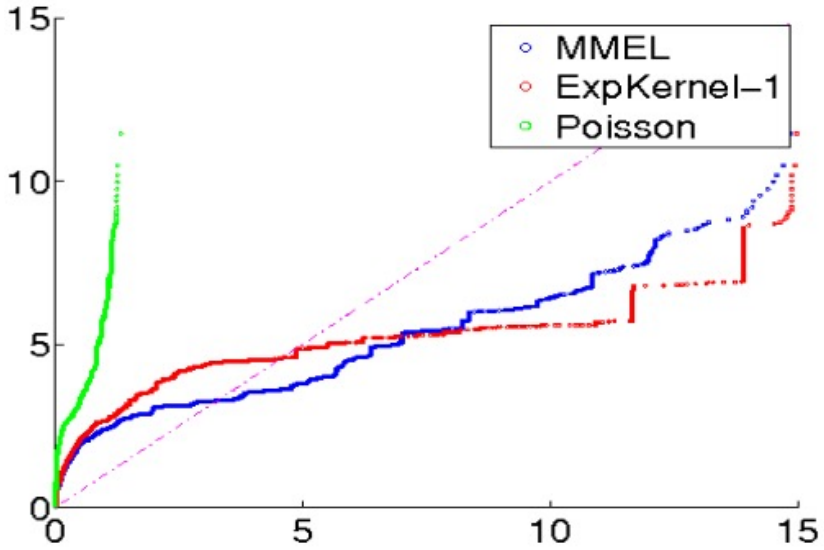
Optimization of kernels

We define $Q(\Theta; \Theta^{(k)})$ s.t.

1. For all Θ and $\Theta^{(k)}$, $Q(\Theta; \Theta^{(k)}) \geq \mathcal{L}_\alpha(\Theta)$.
2. $Q(\Theta^{(k)}; \Theta^{(k)}) = \mathcal{L}_\alpha(\Theta^{(k)})$.

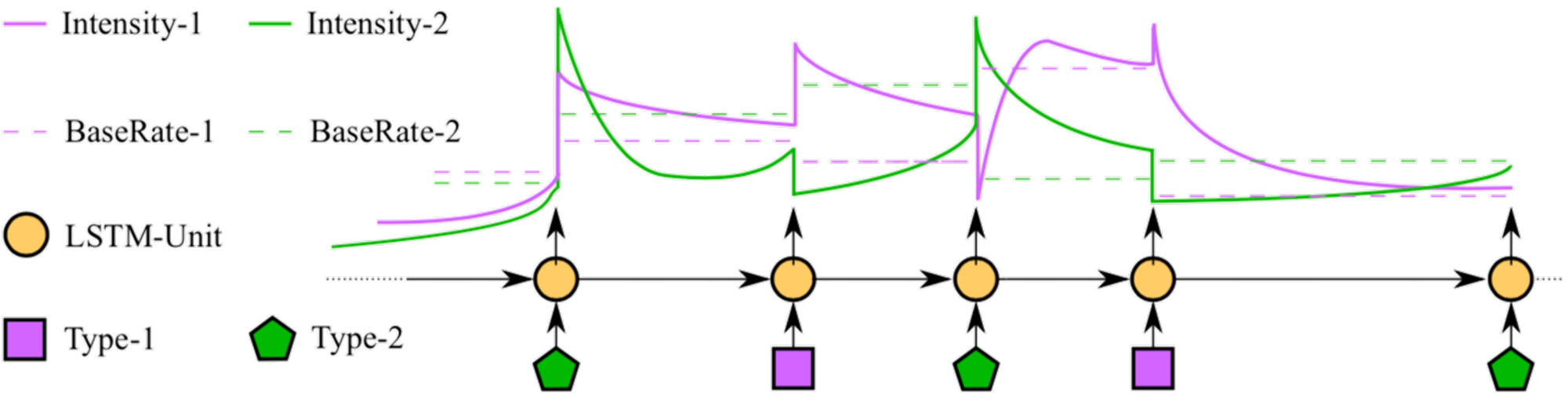
The above two properties imply that if $\Theta^{(k+1)} = \operatorname{argmin}_\Theta Q(\Theta; \Theta^{(k)})$, we have $\mathcal{L}_\alpha(\Theta^{(k)}) \geq \mathcal{L}_\alpha(\Theta^{(k+1)})$.

Separately update kernels and intensities + kernel weights.



Zhou, Ke, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional Hawkes processes. *ICML*, 2013.

Neural Hawkes process



Events affect the intensity of the process in a complex way

1) History effect does not need to be additive

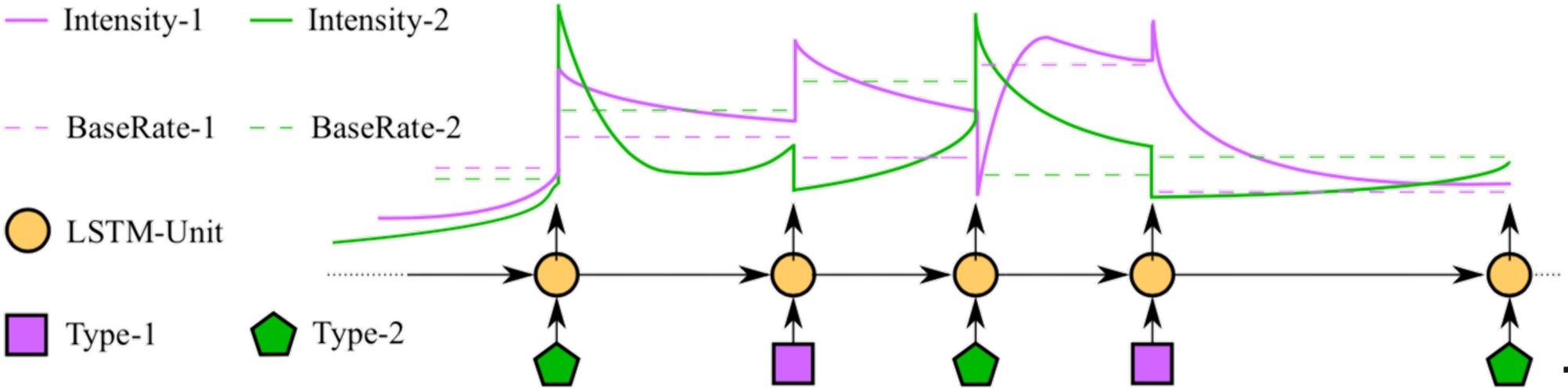
2) Allows for complex memory effects
e.g. delays

Mei, Hongyuan, and Jason M. Eisner. "The neural Hawkes process: A neurally self-modulating multivariate point process." *NeurIPS*. 2017.

Neural Hawkes process at event times

$$\mathbf{h}(t) = \mathrm{RNN}(\mathcal{H}(t))$$ memory via the continuous-time LSTM

$$\lambda_u(t) = f_u(\mathbf{w}_u^\top \mathbf{h}(t))$$ excitation & inhibition via activation function



Mei, Hongyuan, and Jason M. Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. *NeurIPS.* 2017.

Continuous intensity function for Neural Hawkes

Two cell states allow us to define continuous cell state:

$$\mathbf{c}(t) \stackrel{\text{def}}{=} \bar{\mathbf{c}}_{i+1} + (\mathbf{c}_{i+1} - \bar{\mathbf{c}}_{i+1}) \exp\left(-\boldsymbol{\delta}_{i+1}\,(t - t_i)\right) \text{ for } t \in (t_i, t_{i+1}]$$

Continuous hidden state is a function of continuous cell state:

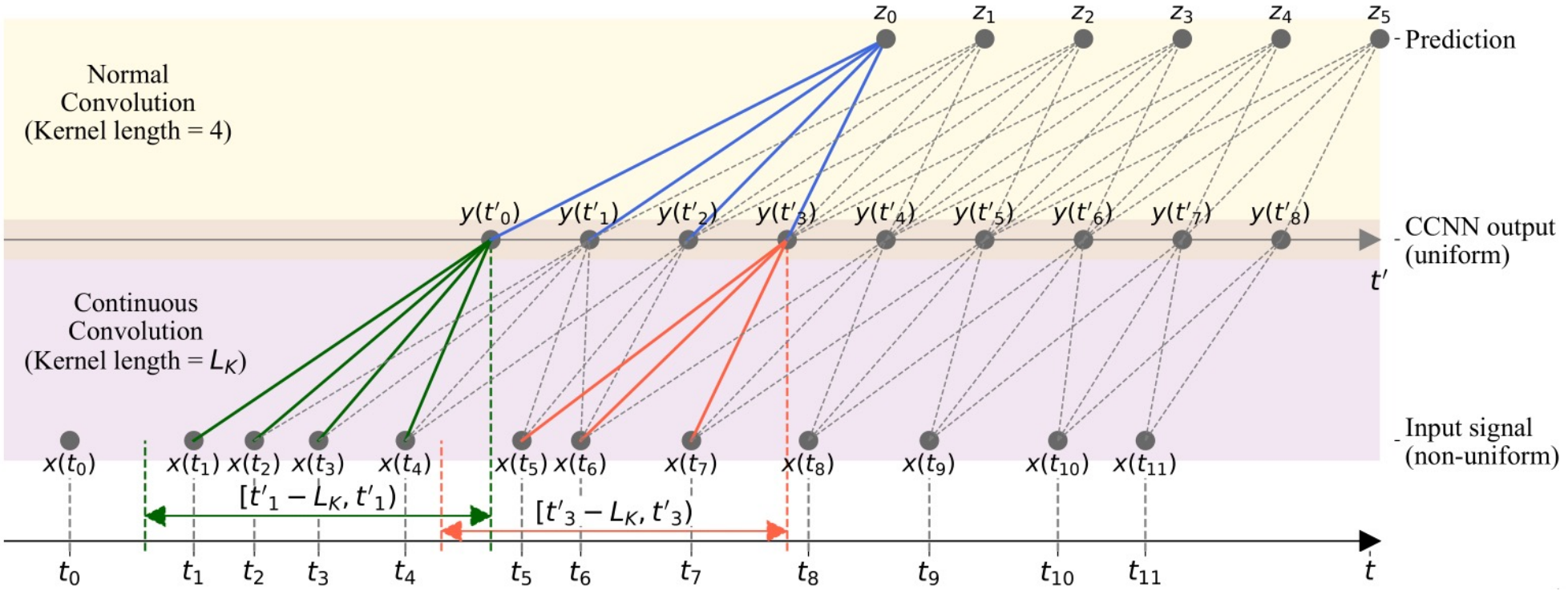
$$\mathbf{h}(t) = \mathbf{o}_i \odot (2\sigma(2\mathbf{c}(t)) - 1) \text{ for } t \in (t_{i-1}, t_i]$$

Now we can define the intensity function:

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

Mei, Hongyuan, and Jason M. Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. *NeurIPS.* 2017.

CNNs for point processes



After uniform resampling at the first layer we can apply normal convolutions

Shi, Hui, et al. "Continuous CNN for nonuniform time series." *ICASSP*. IEEE, 2021.

Skoltech

Continuous time CNNs

The convolution is defined in the following way for input **x** and kernel **phi**:

$$(\mathbf{x} * \boldsymbol{\psi})(t) = \sum_{c=1}^{N_{in}} \int_{\mathbb{R}} x_c(\tau)\psi_c(t-\tau)\,\mathrm{d}\tau.$$

Discretization of the convolution:

$$(\mathbf{x} * \boldsymbol{\psi})(t) = \sum_{c=1}^{N_{in}} \sum_{\tau=0}^{t} x_c(\tau)\psi_c(t-\tau).$$

The kernel is an MLP (Multi-layer perceptron). The activations have *sin* nonlinearities.

$$\mathbf{y} = \mathrm{Sine}(\omega_0[\mathbf{W}\mathbf{x} + \mathbf{b}]).$$

Romero, David W., et al. CKConv: Continuous Kernel Convolution For Sequential Data. *ICLR*. 2021.

Intensity function for continuous CNNs

We get intensity by adding state at the last event and weighted time since the last event:

$$\lambda^*(t) = \exp(\boldsymbol{v}\boldsymbol{h}_{i-1} + w(t - t_{i-1}) + b),$$

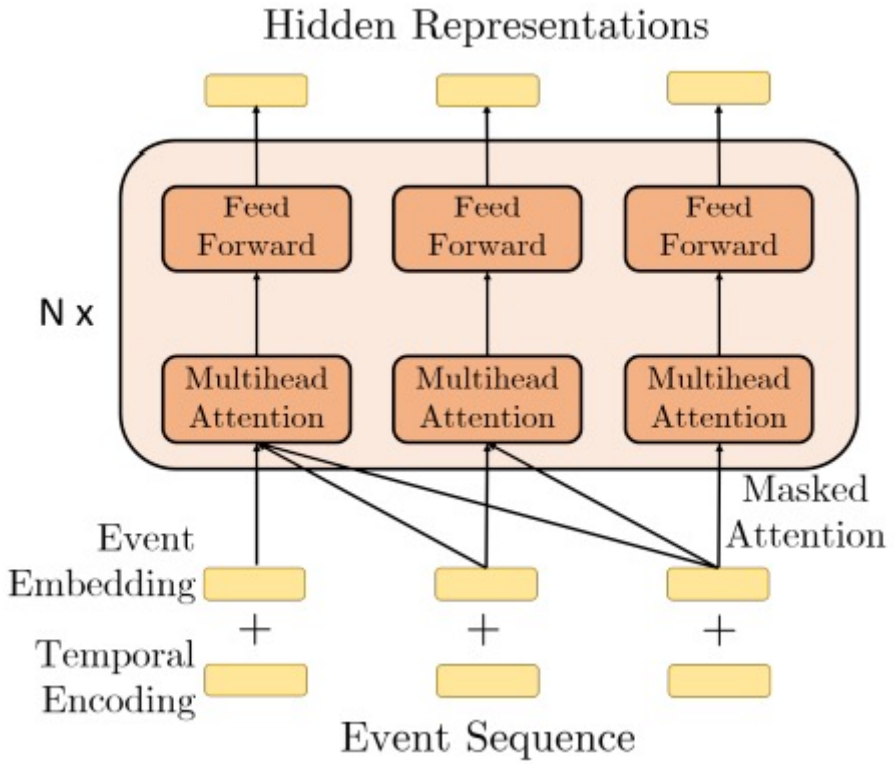
The closed-form expression for conditional intensity is

$$f^*(t) = \exp\left(\boldsymbol{v}\boldsymbol{h}_{i-1} + w(t - t_{i-1}) + b + \frac{1}{w}\Big(\exp(\boldsymbol{v}\boldsymbol{h}_{j-1} + b) - \exp(\boldsymbol{v}\boldsymbol{h}_{j-1} + w(t - t_{j-1}) + b)\Big)\right).$$

The kernel is an MLP (Multi-layer perceptron).

Shi, Hui, et al. "Continuous CNN for nonuniform time series." *ICASSP*. IEEE, 2021.

Transformers for Neural Hawkes process



We calculate densities and predict next event types:

$$p(t|\mathcal{H}_t) = \lambda(t|\mathcal{H}_t) \exp\Big(- \int_{t_j}^{t} \lambda(\tau|\mathcal{H}_\tau) d\tau \Big),$$

$$\widehat{t}_{j+1} = \int_{t_j}^{\infty} t \cdot p(t|\mathcal{H}_t) dt,$$

$$\widehat{k}_{j+1} = \operatorname*{argmax}_k \frac{\lambda_k(t_{j+1}|\mathcal{H}_{j+1})}{\lambda(t_{j+1}|\mathcal{H}_{j+1})}.$$

Similar idea to calculate intensity for k-th event type in the future:

$$\lambda_k(t|\mathcal{H}_t) = f_k\Big(\underbrace{\alpha_k \frac{t - t_j}{t_j}}_{\textit{current}} + \underbrace{\mathbf{w}_k^\top \mathbf{h}(t_j)}_{\textit{history}} + \underbrace{b_k}_{\textit{base}} \Big).$$

Zuo, Simiao, et al. Transformer Hawkes process. ICML. 2020.

Clustering of event sequences

NTPP-MIX framework for the clustering of event sequences

Suitable for any probabilistic model of a point process

EM algorithm with Mean-Field approximation to optimize the cluster parameters and cluster mixing.

After training, an event sequence is input into K NTPPs to get conditional probabilities.

Adding corresponding expectation of mixing coefficients, we get the approximated posterior of cluster assignment.

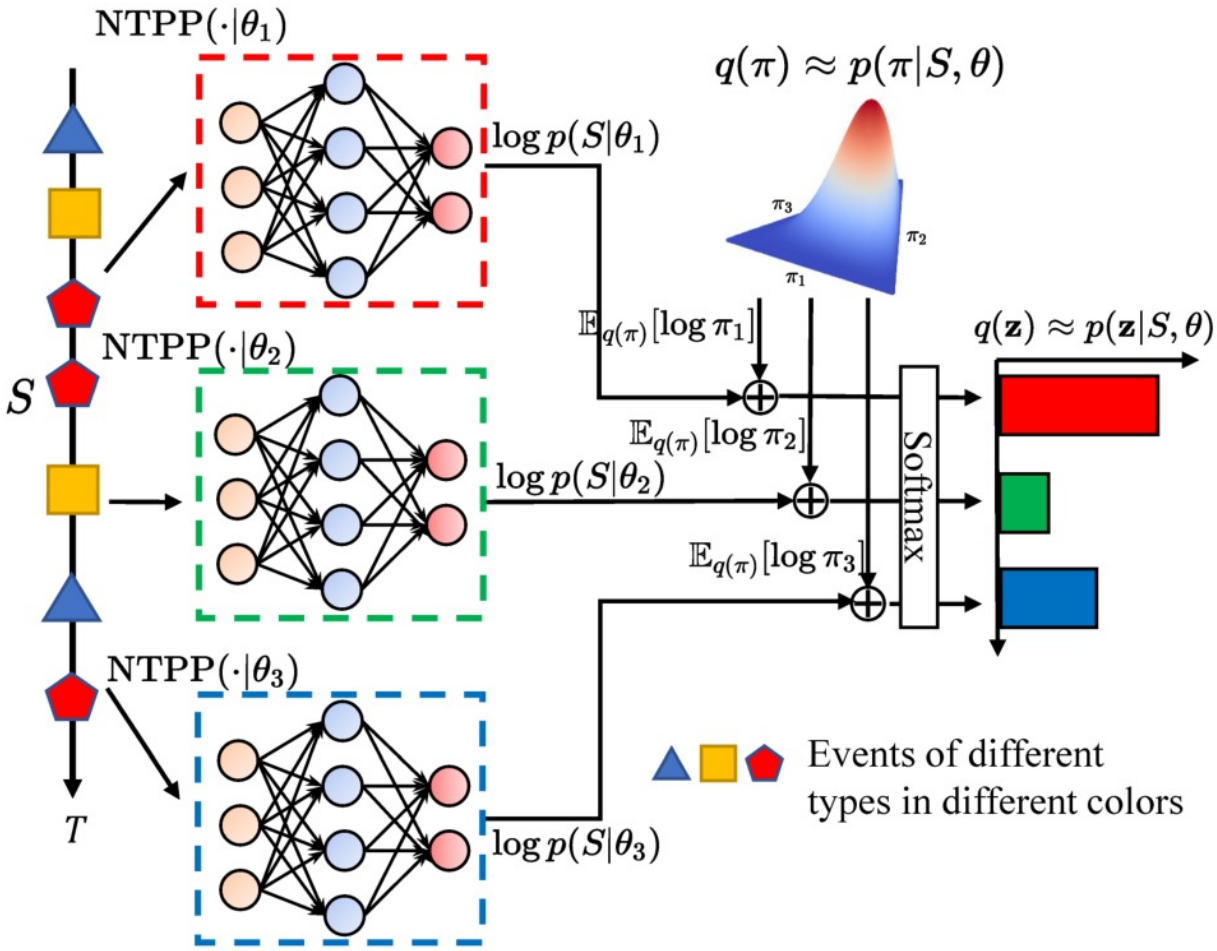
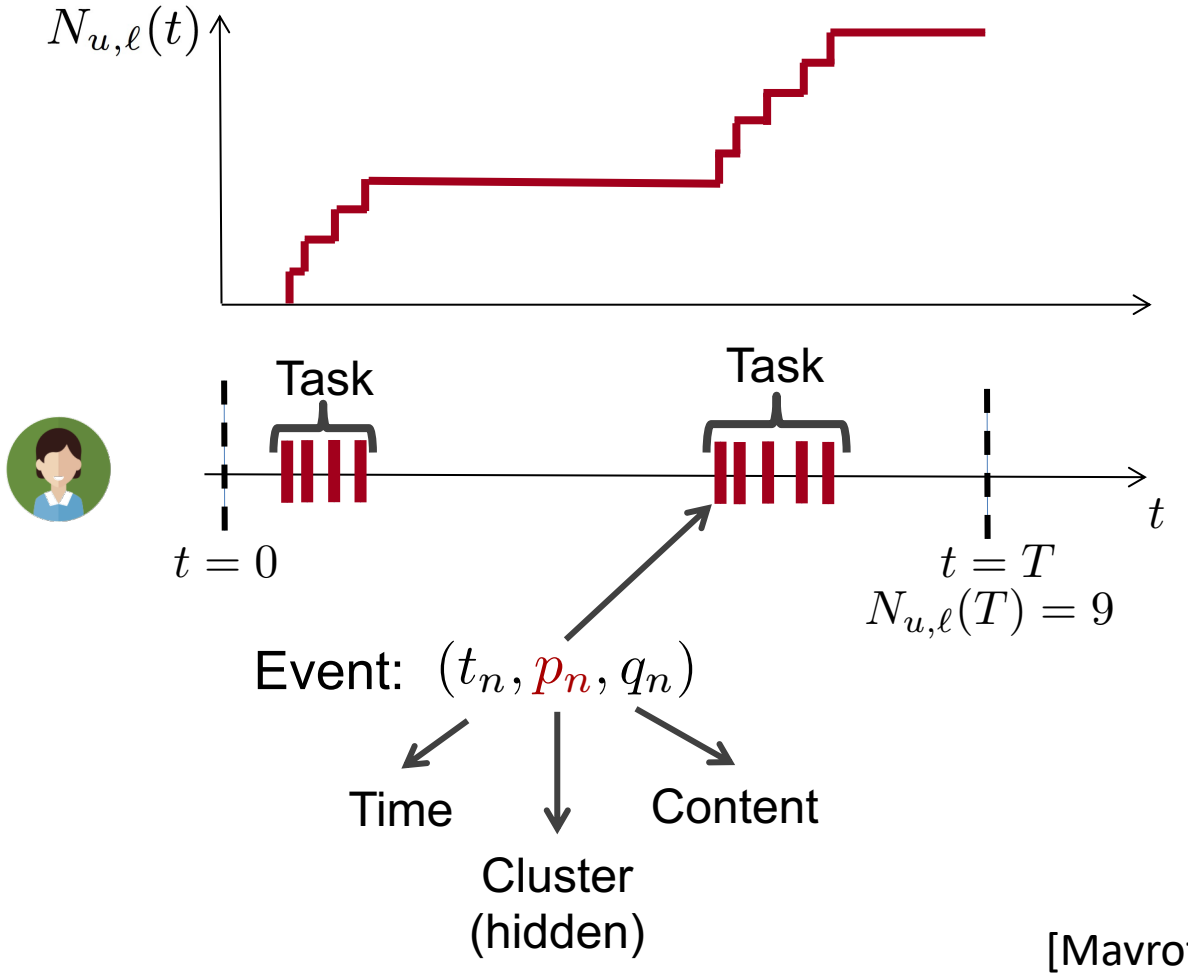


Figure: Inference, example for K = 3

Zhang, Yunhao, et al. Learning mixture of neural temporal point processes for multi-dimensional event sequence clustering. IJCAI, 2022

Events representation

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



[Mavroforakis et al., WWW 2017]

Conclusions

Conclusions

- Temporal point processes is a natural model for event sequences data

- Transformers help for incorporation of the whole history