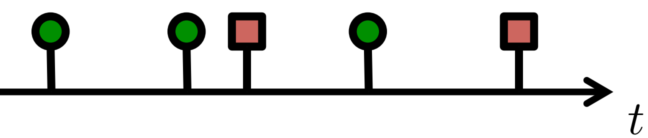
Introduction and Applications of **Temporal Point Processes**



Isabel Valera

MPI for Intelligent Systems

ETHZ Lecture, 14 December 2018

Outline of the Lecture

Intro to Temporal Point processes (TPPs)

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

Application: Clustering Event Sequences

1. Problem Statement
2. Introduction to DPMM
3. CRP + HP (a.k.a. HDHP)
4. Generative process

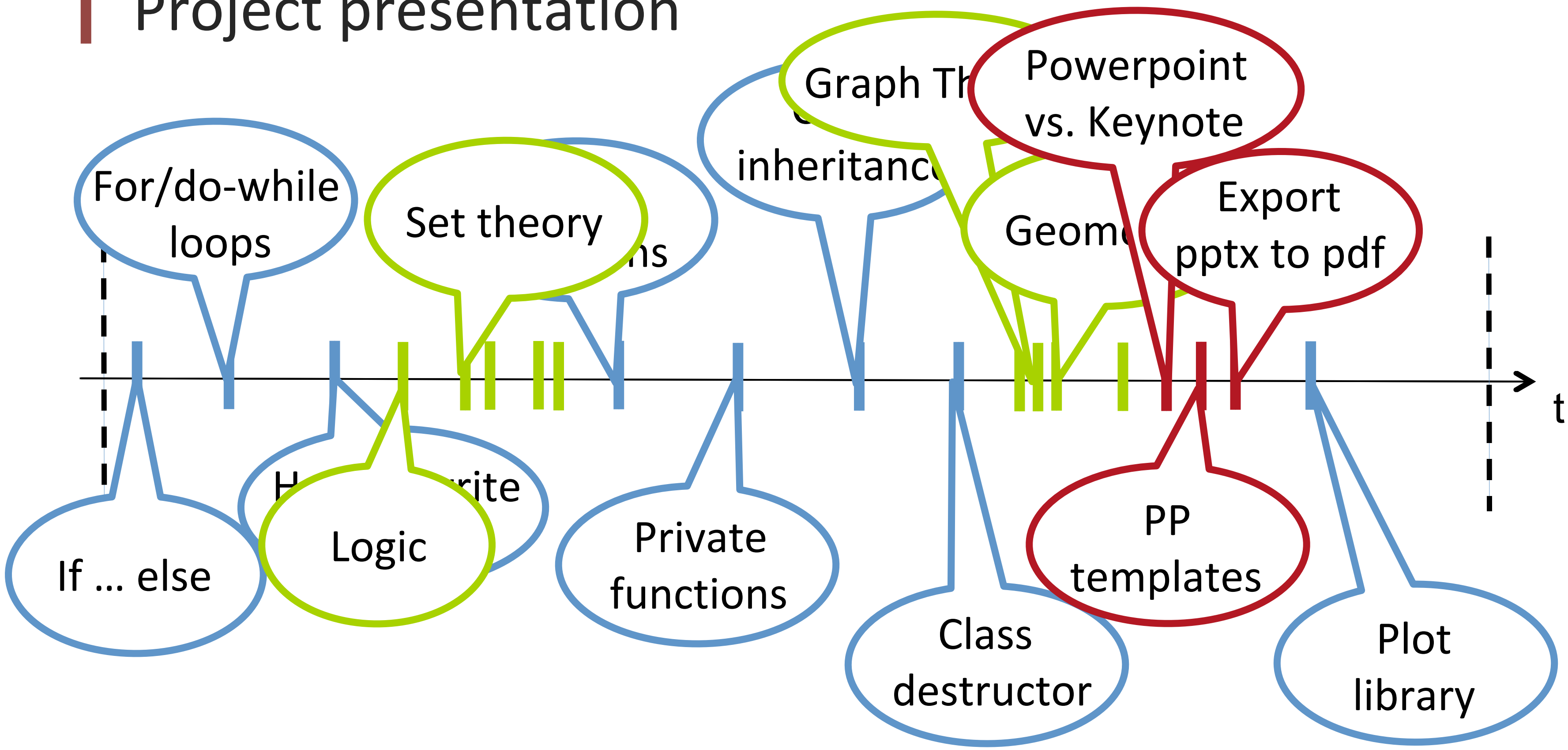
APPLICATION:
CLUSTERING EVENT SEQUENCES

1. **Problem Statement**
2. Introduction to DPs
3. CRP + HP (a.k.a. HDHP)
4. Generative Process

Clustering Event Sequences

1st year computer science student

- Introduction to programming
- Discrete math
- Project presentation



CLUSTERING EVENT SEQUENCES

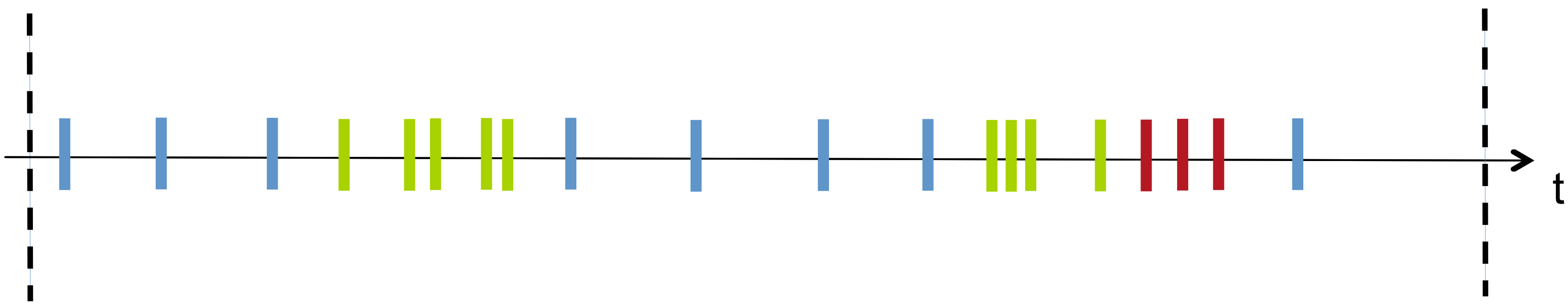
1st year computer science student

Content + Dynamics = Cluster (*learning pattern)*

E.g., **programing** + semester

math + semester

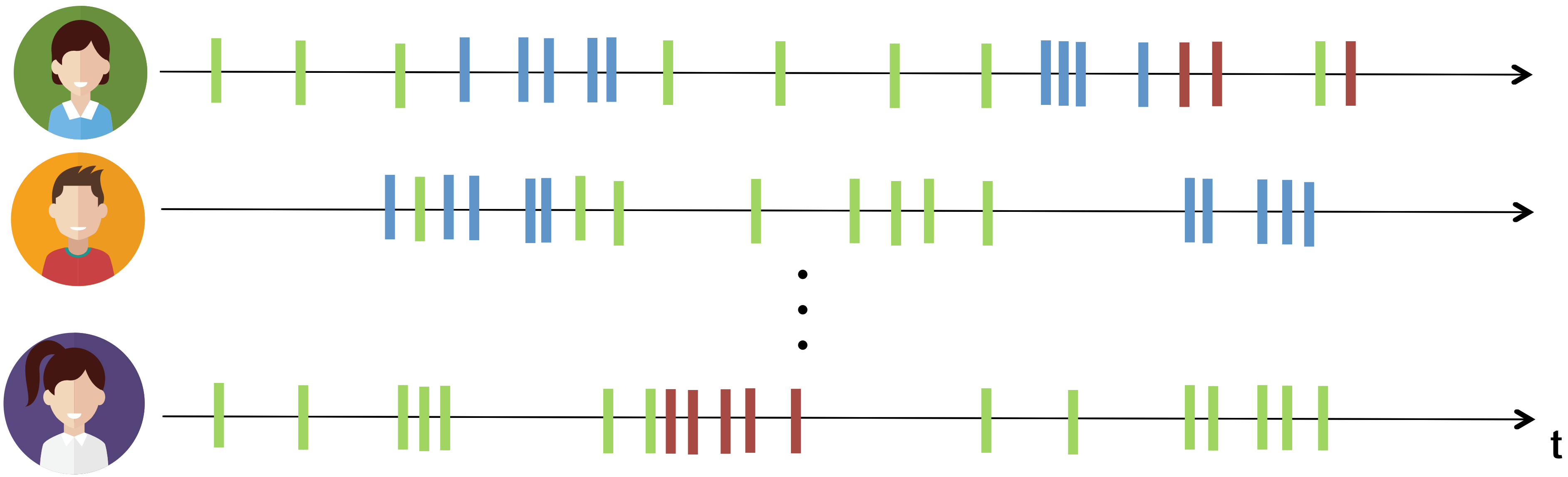
presentation + week



Clustering Event Sequences

Several people share same clusters

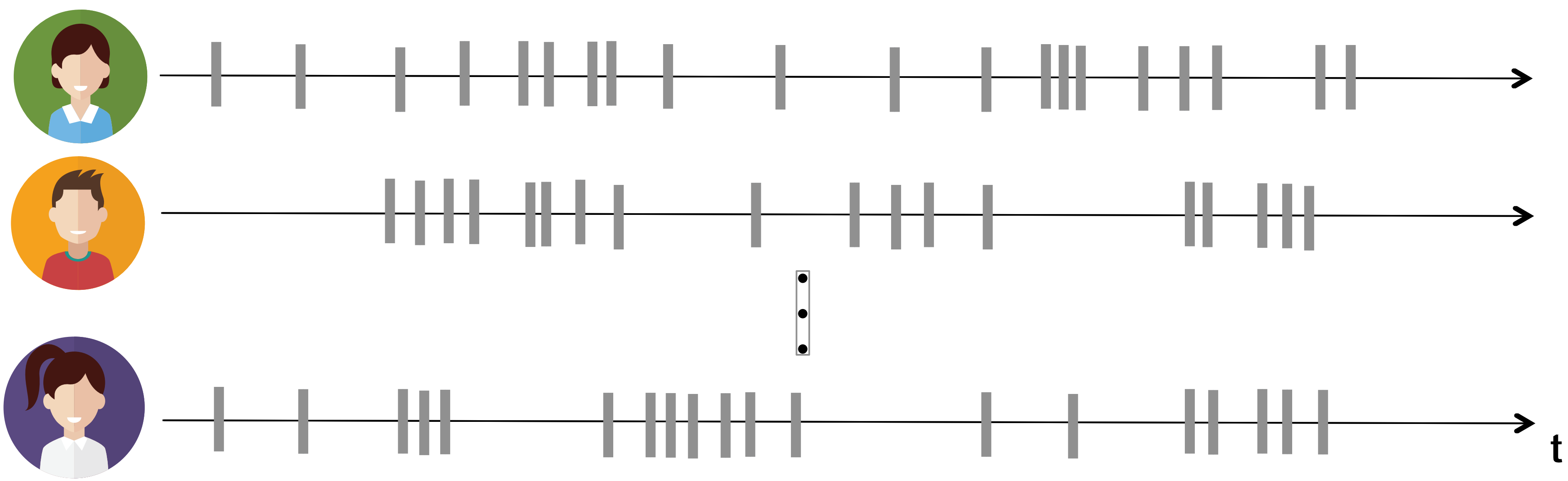
- Introduction to programming
- Discrete math
- Project presentation



Clustering Event Sequences

Event cluster (topic) is hidden ➔ **Clustering of events**

Unknown number of clusters ➔ **Dirichlet Process**



Outline of the Lecture

Intro to Temporal Point processes (TPPs)

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

Application: Clustering Event Sequences

1. Problem Statement
2. Introduction to DPMM
3. CRP + HP (a.k.a. HDHP)
4. Generative process

APPLICATION:

CLUSTERING EVENT SEQUENCES

1. Problem Statement
2. **Introduction to DPs**
3. CRP + HP (a.k.a. HDHP)
4. Generative Process

Dirichlet Process (DP)

Dirichlet Process:

Random process whose realization consists of **probability distributions**

Concentration parameter

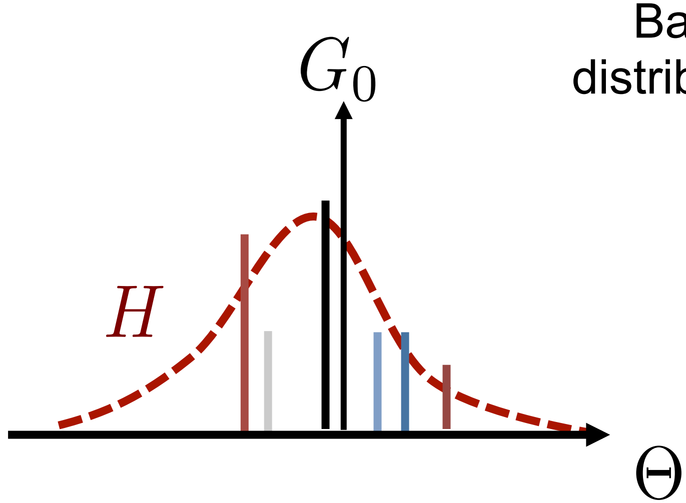
$$G_0 \sim DP(\beta, H)$$

Base distribution

$$\begin{cases} G_0 = \sum_{\ell=1}^{\infty} \pi_\ell \delta_{\theta_\ell} \\ \boldsymbol{\pi} = (\pi_\ell)_{\ell=1}^{\infty} \sim GEM(\beta) \\ \theta_\ell \sim H(\theta) \end{cases}$$

Stick breaking

Cluster parameters



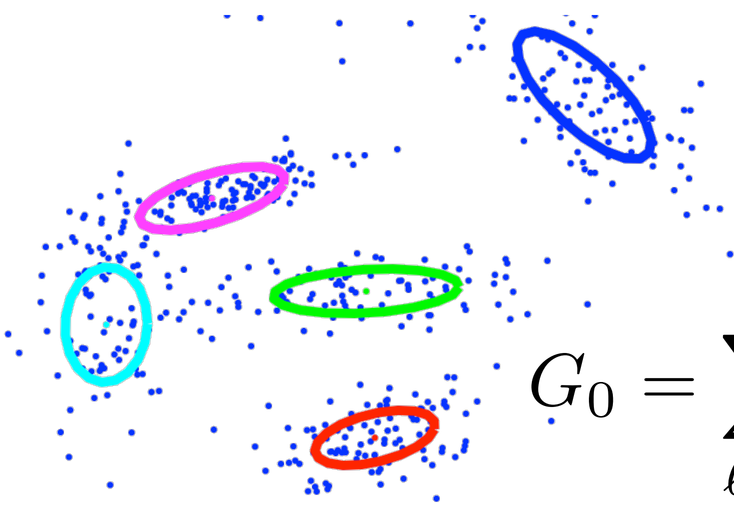
Dirichlet Process (DP)

Properties:

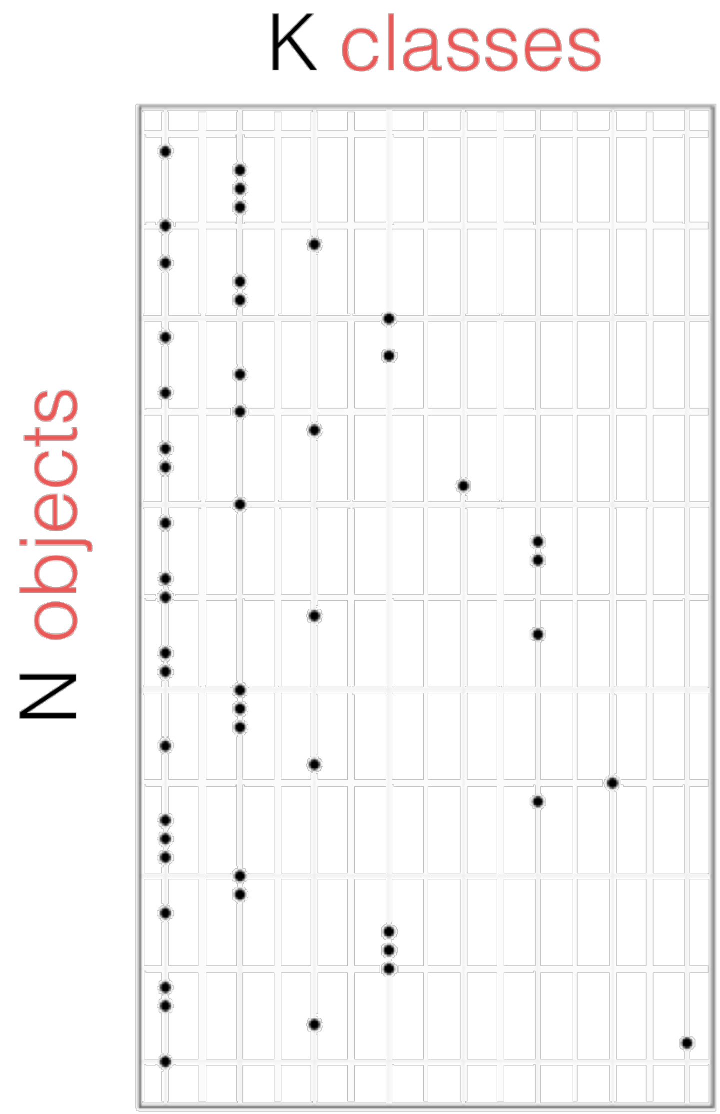
- Infinite-dimensional generalization of Dirichlet distribution

$$\pi \sim \mathrm{Lim}_{K \to \infty} \mathit{Dirichlet}(\alpha/K, \ldots, \alpha/K)$$

- Prior for clustering
- Infinite model complexity = infinite # of groups
- Partition data into groups



$$G_0 = \sum_{\ell=1}^{\infty} \pi_\ell \delta_{\theta_\ell}$$



Conjugacy to the multinomial

- For the Dirichlet distribution we could integrate out π to get: $P(z_j = k|\mathbf{z}_{\neg j}) \propto \sum_{i \neq j} \mathbb{1}(z_i = k) + \alpha_k$.
- We can do something similar for the Dirichlet Process.
- Let m_k be the number of times we have seen $x_i = \theta_k$ in the (first) n observations.
- ... or the number of times that $z_i = k$ for K^+ different values (so far).
- The posterior over G given n observations is:

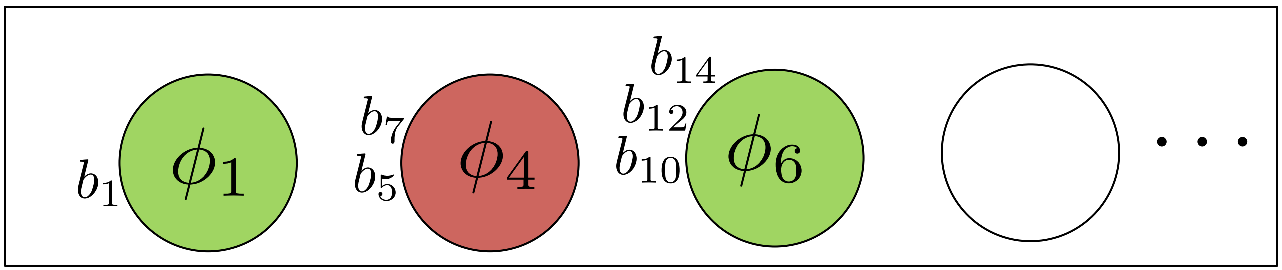
$$G|x_1 \ldots, x_n \sim DP\left(\alpha + n, \frac{\alpha H + \sum_{k=1}^{K^+} m_k \delta_{\theta_k}}{\alpha + n}\right)$$

- So, we have

$$P(z_{n+1} = k|z_1, \ldots, z_n) = \begin{cases} \frac{m_k}{n+\alpha}, & k \leq K^+ \\ \frac{\alpha}{n+\alpha}, & k = K^+ + 1 \end{cases}$$

Results in the CRP

Chinese Restaurant Process (CRP)



Exchangeability:

✓Observations

✓Clusters

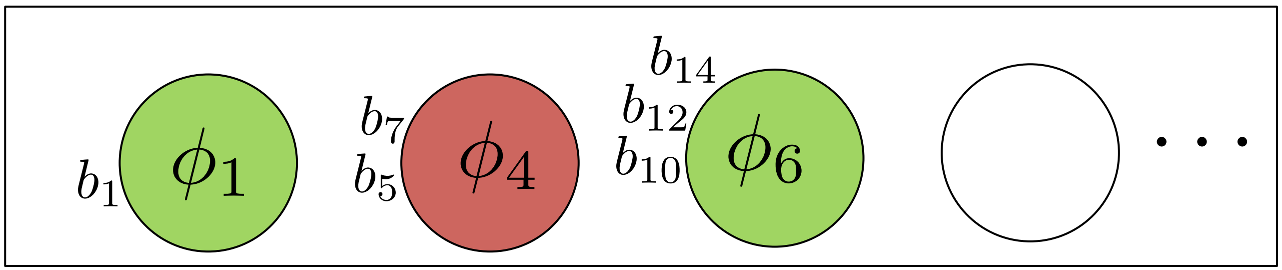
1. Table assignment:

$$p(b_{n+1} = \ell | b_1, \ldots, b_n) = \begin{cases} \frac{m_\ell}{n+\beta} & k \leq K^+ \\ \frac{\alpha}{n+\beta} & k + K^+ + 1 \end{cases}$$

2. Cluster (dish) assignment:

$$\phi_{j(K+1)} = \begin{cases} \theta_\ell & \text{w.p. } \frac{m_\ell}{K+\beta_1} \\ \theta_{L+1} & \text{w.p. } \frac{\beta_1}{K+\beta_1} \end{cases} \qquad \text{for } \ell = 1, \ldots, L$$

Chinese Restaurant Process (CRP)



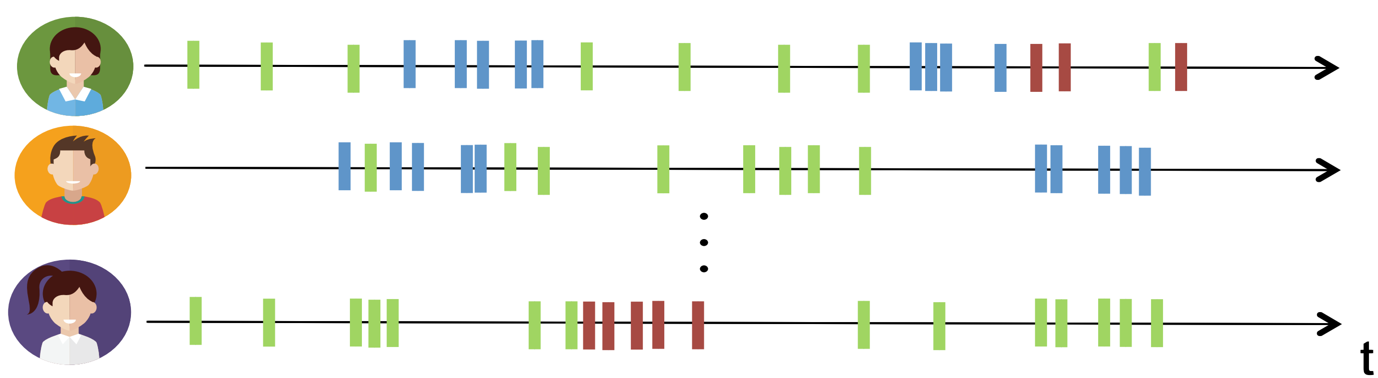
Exchangeability:

✓Observations

✓Clusters

CLUSTERING EVENT SEQUENCES:

- Each user perform a sequence of events
- Events are not exchangeable



Solution
CRP + HP

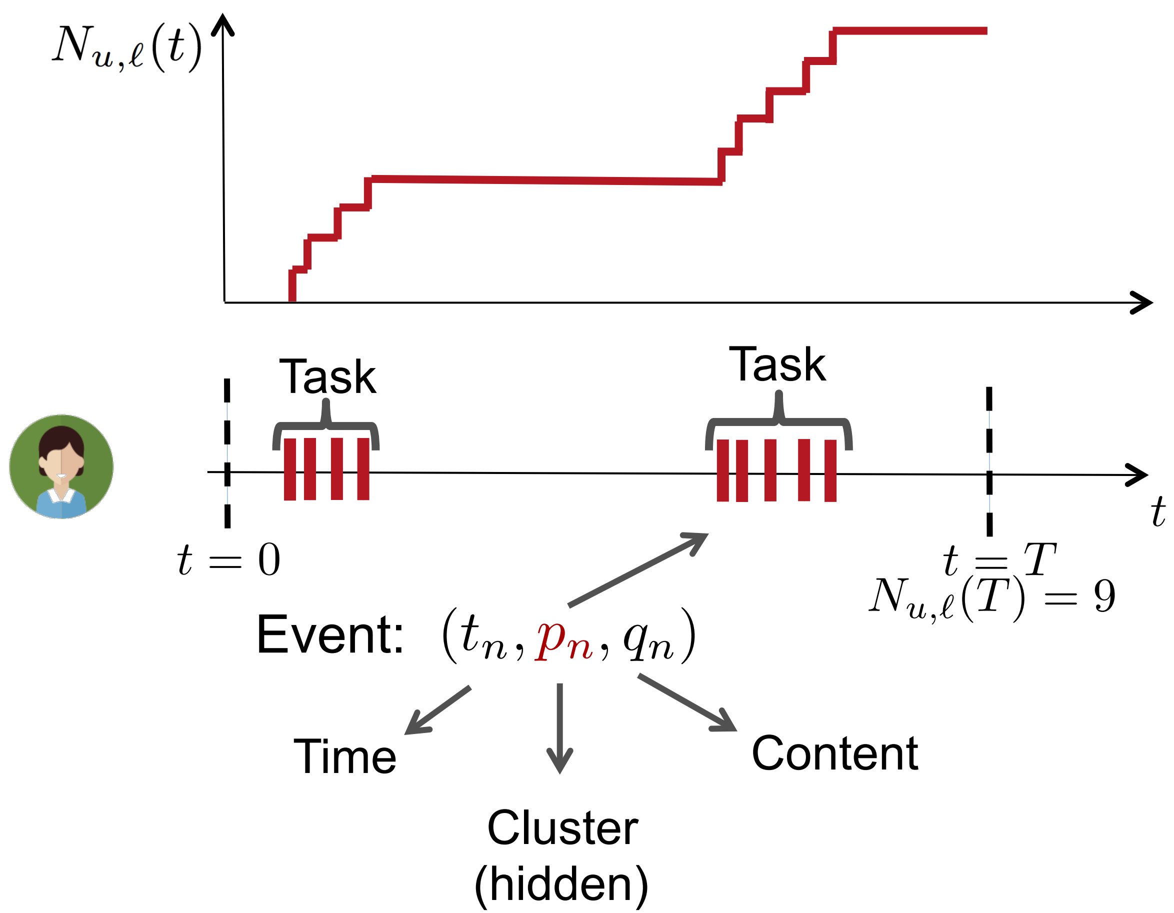
APPLICATION:

CLUSTERING EVENT SEQUENCES

1. Problem Statement
2. Introduction to DPs
3. **CRP + HP (a.k.a. HDHP)**
4. Generative Process

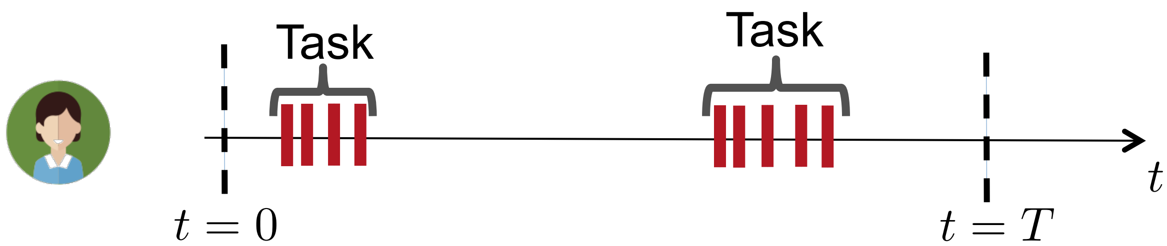
CRP + HP (I)

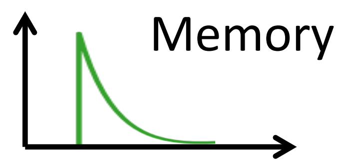
For each user and cluster, we represent events as a counting process:



CRP + HP (II)

Intensity for each user and cluster from a Hawkes process:





New task rate

Cluster probability (popularity)

$$\lambda^*_{u,\ell}(t) = \mu_u \pi_\ell + \sum_{j: t_j \in \mathcal{H}_{u,\ell}(t)} k_{\theta_\ell}(t - t_j)$$

Intensity or rate (events / hour) — $\lambda^*_{u,\ell}(t)$

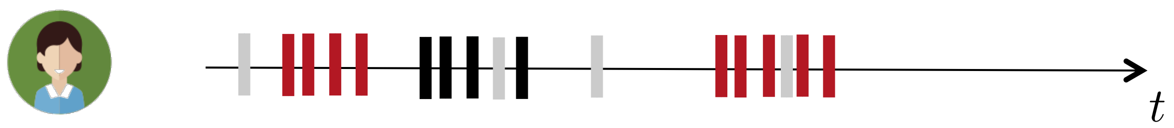
Exogeneous event — $\mu_u \pi_\ell$

Endogenous event — $\sum_{j: t_j \in \mathcal{H}_{u,\ell}(t)} k_{\theta_\ell}(t - t_j)$

Hawkes process

CRP + HP (III)

Users adopt more than one learning pattern



User's events sampled for an infinite multivariate Hawkes:

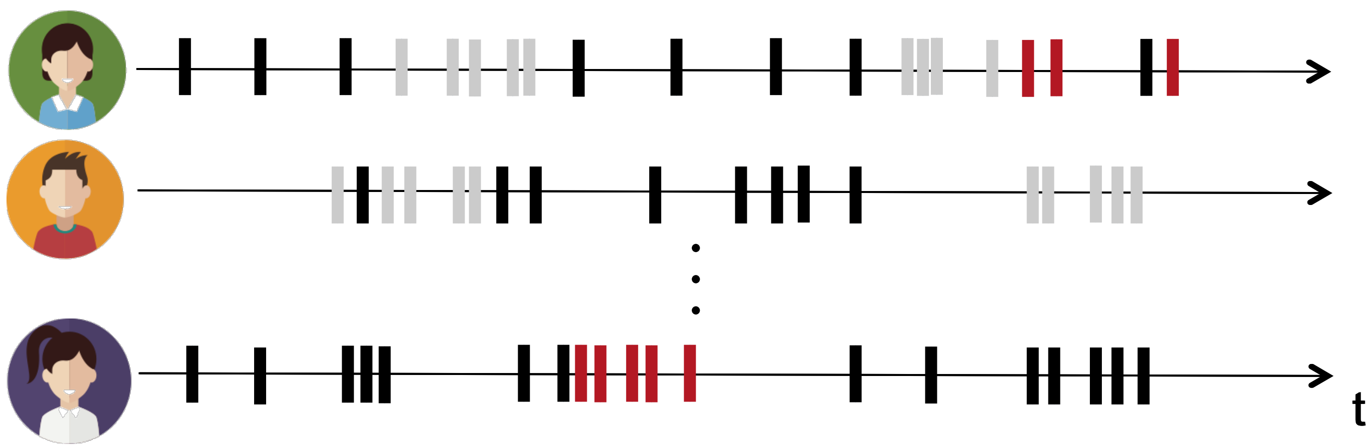
Time, Cluster

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

Mark (content/words) $\longrightarrow \omega_j \sim Multinomial(\boldsymbol{\theta}_p)$

CRP + HP (IV)

Different users adopt same learning patterns



Learning pattern distribution from a **Dirichlet process:**

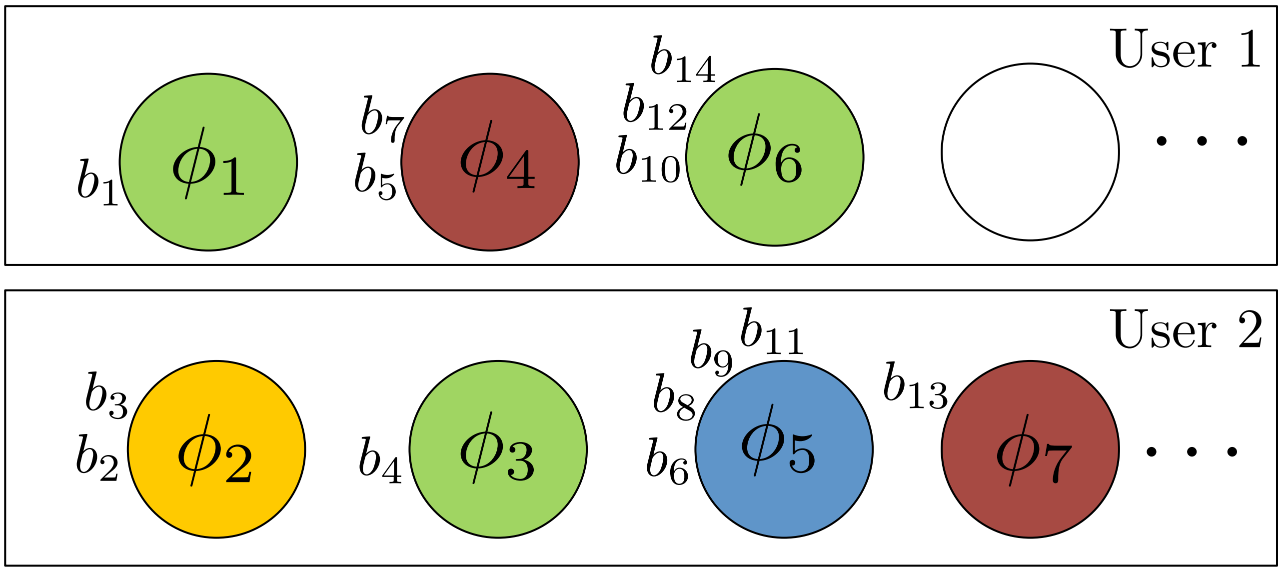
- Infinite # of learning patterns.
- Shared parameters across users.

How to generate samples (events)?

APPLICATION:
CLUSTERING EVENT SEQUENCES

1. Problem Statement
2. Introduction to DPs
3. CRP + HP (a.k.a. HDHP)
4. **Generative Process**

Generative Process (I)



Exchangeability:

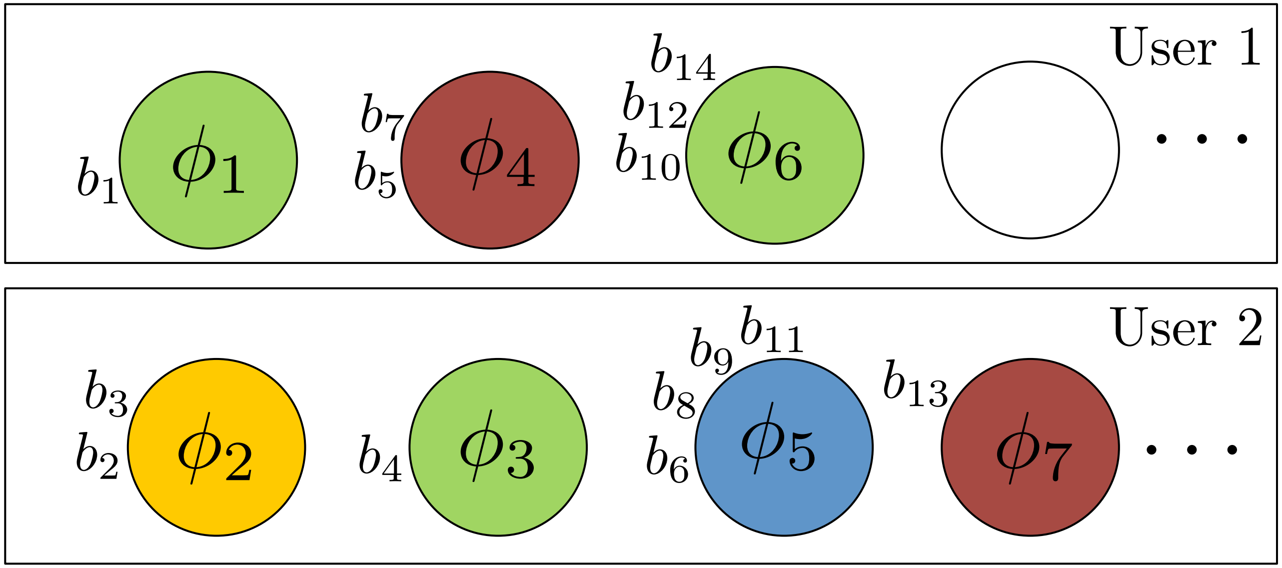
✗ Events

✓ Clusters

1. Time and user sampling:

$$(t_n, u_n) \sim \text{Hawkes}\begin{pmatrix} \lambda_0 + \sum_{i:t_i \in \mathcal{H}_u(t_n)} \kappa_{b_i}(t_n, t_i) \\ \vdots \\ \lambda_0 + \sum_{i:t_i \in \mathcal{H}_{\mathcal{U}}(t_n)} \kappa_{b_i}(t_n, t_i) \end{pmatrix}$$

Generative Process (I)



Exchangeability:

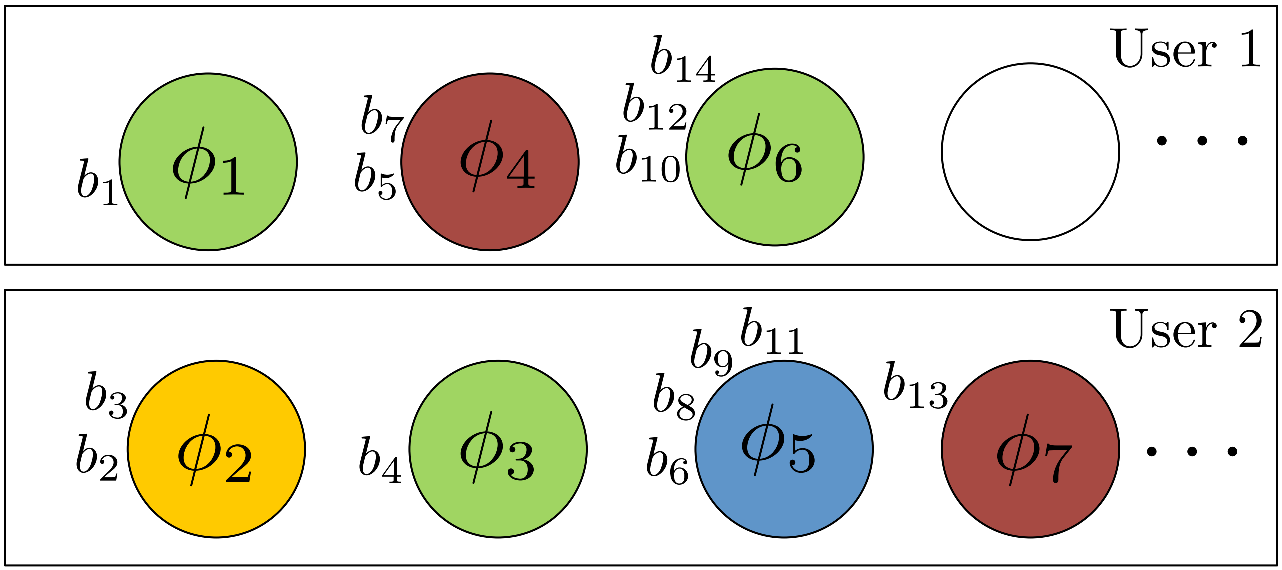
✗ Events

✓ Clusters

2. Task (table) assignment :

$$b_n = \begin{cases} k & \text{w. p.} \quad \frac{\lambda_{u_n,k}(t_n)}{\lambda_{u_n}(t_n)}, \quad \text{for} \quad k = 1, \ldots, K \\ K_{new} & \text{w. p.} \quad \frac{\lambda_0}{\lambda_{u_n}(t_n)} \end{cases}$$

Generative Process (I)



Exchangeability:

✗ Events

✓ Clusters

3. Learning pattern (dish) assignment:

$$\phi_{j(K+1)} = \begin{cases} \theta_\ell & \text{w.p. } \frac{m_\ell}{K+\beta_1} \\ \theta_{L+1} & \text{w.p. } \frac{\beta_1}{K+\beta_1} \end{cases} \qquad \text{for } \ell = 1, \ldots, L$$

Generative Process (II)

General approach to sample from BNP+HP processes:

Algorithm 1: BNP+HP: Generative process

Input: a BNP, triggering kernel function $\gamma(\cdot)$, N

Output: $\{e_n = (t_n, u_n, m_n, z_n)\}_{n=1}^{N}$

for $n = 1 \dots N$ **do**

  Compute $\lambda(t) = \sum_u \lambda_u(t|\mathcal{H}(t))$;

  Sample $t_n \sim \mathsf{Hawkes}(\lambda(t))$;

  Sample $u_n \sim \mathsf{Cat}(\{\lambda_u(t_n)/\lambda(t_n)\}_{u\in\mathcal{U}})$;

  Sample $b_n \sim \mathsf{Ber}(\lambda_{u_n}^{\text{endo}}(t_n)/\lambda_{u_n}(t_n))$;

  if $b_n = 1$ **then**

    Sample $z_n \sim \mathsf{Cat}(\{\lambda_{u_n,k}^{\mathsf{endo}}(t_n)/\lambda_{u_n}^{\mathsf{endo}}(t_n)\}_{k=1}^{K})$;

  else

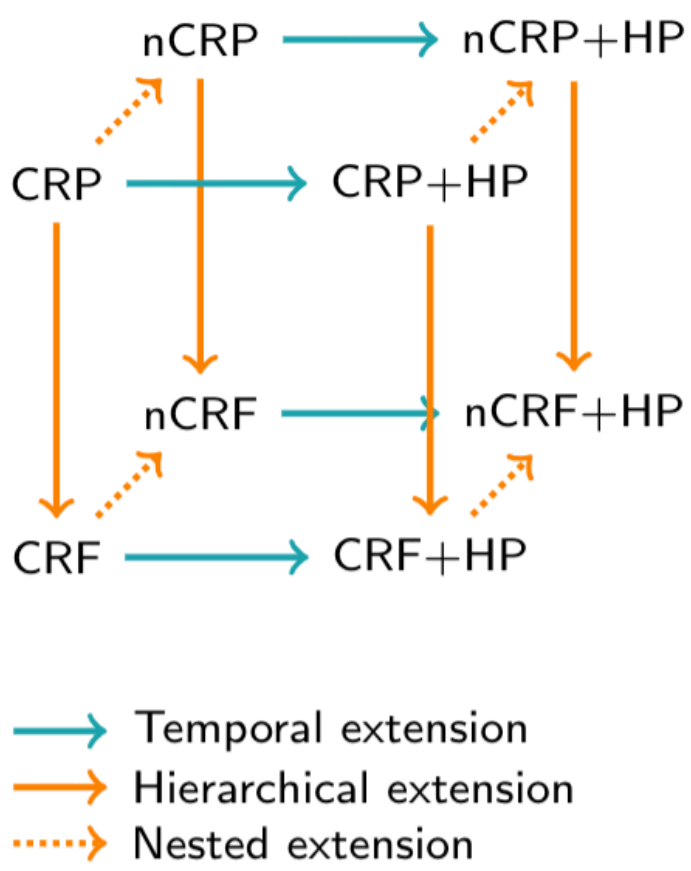
    Sample $z_n \sim \mathsf{BNP}$;

  Update $\mathcal{H}_{u_n}(t)$;

  if $z_n = K + 1$ **then**

    Sample parameters for the new pattern;

  Sample mark $m_n \sim p(m_n|z_n)$;



Generative Process (II)

Suitable for online inference (Sequential Monte Carlo):

Algorithm 2: BNP+HP: inference

Input: A sequence of events $\{e_n = (t_n, u_n, m_n)\}_{n=1}^{N}$, P number of particles, θ threshold for particle resampling, triggering kernels prior $\Gamma(a_\alpha, b_\alpha)$, user intensities prior (a_μ, b_μ),τ

Output: user intensities $\{\mu_u\}_{u=1}^{U}$, triggering kernels $\{\alpha_l\}_{l=1}^{L}$, BNP-specific counts and parameters

Initialize $\mu_u \sim \Gamma(a_\mu, b_\mu) \forall u \in \mathcal{U}$, all counts to 0;
Initialize particle weights $w_n^{(p)} = \frac{1}{P} \quad \forall p \in 1 \dots P$;
for $n = 1 \dots N$ **do**
    `// for all particles`
    for $p = 1 \dots P$ **do**
        Sample pattern z_n;
        if z_n *is globally new* **then**
            Sample $\alpha_{z_n} \sim \Gamma(a_\alpha, b_\alpha)$;
        Update triggering kernel prior for pattern $a_{\alpha_{z_n}}, b_{\alpha_{z_n}}$;
        Compute $p(m_n | z_{1:n-1})$;
        Update user intensities μ_u parameters $\forall u \in 1 \dots U$;
        Compute $p(t_n | u_n)$;
        Update $w_n^{(p)} = w_{n-1}^{(p)} p(m_n | z_{1:n-1}) p(t_n | u_n)$;
        Update BNP-specific counts;
    Normalize particle weights;
    if $\sum_{p=1}^{P} {w_n^{(p)}}^2 \leq \theta$ **then**
        Resample particles using systematic resampling;

Example: Online Learning Activity

Experiments

We gathered data from stackoverflow . Over 4 years (09/2010 – 08/2014):

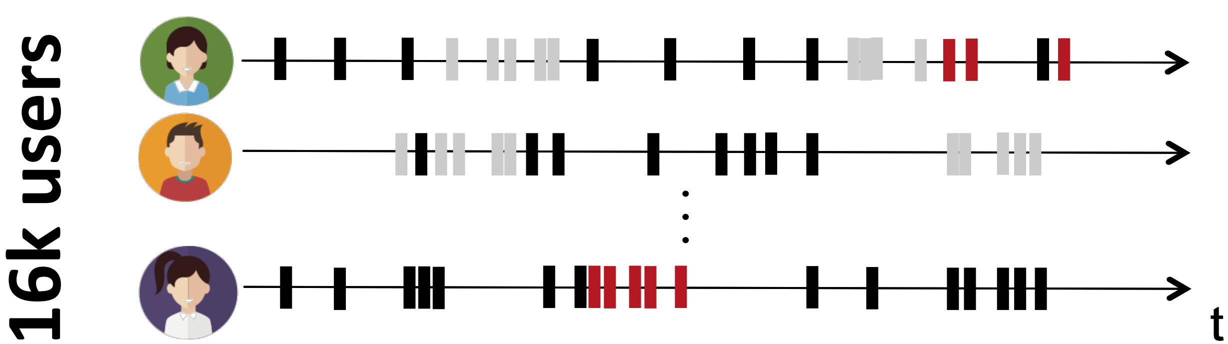
learning event: question (and answers)

user: user who asked the question

time: time when the question was asked

content: question tags

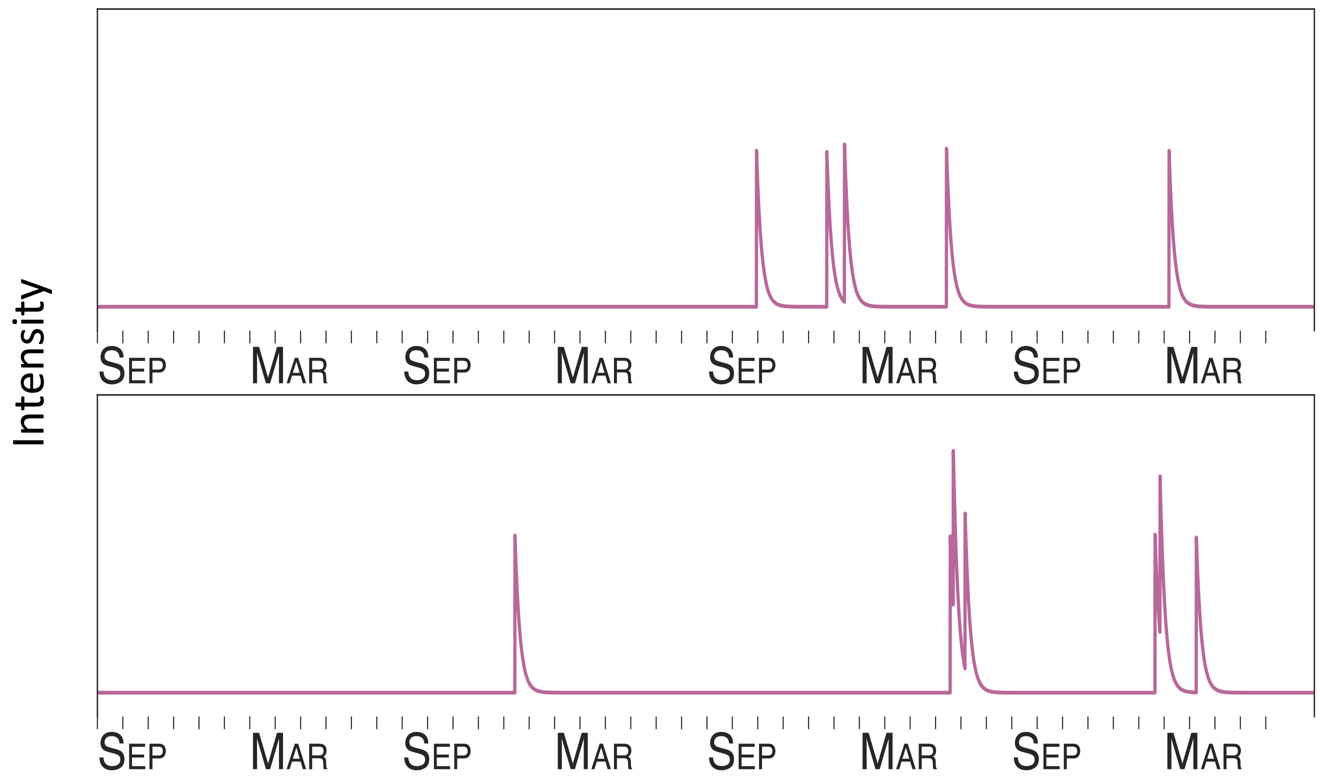
1.6 million questions using **31.4k tags**



Example: Online Learning Activity

Content

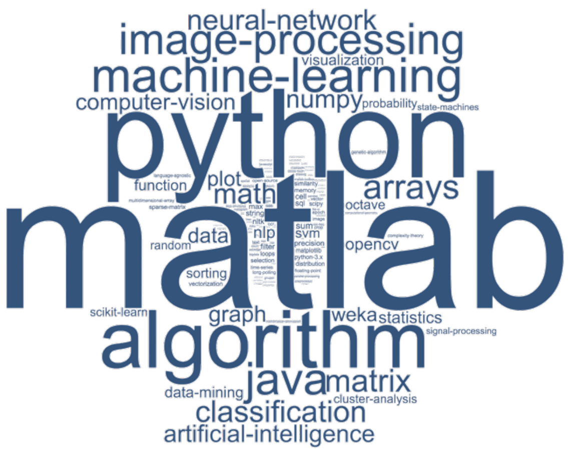
Intensities



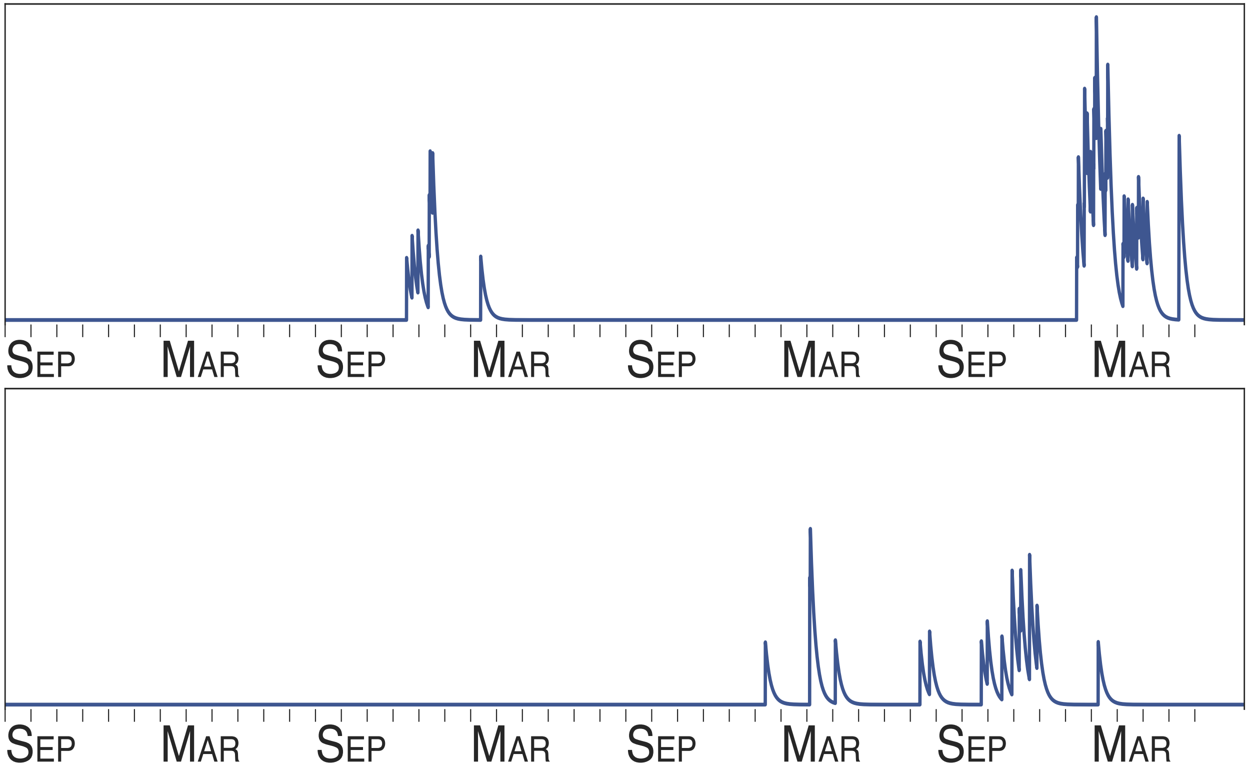
Version control tasks tend to be specific, quickly solved after performing few questions

Example: Online Learning Activity

Content



Intensities



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

Outline of the Lecture

Intro to Temporal Point processes (TPPs)

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

Application: Clustering Event Sequences

1. Problem Statement
2. Introduction to DPMM
3. CRP + HP (a.k.a. HDHP)
4. Generative process

THANKS!

More about TTPs

Temporal Point processes (TPPs): Intro

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

RL & Control

1. Marked TPPs: a new setting
2. Stochastic optimal control
3. Reinforcement learning

Slides/references: learning.mpi-sws.org/tpp-icml18

THANKS!