Machine learning for
Dynamic Social Network Analysis

Applications: Models

Manuel Gomez Rodriguez
Max Planck Institute for Software Systems

IIT HYDERABAD, DECEMBER 2017

Outline of the Seminar

REPRESENTATION: TEMPORAL POINT PROCESSES

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

APPLICATIONS: MODELS

1. Information propagation
2. Opinion dynamics
3. Information reliability
4. Knowledge acquisition

This lecture

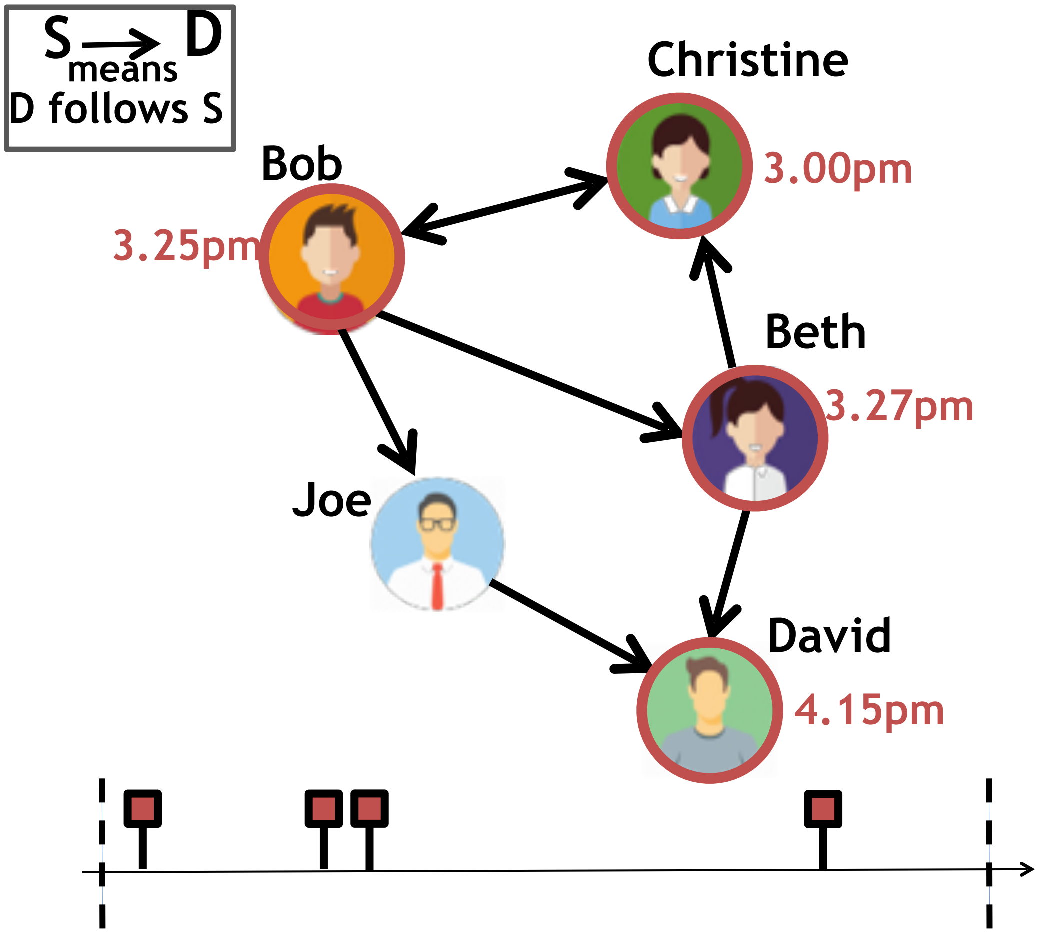
APPLICATIONS: CONTROL

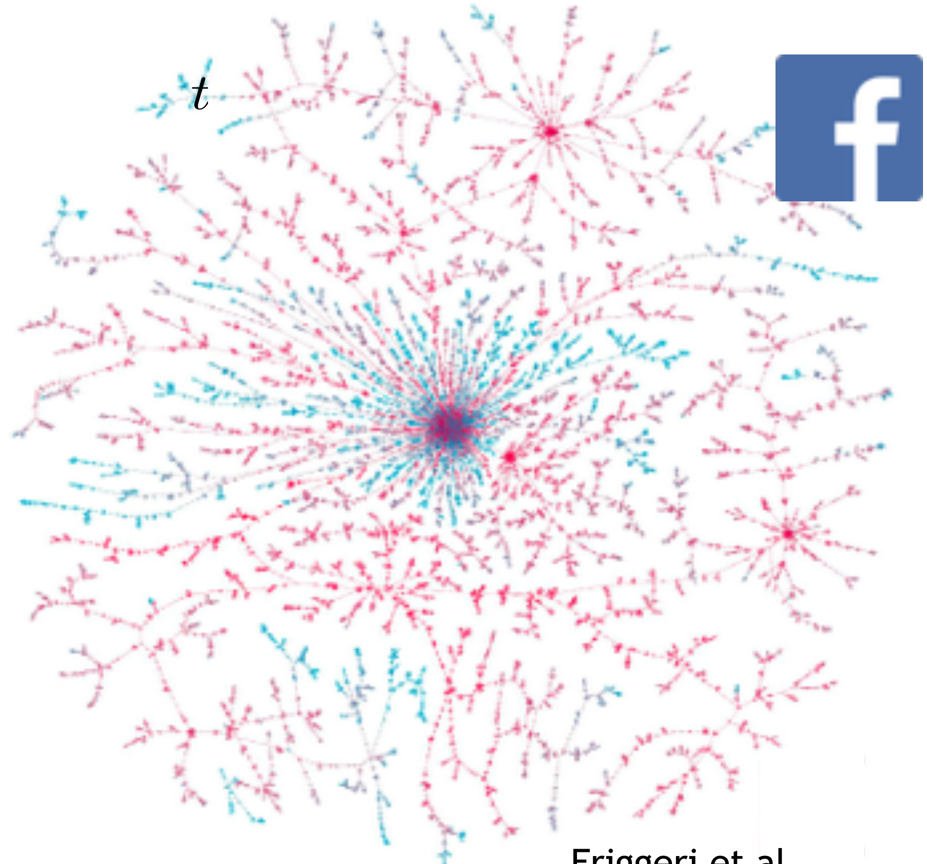
1. Influence maximization
2. Activity shaping
3. When to post
4. When to fact check

Applications:
Models

1. Idea adoption
2. Opinion dynamics
3. Information reliability
4. Knowledge acquisition

Idea adoption: an example





Friggeri et al., 2014

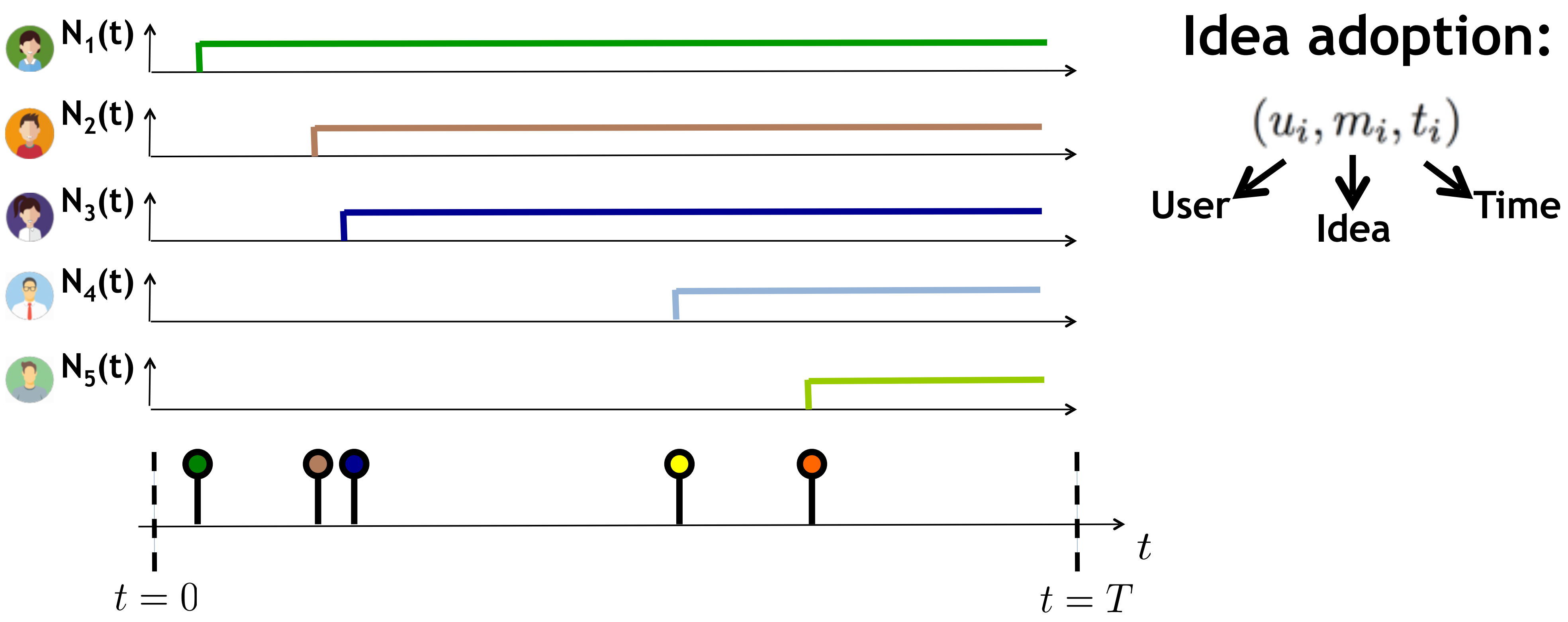
They can have an impact in the off-line world

theguardian

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

Idea adoption representation

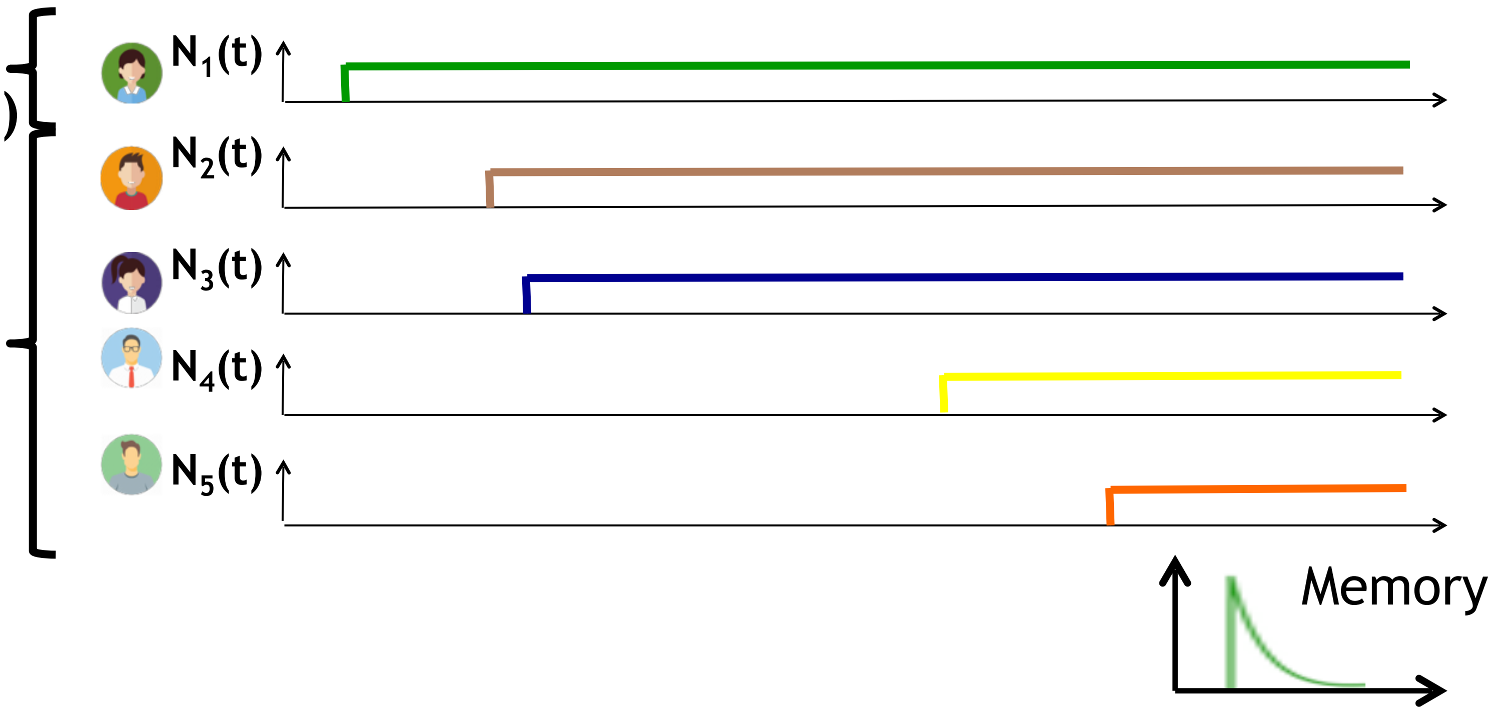
We represent an idea adoptions using **terminating temporal point processes:**



Idea adoption intensity

Source
(given, not modeled)

Follow-ups
(modeled)



$$\lambda_u^*(t) = \underbrace{(1 - N(t))}_{\text{Adopt idea only once}} \sum_{v \in [m]} b_{vu} \underbrace{\sum_{e_i \in \mathcal{H}_v(t)} \kappa(t - t_i)}_{\text{Previous messages by user v}}$$

Influence from user v on user u

[Gomez-Rodriguez et al., ICML 2011]

Model inference from multiple adoptions

Conditional intensities

$$\lambda_u^*(t)$$

Idea adoption log-likelihood

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

Sum up log-likelihoods of multiple ideas!

Maximum likelihood approach to find model parameters!

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

[Gomez-Rodriguez et al., ICML 2011]

Topic-sensitive rates

Topic-modulated influence:

$$b_{vu} = \sum_{l=1}^{K} b_{vu}^{l} m_l$$

LDA weight for topic l

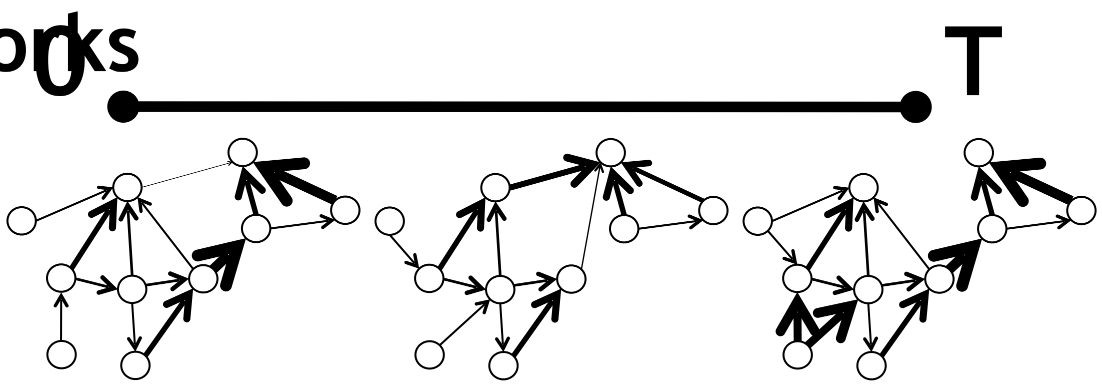
[Du et al., AISTATS 2013]

Dynamic influence

In some cases, influence change over time:

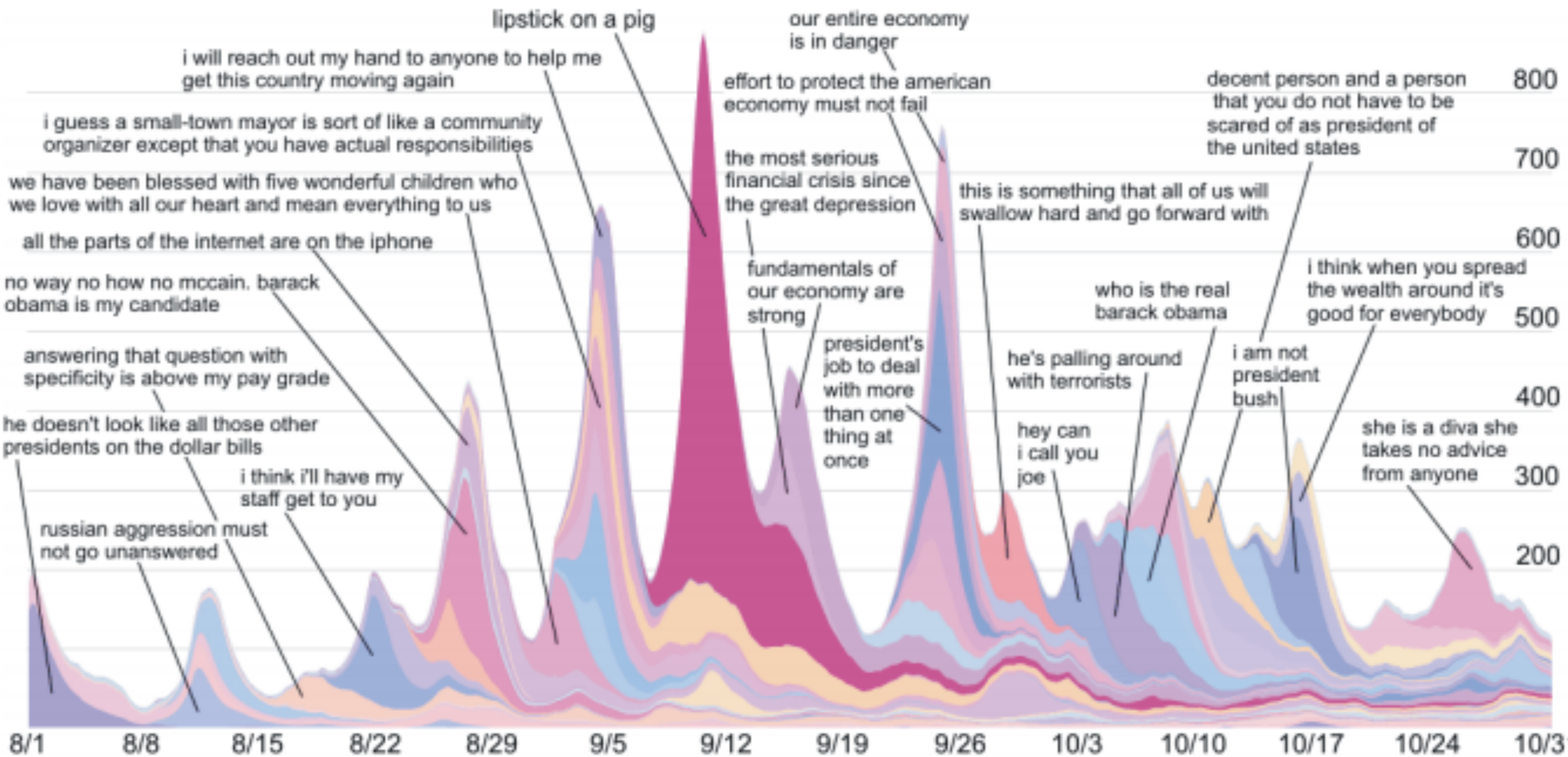
#greece retweets

Propagation over networks with variable influence



[Gomez-Rodriguez et al., WSDM 2013]

Memetracker



[Leskovec et al., KDD '09]

Insights I: real world events

Youtube video: http://youtu.be/hBeaSfRCU4c

Insights II: dynamic clusters

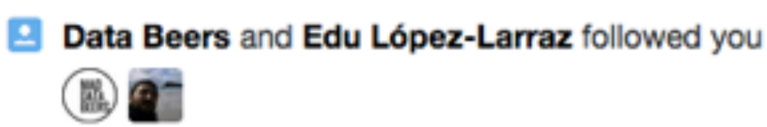
Youtube video: http://youtu.be/hBeaSfRCU4c

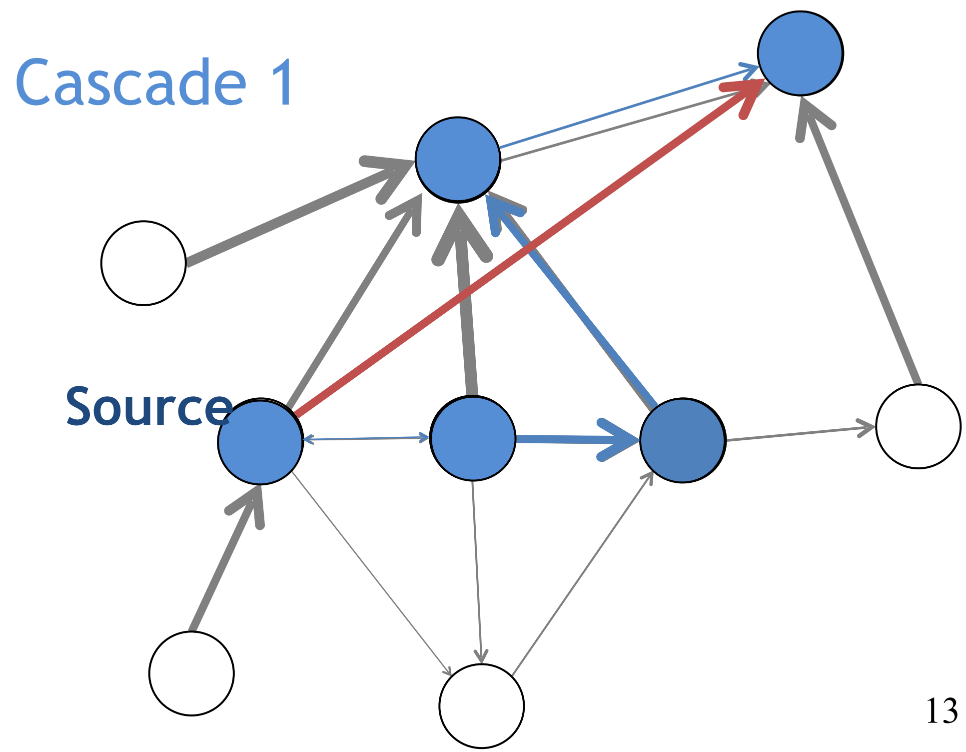
Cascades and network evolution

Recent empirical studies [Antoniades and Dovrolis, Myers & Leskovec] show that **cascades also change the structure of social networks:**

Information propagation triggers new links

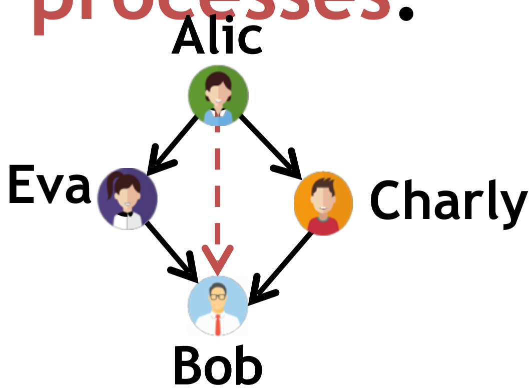
↓ **1 hour later**



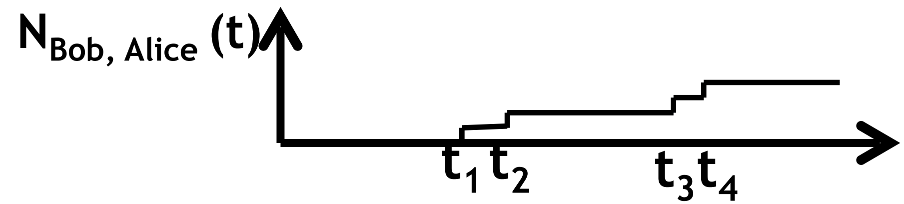


Co-evolution as interwoven point processes (I)

We model user's retweet and link events as nonterminating and terminating counting processes:



Bob *retweets (is exposed to)* Alice



Bob follows Alice

Key idea

Both counting processes have **memory** and **depend on each other**

Co-evolution as interwoven point processes (II)

We characterize retweet and link counting processes using their respective conditional intensities:

$$\mathbb{E}[\overbrace{d\boldsymbol{N}(t)}^{\text{Changes on retweets in [t, t+dt]}} \,|\, \underbrace{\mathcal{H}^r(t) \cup \mathcal{H}^l(t)}_{\text{History of retweets and links up to t}}] = \underbrace{\boldsymbol{\Gamma}^*(t)}_{\text{Instantaneous rates or intensities}}\, dt$$

$$\mathbb{E}[\underbrace{d\boldsymbol{A}(t)}_{\text{Changes on links in [t, t+dt]}} \,|\, \overbrace{\mathcal{H}^r(t) \cup \mathcal{H}^l(t)}] = \overbrace{\boldsymbol{\Lambda}^*(t)}\, dt$$

They are coupled through the histories

$$\mathcal{H}^r(t) \cup \mathcal{H}^l(t)$$

Intensity for information propagation

$N_{us}(t)$

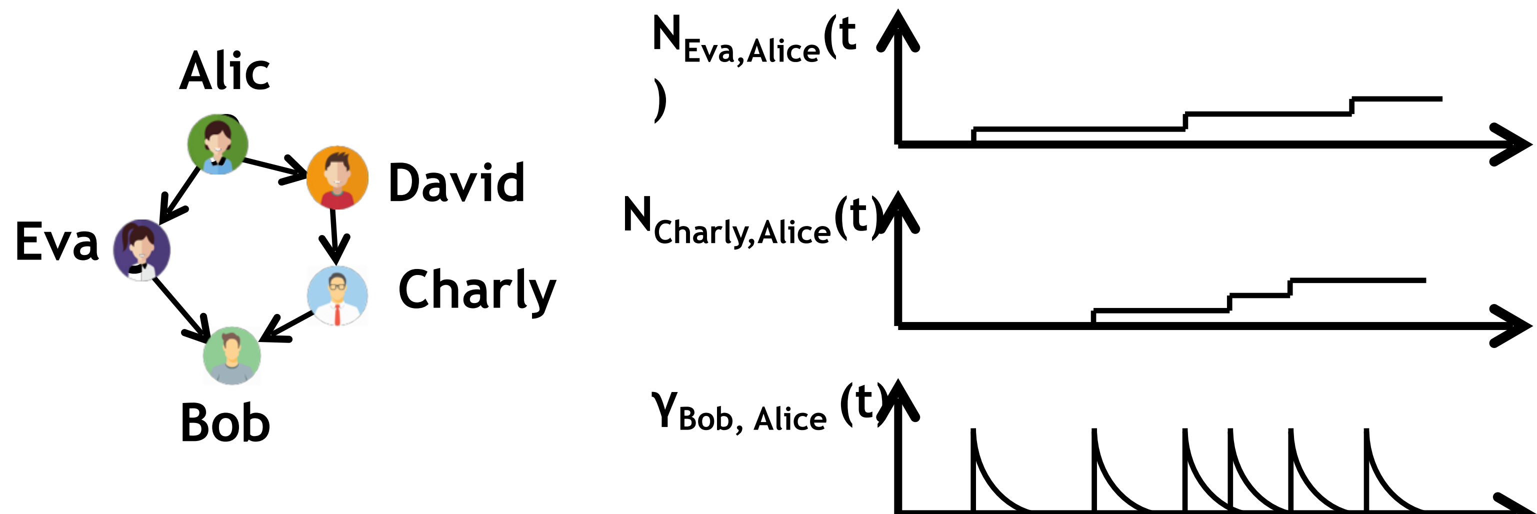
$\Updownarrow$

Tweets on her own initiative

Time-varying network topology

$$\gamma^*_{us}(t) = \begin{cases} \eta_u & u = s \\ \underbrace{\beta_s \sum_{v\in\mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star \left(A_{uv}(t)\; dN_{vs}(t)\right)}_{\text{Propagation of peer influence over the network}} & u \neq s \end{cases}$$

Node u does not need to follow s!

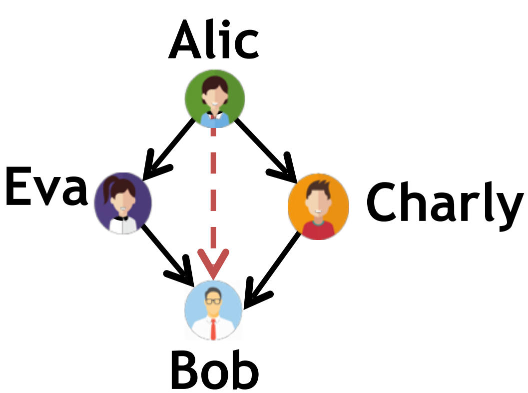


Intensity for network evolution

$$\lambda^*_{us}(t) = \underbrace{(1 - A_{us}(t))}_{\text{Ensures a link is created only once}}(\mu_u + \underbrace{\alpha_u\, \kappa_{\omega_2}(t) \star dN_{us}(t)}_{\text{Influence of retweet intensity on the link creation}})$$

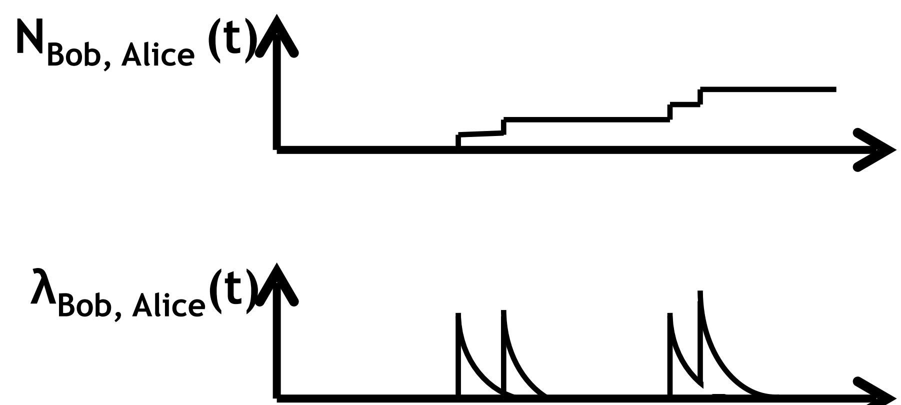
$A_{us}(t)$

Links on her own initiative



Bob *retweets (is exposed to)* Alice

Bob's *risk* of following Alice



Model inference from historical data

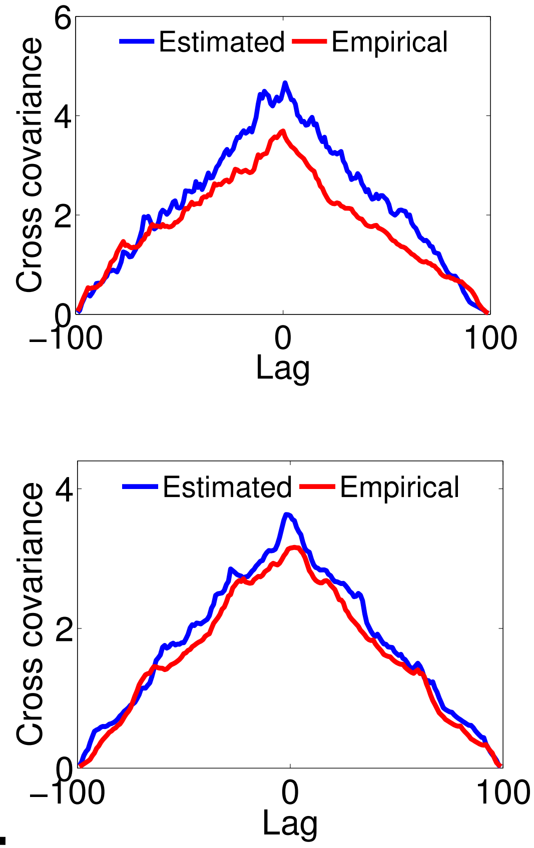
Find *optimal* parameters using **maximum likelihood estimation (MLE):**

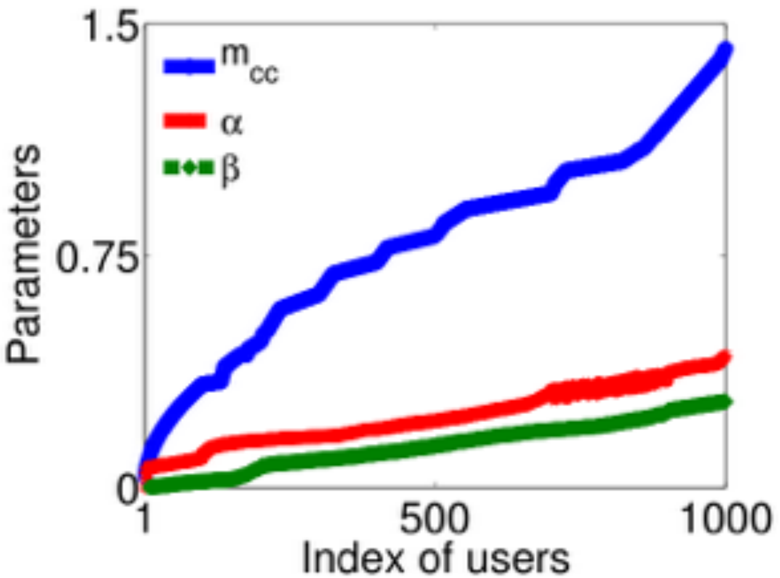
$$\mathfrak{L}(\{\mu_u\},\{\alpha_u\},\{\eta_u\},\{\beta_s\}) = \underbrace{\sum_{e_i^r \in \mathcal{E}} \log\left(\gamma^*_{u_i s_i}(t_i)\right) - \sum_{u,s\in[m]} \int_0^T \gamma^*_{us}(\tau)\,d\tau}_{\text{tweet / retweet}} + \underbrace{\sum_{e_i^l \in \mathcal{A}} \log\left(\lambda^*_{u_i s_i}(t_i)\right) - \sum_{u,s\in[m]} \int_0^T \lambda^*_{us}(\tau)\,d\tau}_{\text{links}}$$

For the choice of information propagation and link intensities, the MLE problem above is **parallelizable & convex.**

Retweet and link coevolution

Cross-covariance for two users

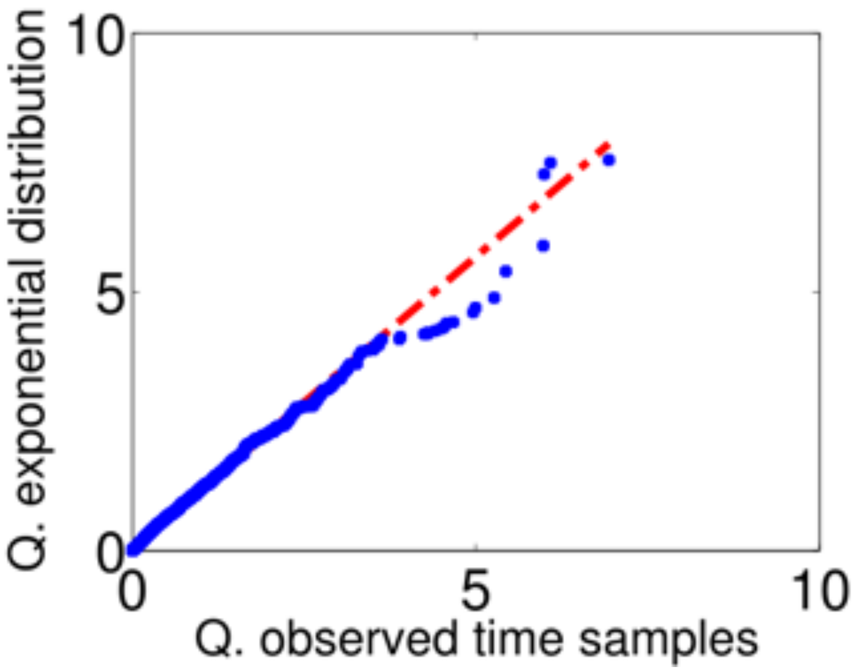




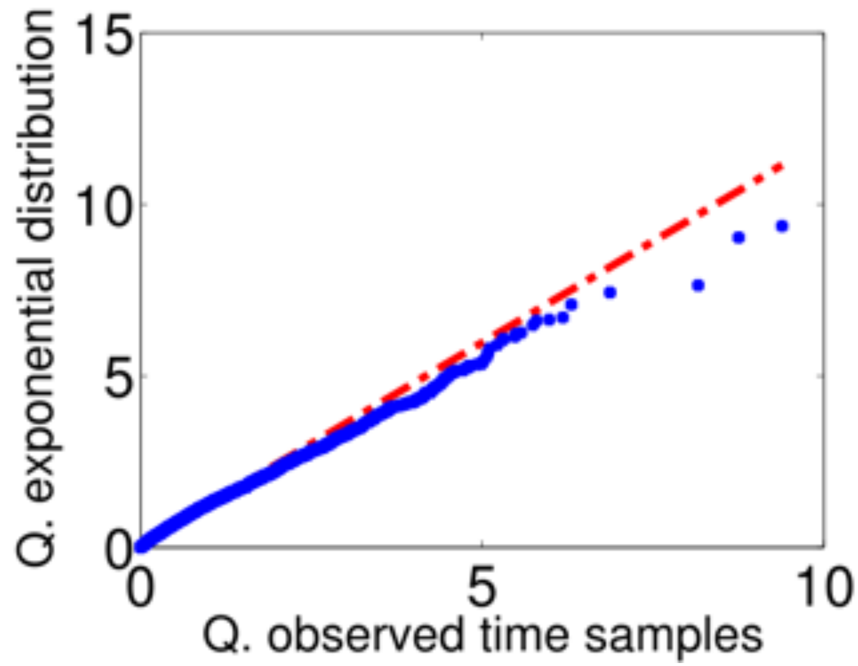
Average cross-covariance vs model parameters

The fitted model generate **link and information diffusion events that coevolve similarly** (in terms of cross-covariance) **as real events.**

Model checking



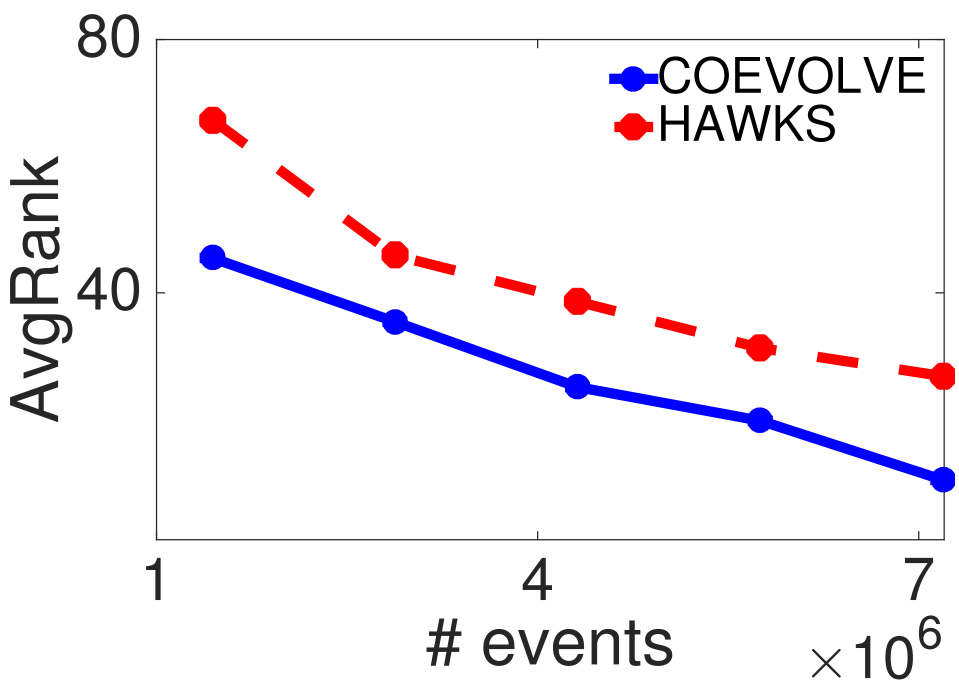
Link events



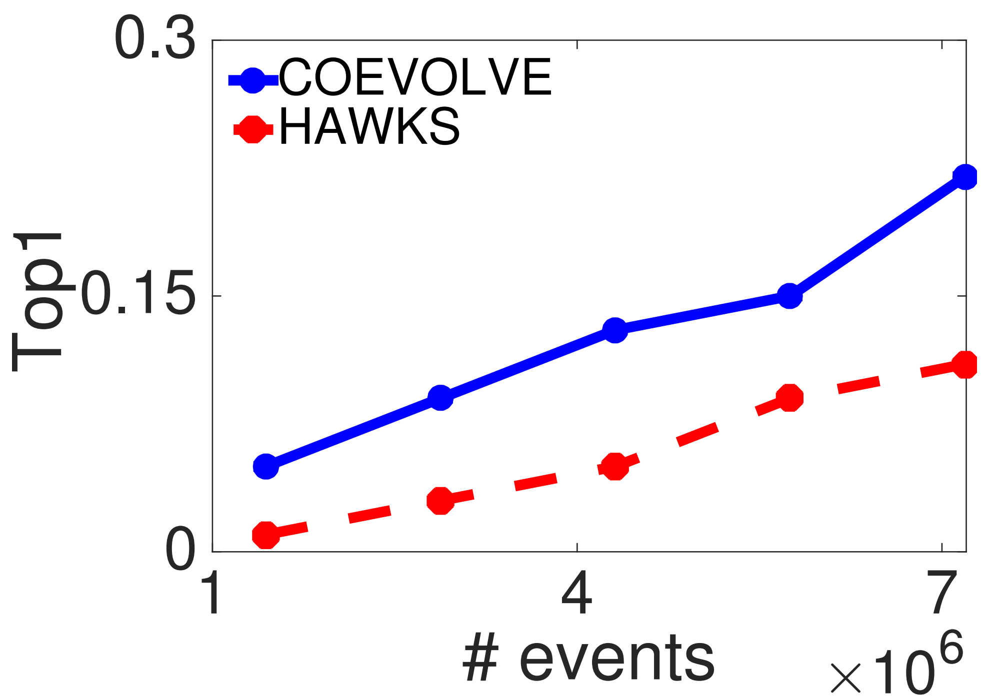
Retweet events

The quantiles of the intensity integ $\int_{t_i}^{t_{i+1}} \lambda(t)\, dt$ computed using the fitted intensities match the quantiles of the unit-rate exponential distribution

Information diffusion prediction



Average rank



Success probability

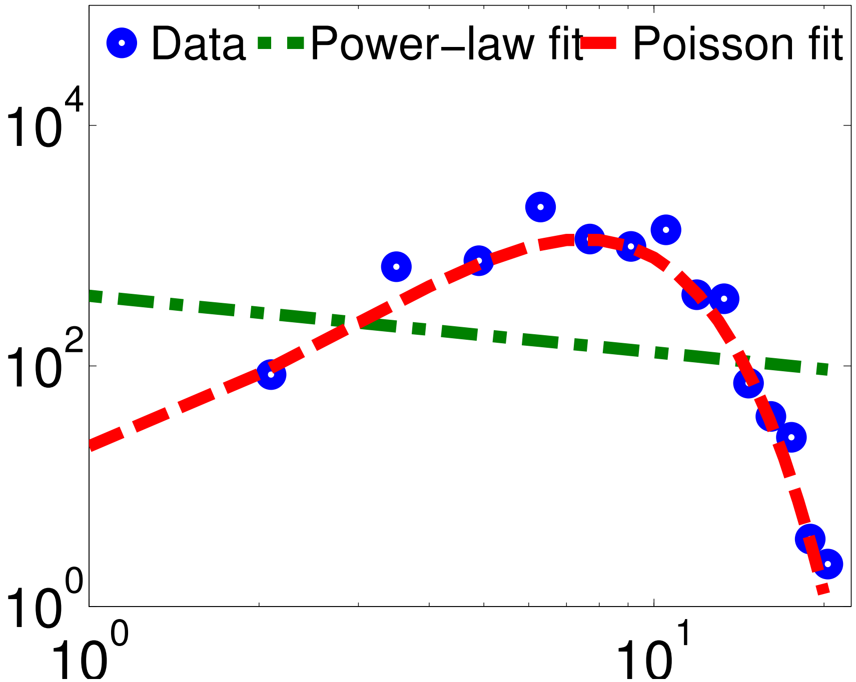
The model beats the predictions given by a standard Hawkes process

Network properties & cascade patterns

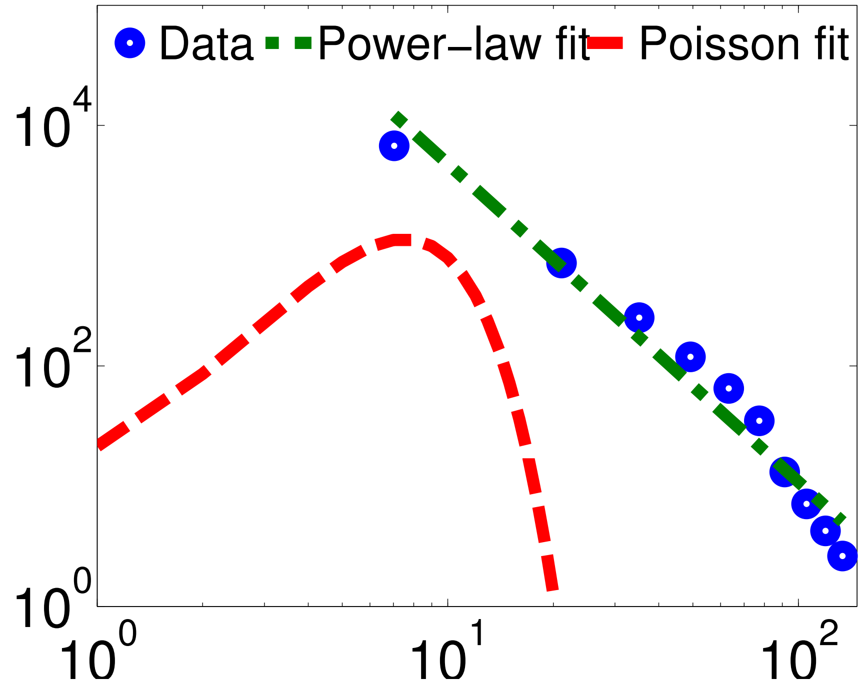
Can the model generate **realistic macroscopic static and temporal network patterns and information cascades?**

Network	Cascade s
Degree distributions	Cascade size distribution
Network diameter	Cascade depth distribution
Level of triadic closure	Cascade structure

Degree distributions



Poisson, α = 0

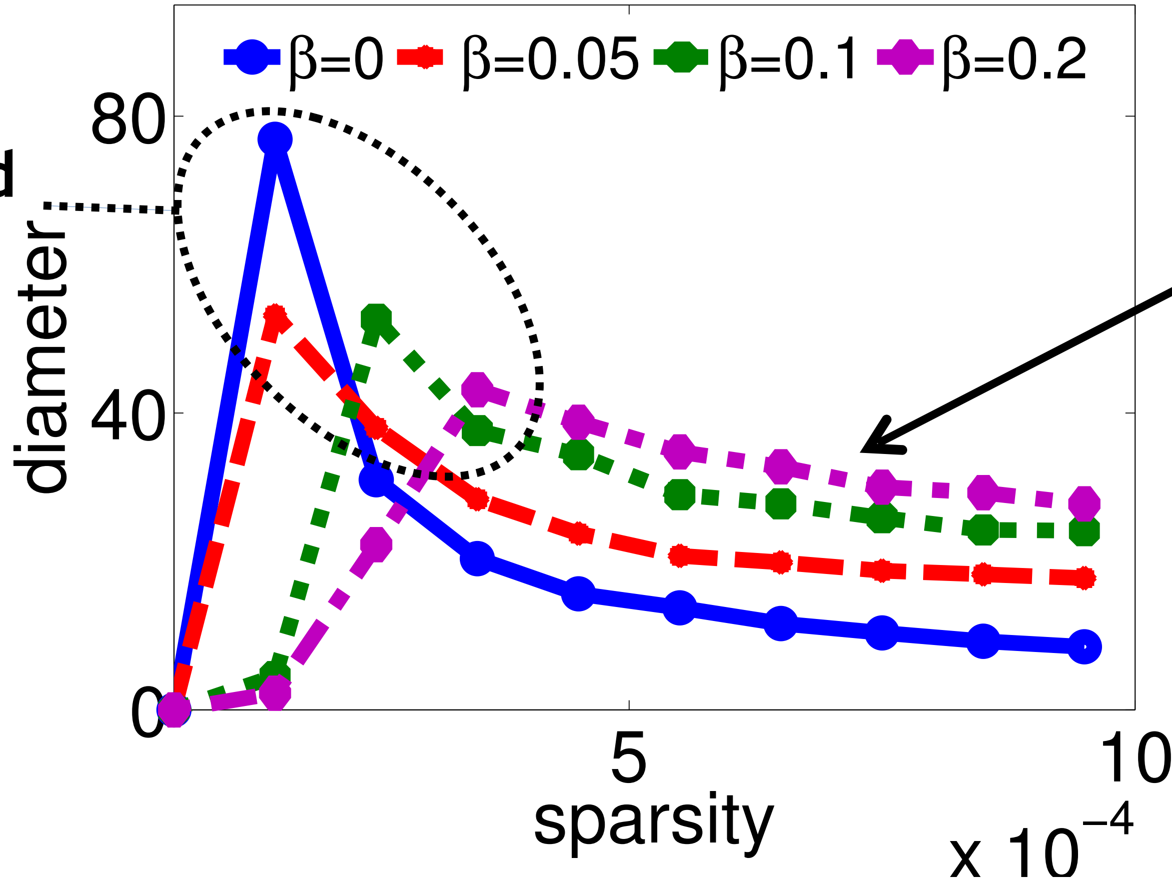


Power-law, α = 0.2

The higher the parameter α (or β), the closer the degree distribution is to a power-law

Small (shrinking) diameters

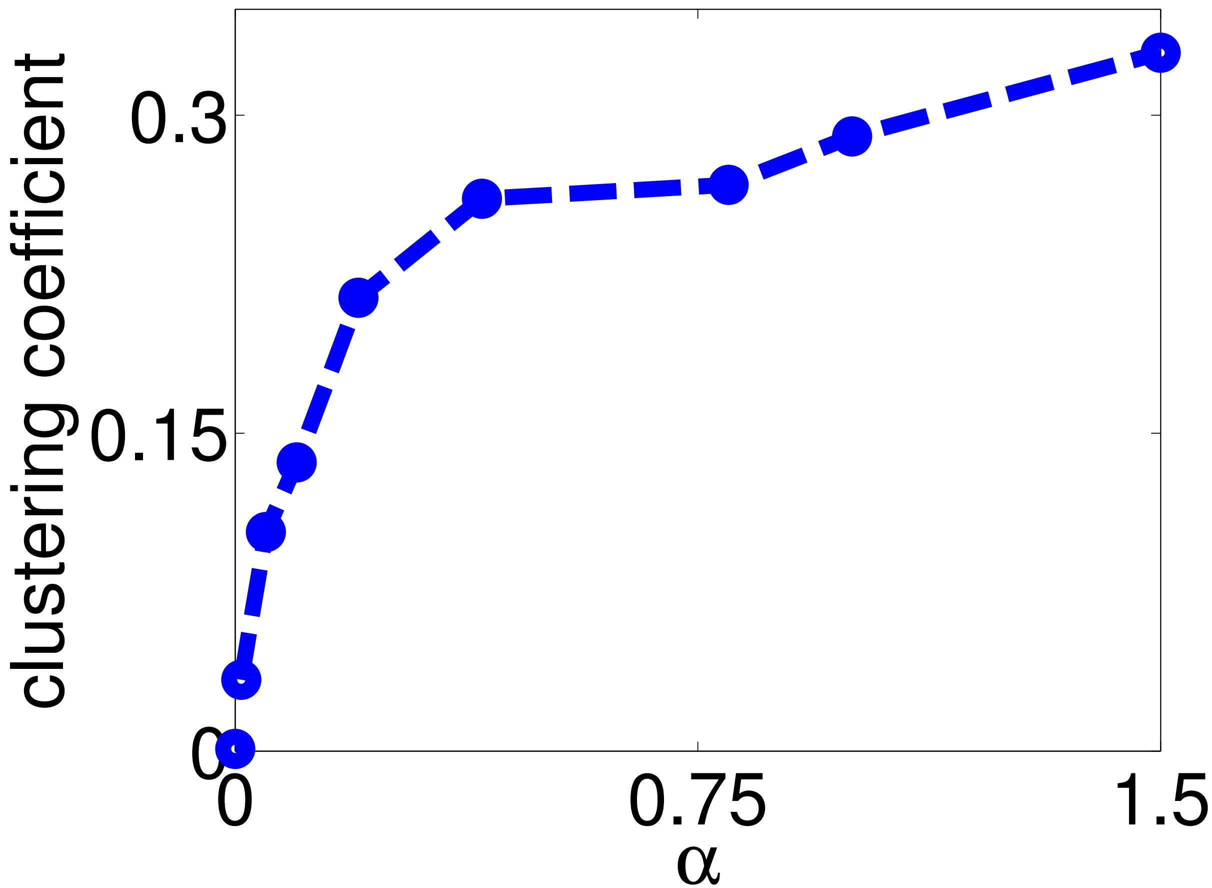
Small connected components merge



Diameter shrinks

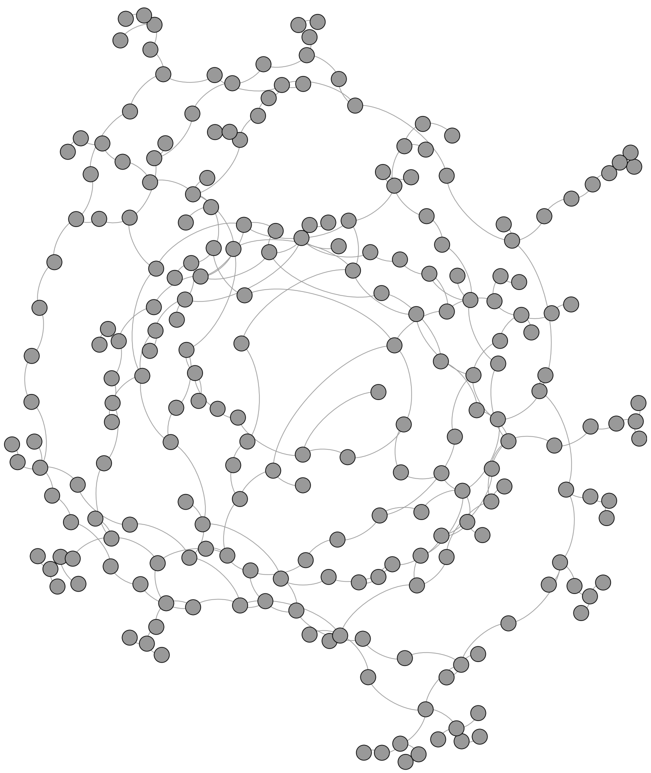
Our model generate networks with **small shrinking (or flattening) diameter over time**, as observed empirically.

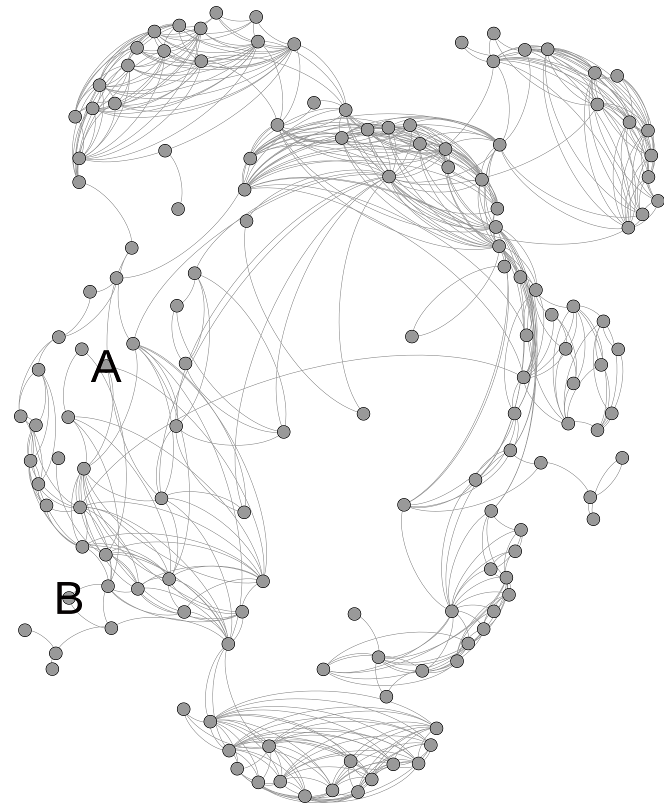
Clustering coefficient



We can generate networks with **different levels of triadic closure**, as observed empirically

Different type of networks



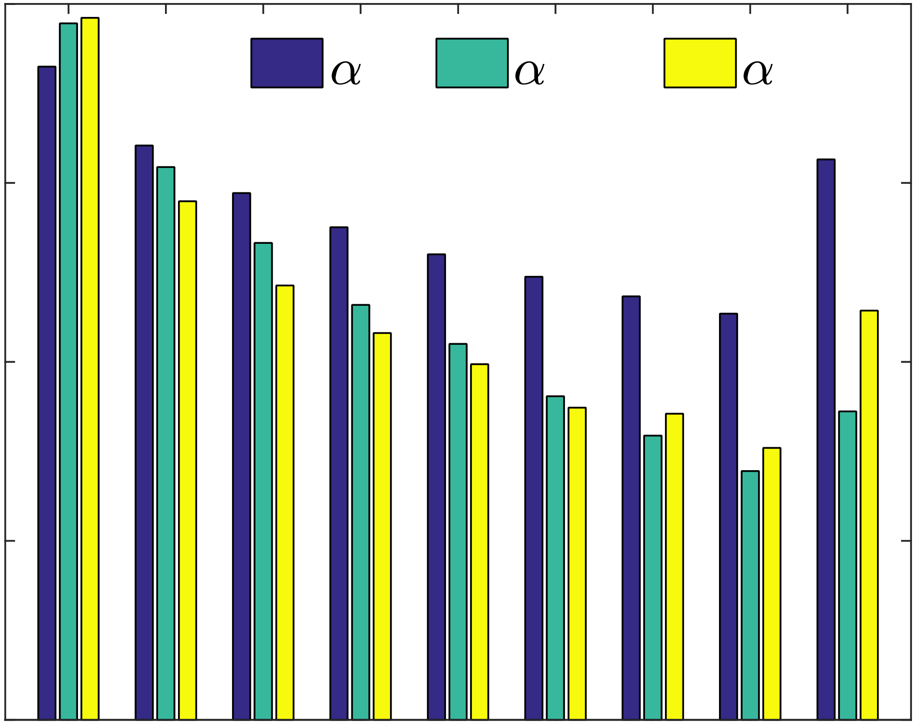


Erdos-Renyi, β = 0

Scale-free network, β = 0.8

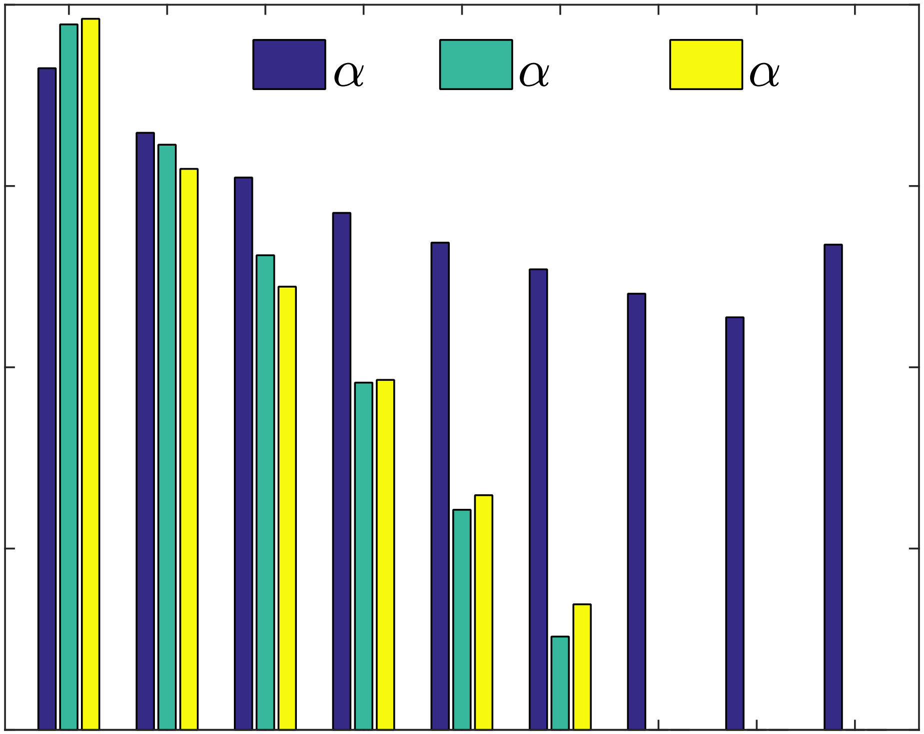
Our model allows us to generate **networks with very different structure**

Cascade patterns: size



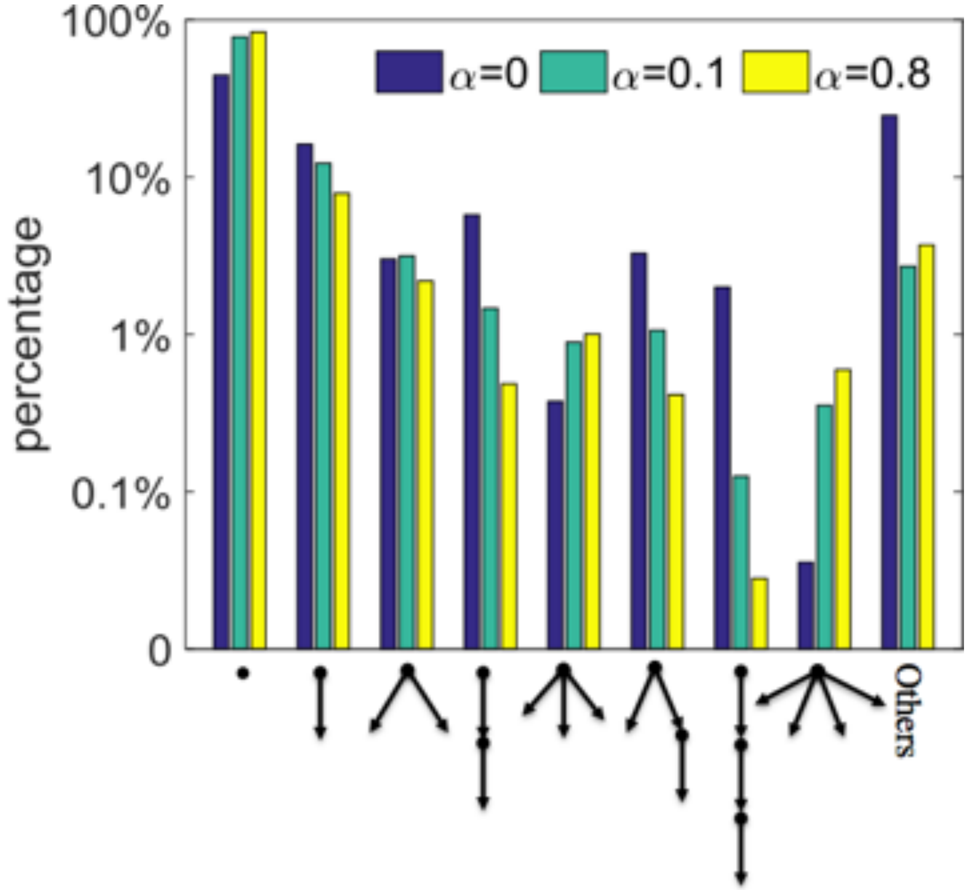
As α (or β) increases, **longer cascades become more seldom.**

Cascade patterns: depth



As α (or β) increases, **deeper cascades are more seldom**, as observed in real cascade data.

Cascade patterns: structure

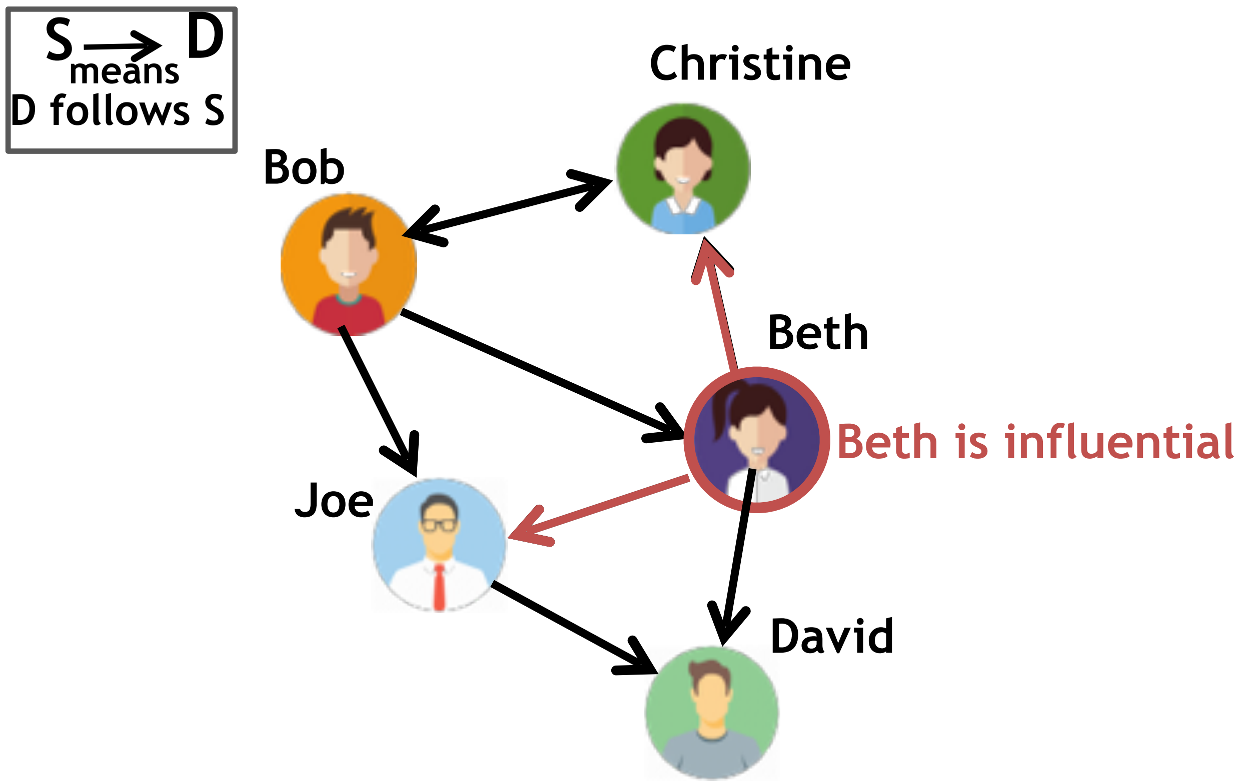


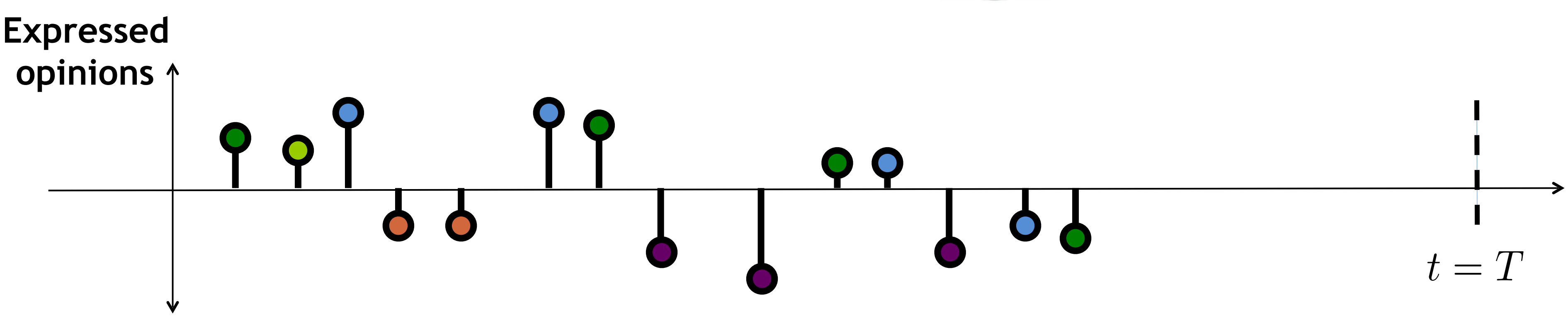
The **structure** of the generated cascades becomes ***more realistic*** **as α (or β) increases.**

Applications: Models

1. Information propagation
2. Opinion dynamics
3. Information reliability
4. Knowledge acquisition

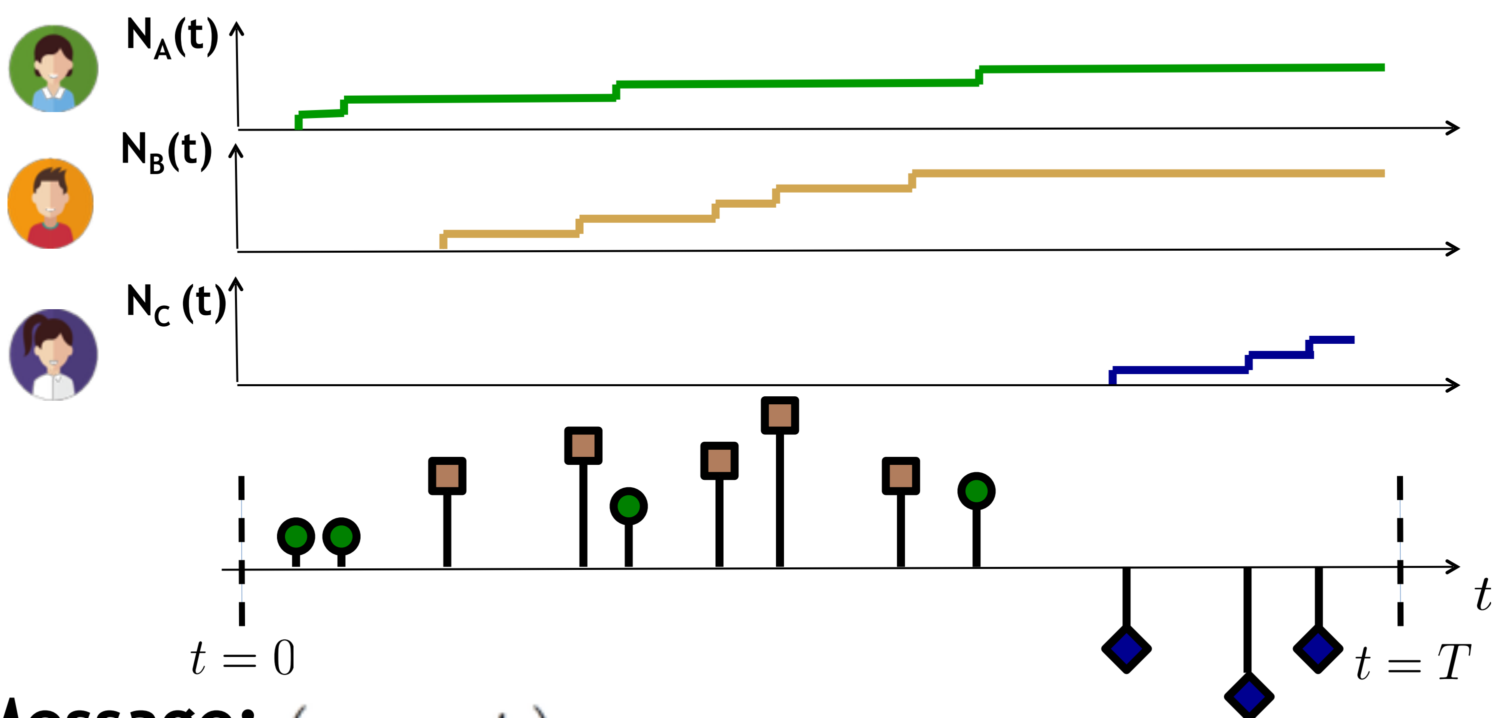
Opinion dynamics: an example





Message representation

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



Message: (u_i, m_i, t_i)

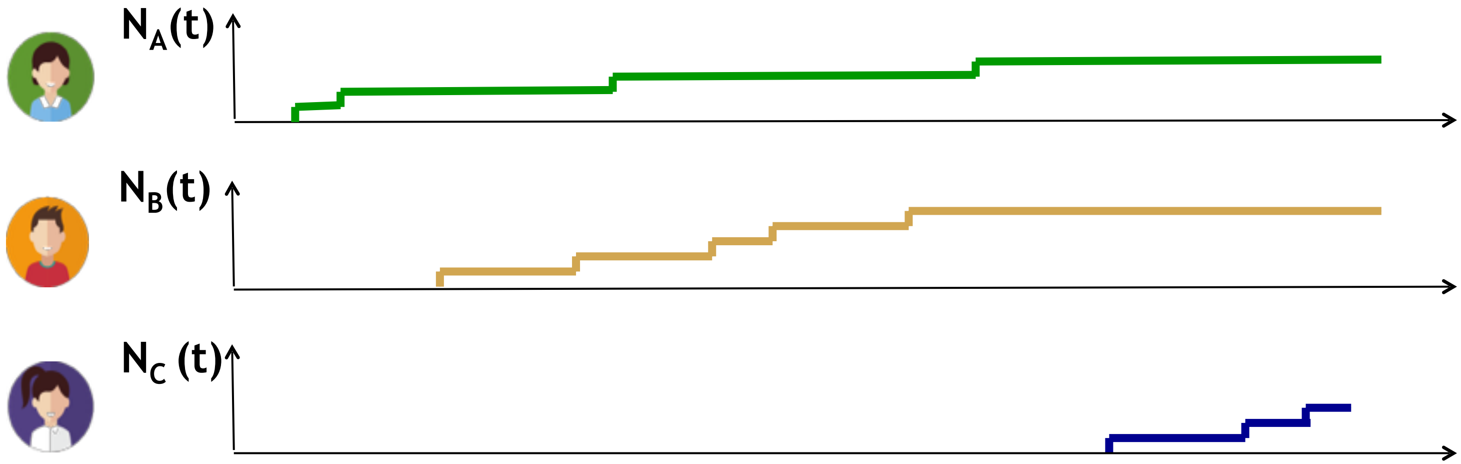
User

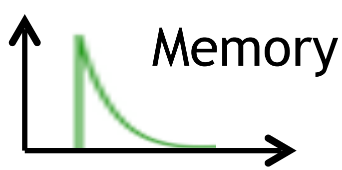
Sentiment (mark) → Noisy observation of latent opinion

Time

[De et al., NIPS 2016]

Message intensity





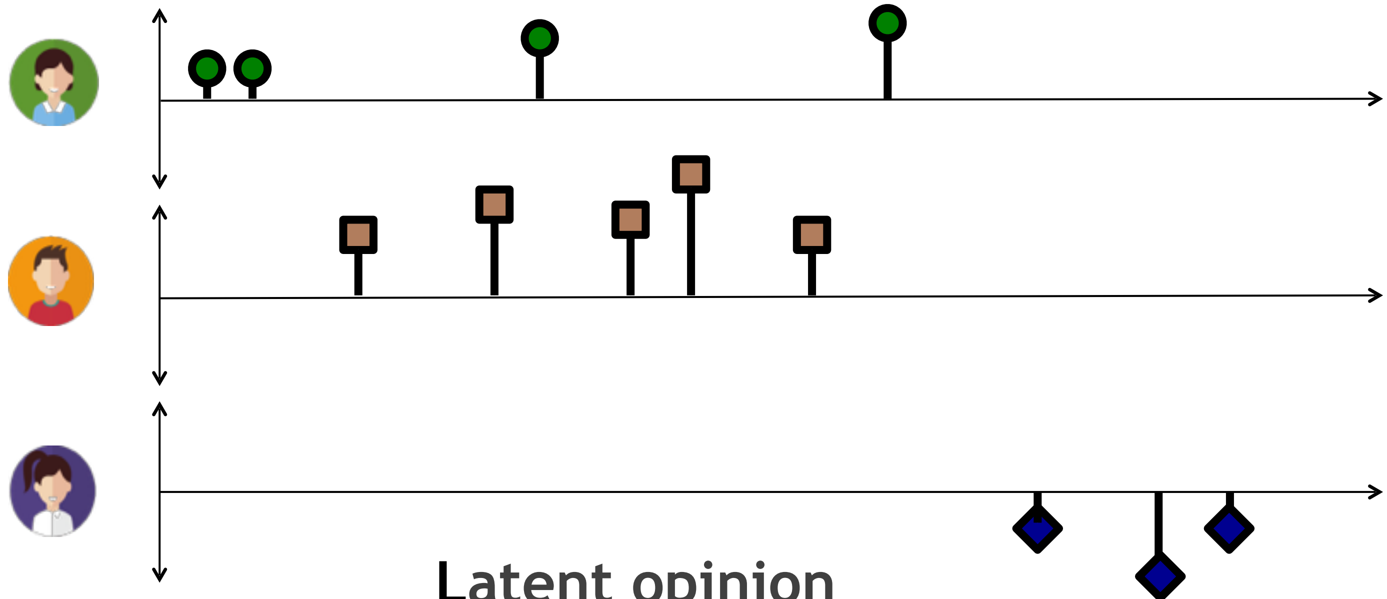
$$\underbrace{\lambda_u^*(t)}_{\text{User's intensity}} = \underbrace{\mu_u}_{\text{Messages on her own initiative}} + \sum_{v\in u\cup\mathcal{N}(u)} b_{vu} \underbrace{\sum_{e_i\in\mathcal{H}_v(t)} \kappa(t-t_i)}_{\text{Previous messages by user v}}$$

Influence from user v on user u

Hawkes process

[De et al., NIPS 2016]

Sentiment distribution



Latent opinion

Sentiment: $m_u \sim p(m|x_u^*(t))$

It depends on the recorded data

Continuous (based on sentiment analysis):

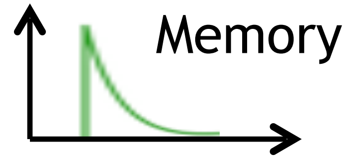
$$p(m|x_u^*(t)) = \mathcal{N}(x_u(t), \sigma_u)$$

Discrete (based on upvotes/

$$p(m|x_u^*(t)) = 1/(1 + \exp(-m \cdot x_u(t)))$$

[De et al., NIPS 2016]

Stochastic process for (latent) opinions



$$\underbrace{x_u^*(t)}_{\text{User's latent opinion}} = \underbrace{\alpha_u}_{\text{User's initial opinion}} + \sum_{v\in\mathcal{N}(u)} \underset{\text{influence from user v on user u}}{a_{vu}} \underbrace{\sum_{e_i\in\mathcal{H}_v(t)} m_i g(t-t_i)}_{\text{Previous sentiment by user v}}$$

[De et al., NIPS 2016]

Opinion model as Jump SDEs

Proposition. The tuple ($x^*(t)$, $\lambda^*(t)$, N(t)) is a **Markov process**, whose dynamics are defined by the following **marked jumped stochastic differential equations** (SDEs)

Latent opinions

Network!

Informational influence

Expressed opinions

$$d\mathbf{x}^*(t) = \omega(\boldsymbol{\alpha} - \mathbf{x}^*(t))dt + \mathbf{A}(\mathbf{m}(t) \odot d\boldsymbol{N}(t))$$

$$d\boldsymbol{\lambda}^*(t) = \nu(\boldsymbol{\mu} - \boldsymbol{\lambda}^*(t))dt + \mathbf{B}\, d\boldsymbol{N}(t)$$

Message intensitie s

Temporal influence

Network!

[De et al., NIPS 2016]

Model inference from opinion data

Events likelihood

$$\underbrace{\sum_{e_i \in \mathcal{H}(T)} \log p(m_i | x^*_{u_i}(t_i))}_{\text{Message sentiments (marks)}} + \underbrace{\sum_{e_i \in \mathcal{H}(T)} \log \lambda^*_{u_i}(t_i) - \sum_{u \in \mathcal{V}} \int_0^T \lambda^*_u(\tau)\, d\tau}_{\text{Message times}}$$

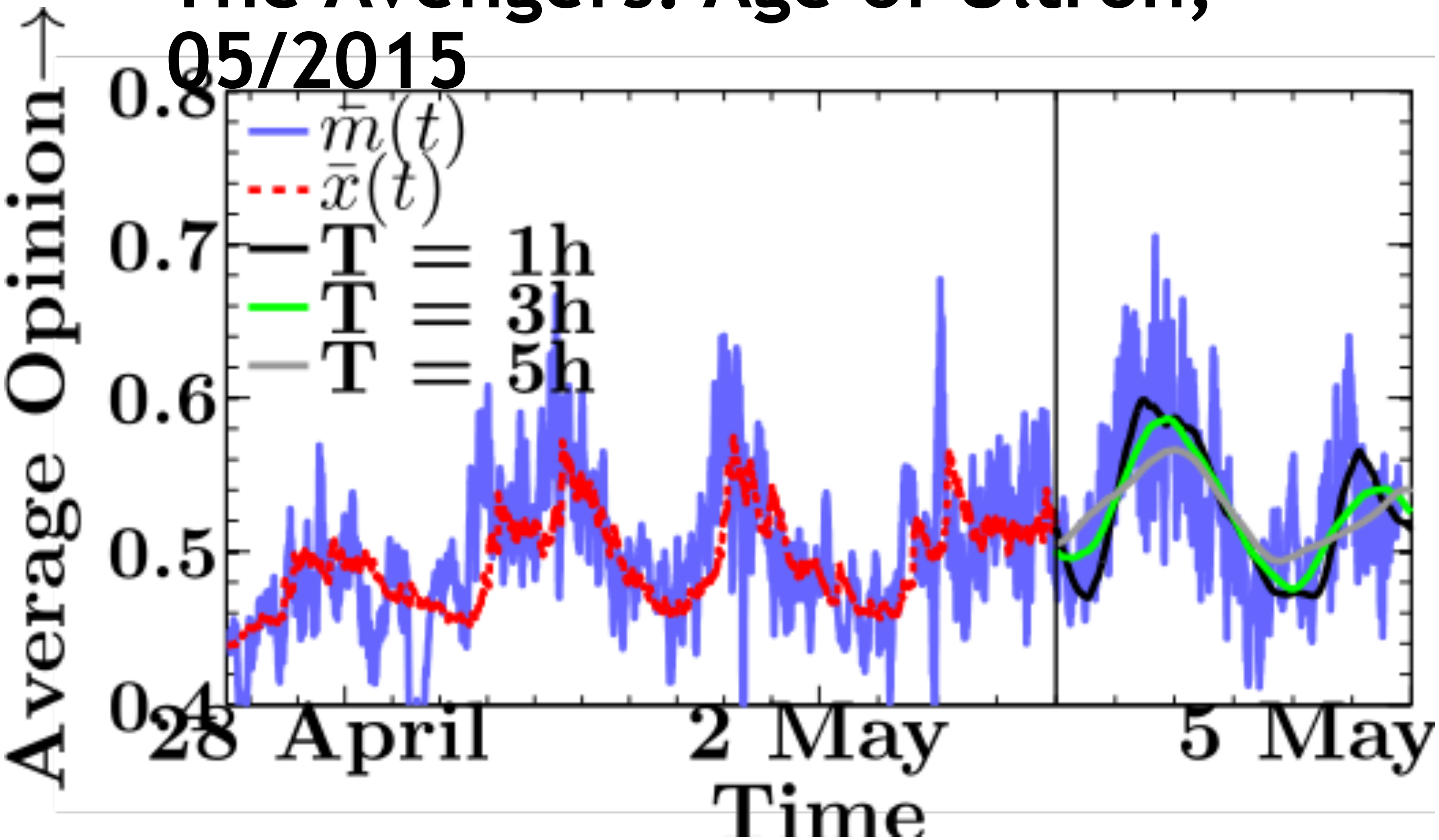
Theorem. The **maximum likelihood** problem **is convex in the model parameters.**

Markov property → **Sums and integrals in linear time!**

[De et al., NIPS 2016]

Opinion forecasting

The Avengers: Age of Ultron, 05/2015



The forecasted opinion becomes less accurate as T increases, as one may expect.

[De et al., NIPS 2016]

Applications: Models

1. Information propagation
2. Opinion dynamics
3. Information reliability
4. Knowledge acquisition

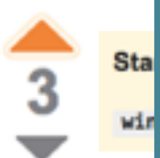
Information reliability: an example

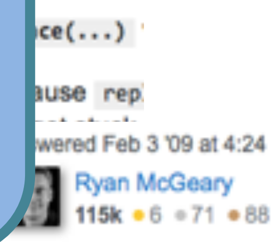
Learning from the crowd ('crowdlearning') has become very popular:

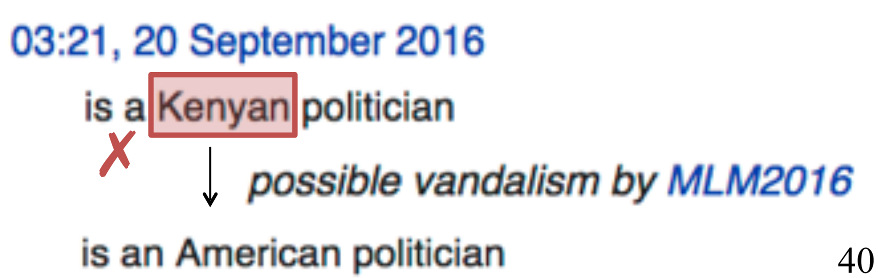
Users learn from the k

Knowledge is reviewed
an
it

Refutations and verifications depend on the source trustworthiness



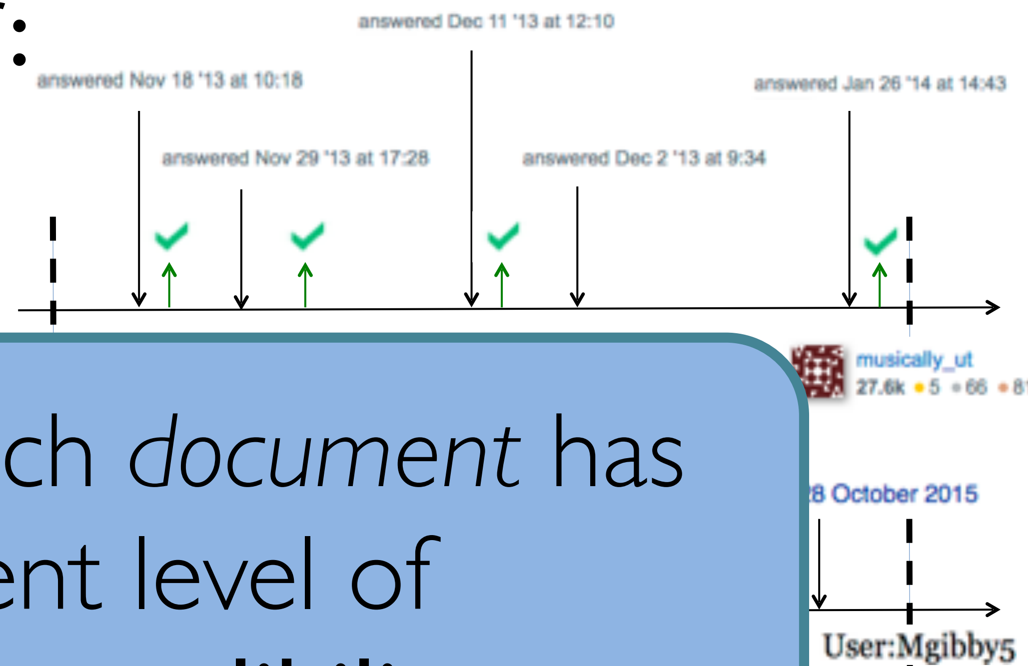




Information reliability: key, simple idea

A **source is trustworthy if:**

Its contributions are
verified more
f



Its c
refu

Over time, each *document* has
a different level of
***inherent* unrealibility**

Challenge

At a time t, a ***document* may be *disputed***

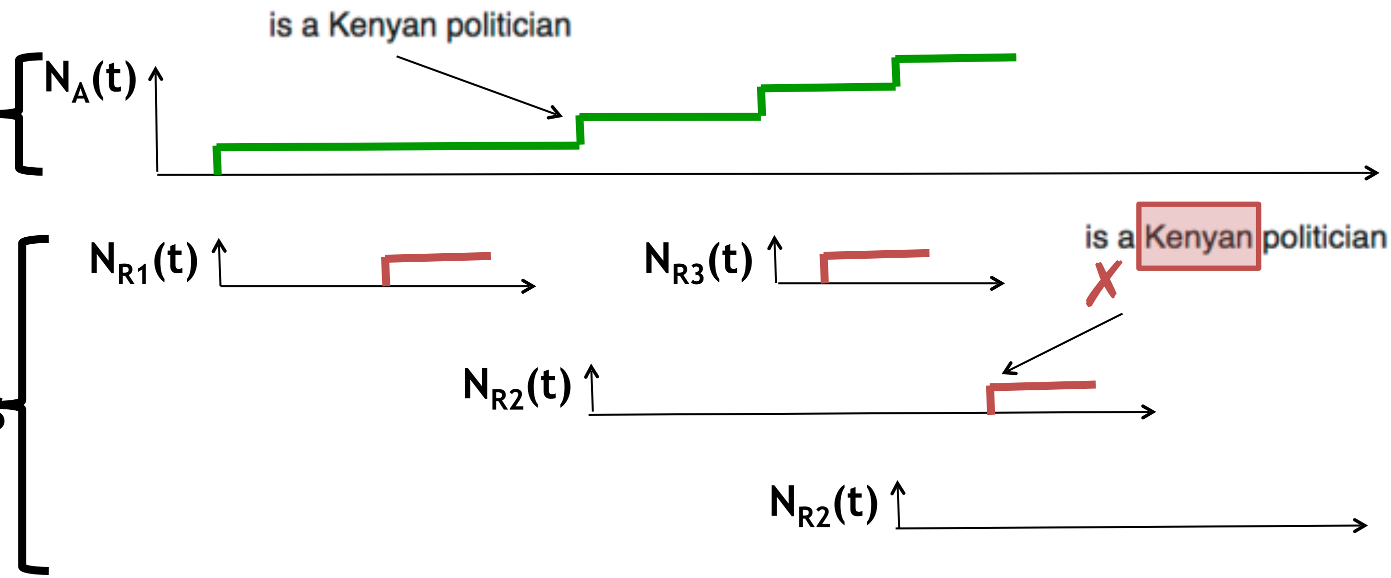
Verifications: rarer
Refutations: more frequent

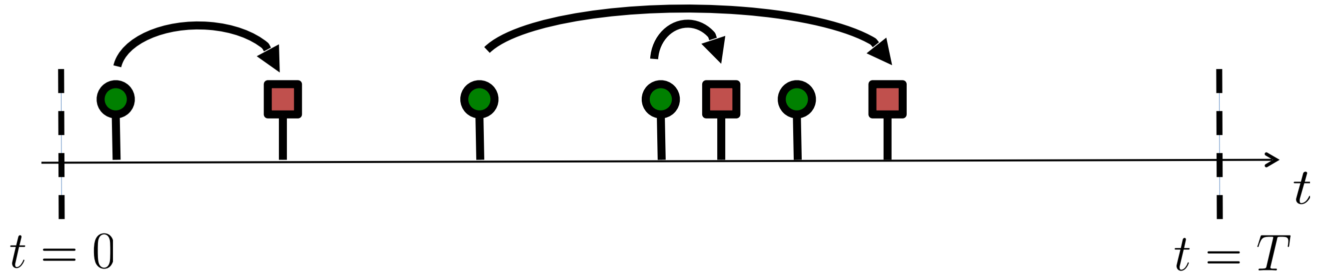
[Tabibian et al., WWW 2017]

Representation: temporal point processes

Statement additions
(one process per document)

Statement refutations
(one process per statement)





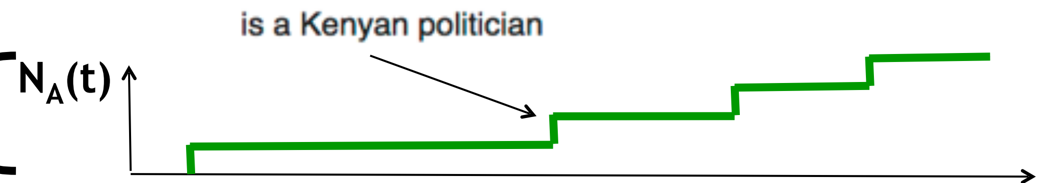
source — refutation time — addition time

Statement: $e = (s, \; t, \; \tau)$

[Tabibian et al., WWW 2017]

Intensity of statement additions

Statement additions
(one process per article)



$$\lambda_d(t) = \underbrace{\sum_j \phi_{d,j} k(t - t_j)}_{} + \underbrace{\sum_{e_i \in \mathcal{H}_d(t)} \mathbf{w}_d^\top \gamma_{s_i} g(t - \tau_i)}_{}$$

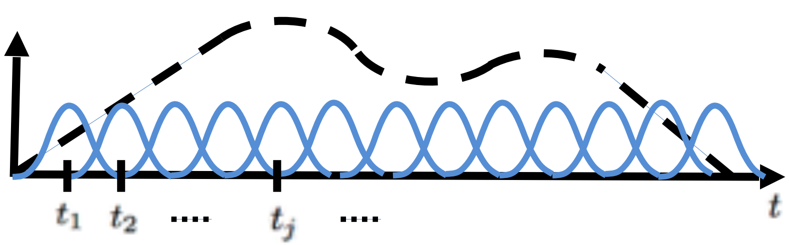
Intensity or rate
(Statements per time unit)

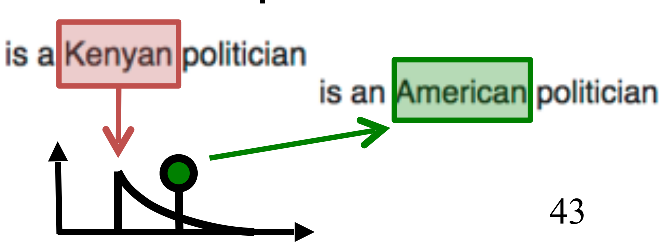
Article unreliability
(Mixture of Gaussians)

Effect of past refutations
(topic dependent; topic weight w_d)

Temporal evolution of the *intrinsic* reliability of the article

Refuted statements trigger the arrival of new statements to replace them



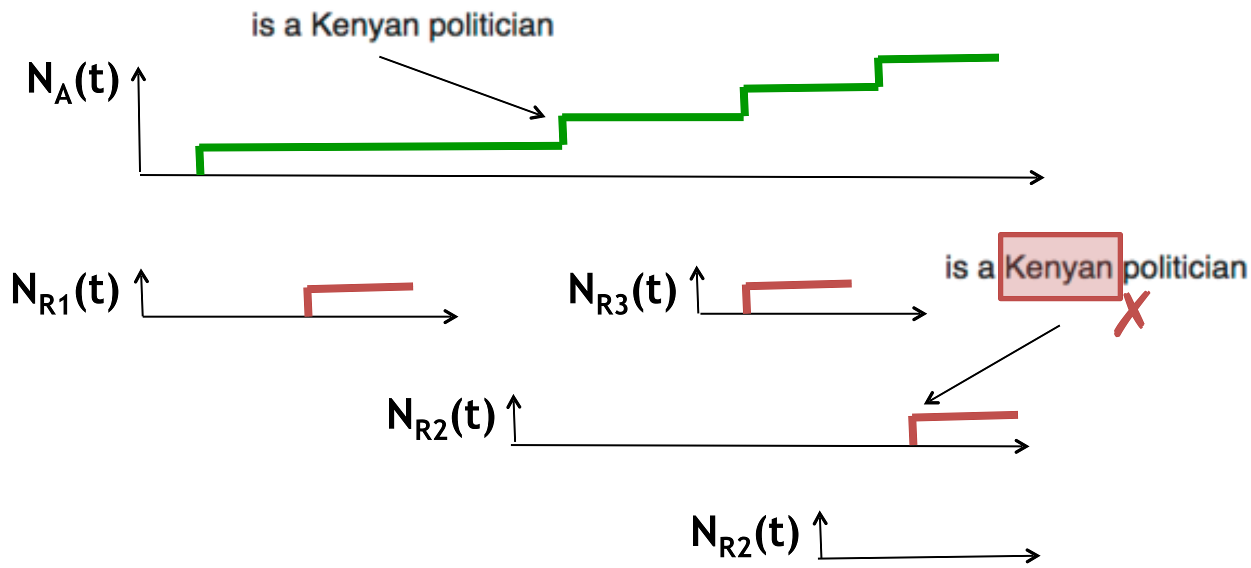


[Tabibian et al., WWW 2017]

Intensity of statement refutations

Statement additions
(one process per article)

Statement refutations
(one process per statement)



$$\mu_i(t) = \underbrace{(1 - N_i(t))}_{\text{Refutations happen only once}} \Big[\underbrace{\sum_j \beta_{d,j} k(t + t_i - t_j)}_{\text{Article unreliability}} + \underbrace{\mathbf{w}_d^\top \boldsymbol{\alpha}_{s_i}}_{\text{Source trustworthiness}} \Big]$$

Intensity or rate
(Statements per time unit)

Refutations happen only once

Article unreliability
(Mixture of Gaussians)

Shared across statements of an article!

Source trustworthiness
(topic dependent; topic weight w_d)

The higher the parameter α_{s_i}, the quickest an article gets refute

[Tabibian et al., WWW 2017]

Model inference from event data

Conditional intensities

$$\{\lambda_d(t)\} \quad \{\mu_i(t)\}$$

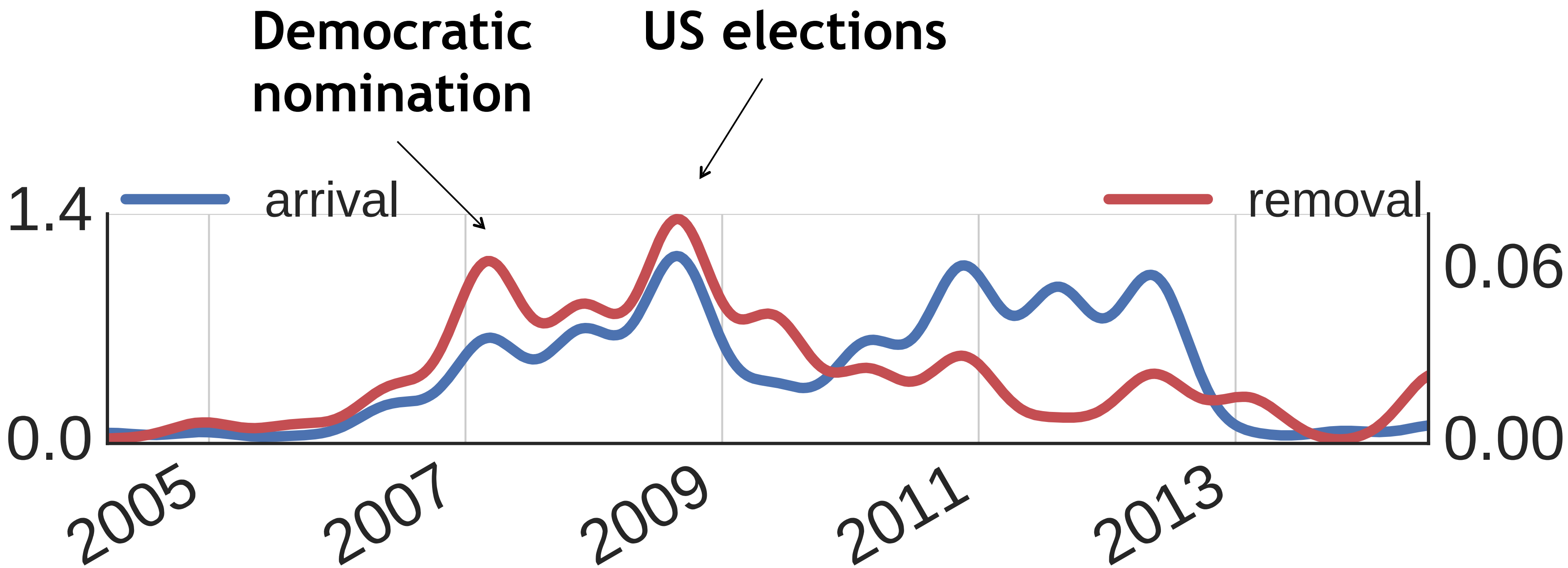
Events

$$\sum_{d=1}^{|\mathcal{D}|} \sum_{i:e_i \in \mathcal{H}_d(T)} \log \underbrace{p(t_i | \mathcal{H}_d(t_i), \boldsymbol{\phi}_d, \{\boldsymbol{\gamma}_s\}, \mathbf{w}_d)}_{\text{statements additions}} + \sum_{d=1}^{|\mathcal{D}|} \sum_{i:e_i \in \mathcal{H}_d(T)} \log \underbrace{p(\Delta_i | t_i, \boldsymbol{\beta}_d, \{\boldsymbol{\alpha}_s\}, \mathbf{w}_d)}_{\text{statements evaluations}}$$

Theorem. The **maximum likelihood** problem **is convex in the model parameters.**

[Tabibian et al., WWW 2017]

Wikipedia article reliability

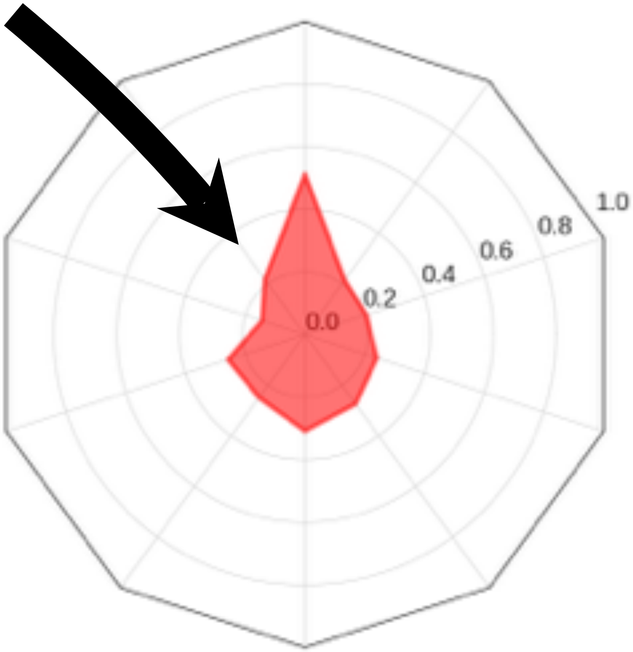


Barack Obama's Wikipedia Article
(Arrival of information vs intrinsic unreliability)

[Tabibian et al., WWW 2017]

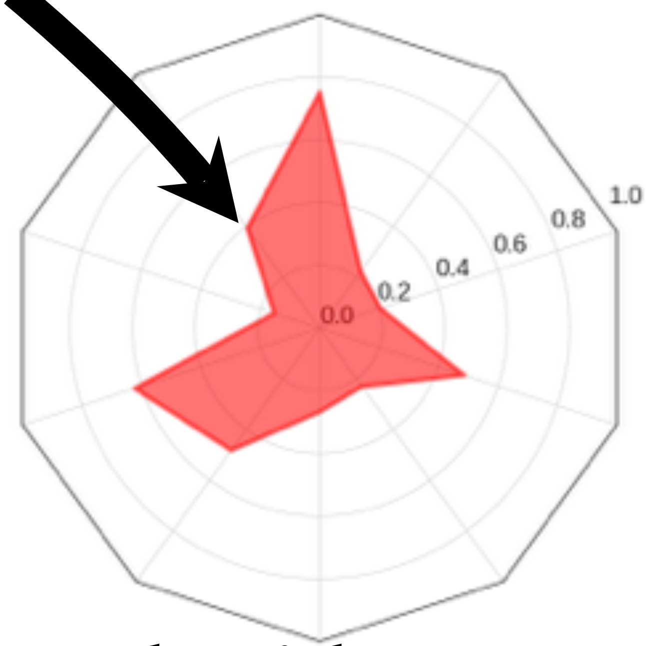
Source trustworthiness

Politics



bbc.co.uk

Politics



breitbart.co
m

Probability of refutation within 6 months in a *stable* Wikipedia article

[Tabibian et al., WWW 2017]

Applications:
Models

1. Information propagation
2. Opinion dynamics
3. Information reliability
4. Knowledge acquisition

Crowdlearning: Learning from the crowd

Learning from the crowd ('crowdlearning') has become very popular in the Web and social media

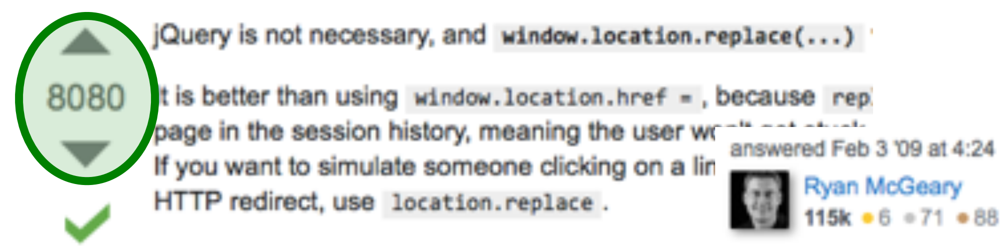
StackExchange

Quora

stackoverflow

Common Functionality:

Users learn from the knowledge other users contribute



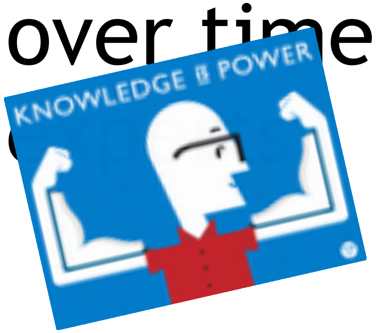
Knowledge is reviewed by users using assessment measures such as upvotes or likes

Our Goal: Uncover Learning and Knowledge

Can we **design a data-driven model c crowdlearning?**

Why this goal?

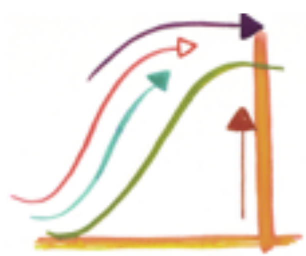
Understand **how people learn**
over time and become



Identify **questions and answers** with **high knowledge value**

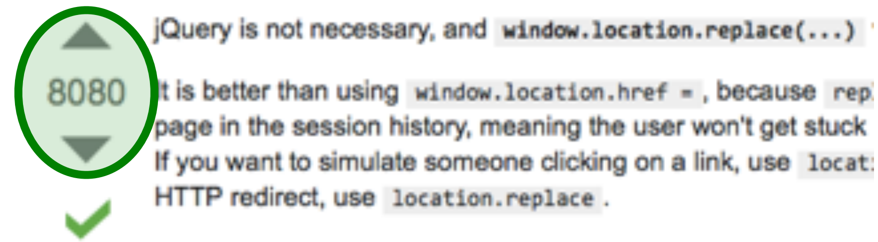
Key Idea

Every time a **user learns** from a **contribution** by other users



She may increase her **expertise**

upvotes from users



Her **contributions** become **more knowledgeable** and **assessed** more **highly**

[Upadhyay et al., WSDM 2017]

Learning and contributing events

A knowledge item
(smallest quantum of knowledge)

5 votes | 3 answers | 818 views — How to define private constructors in javascript? — javascript

A learning event
(user learns from a knowledge item)

$$(u, t, q)$$

user

time

asked Feb 10 '14 at 0:28
musically_ut
20.9k • 3 • 49 • 65

knowledge item

5 votes | 3 answers | 818 views — How to define private constructors in javascript? — javascript

A contributing event
(user contributes to a knowledge item)

$$(u, t, q, s)$$

user

time

answered Feb 10 '14 at 0:28
musically_ut
20.9k • 3 • 49 • 65

knowledge item

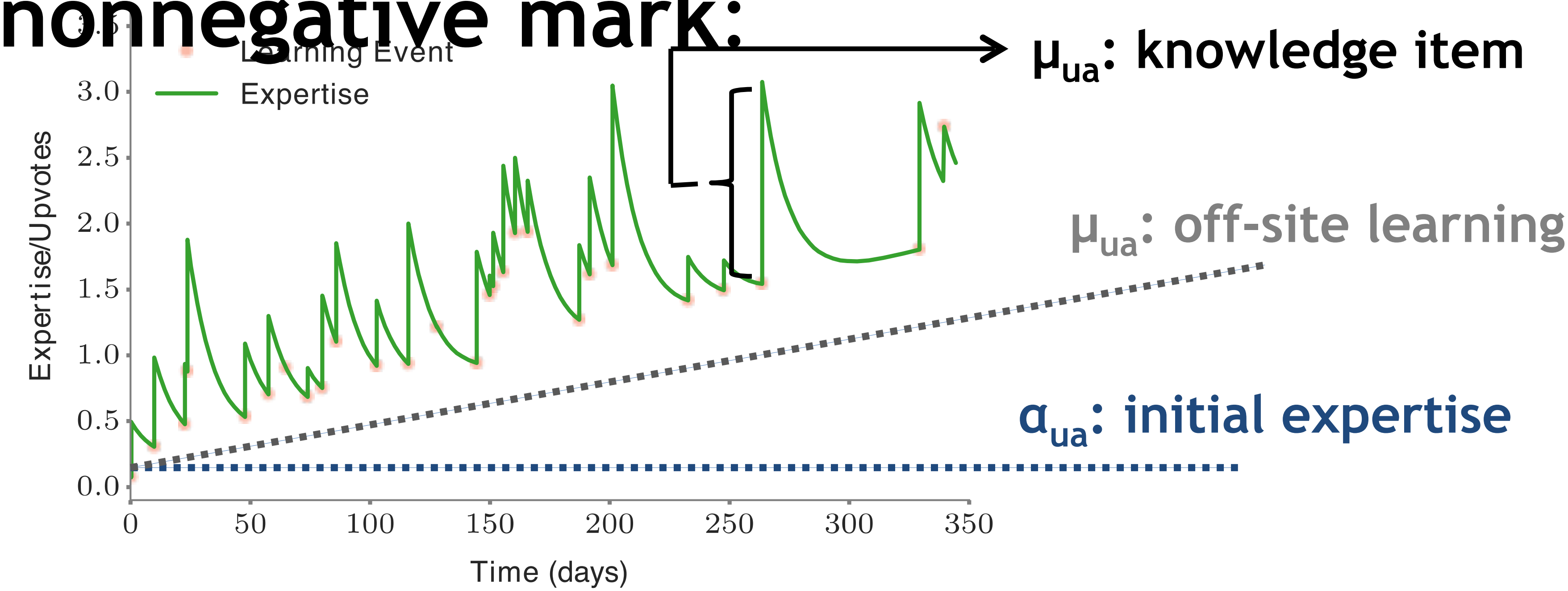
1435 votes — Q: Replacing all occurrences of a string in JavaScript — 26 answers — javascript

score 2

[Upadhyay et al., WSDM 2017]

Stochastic process for expertise (I)

Expertise of a **user u** on a **topic a** at **time t** as
real nonnegative mark:

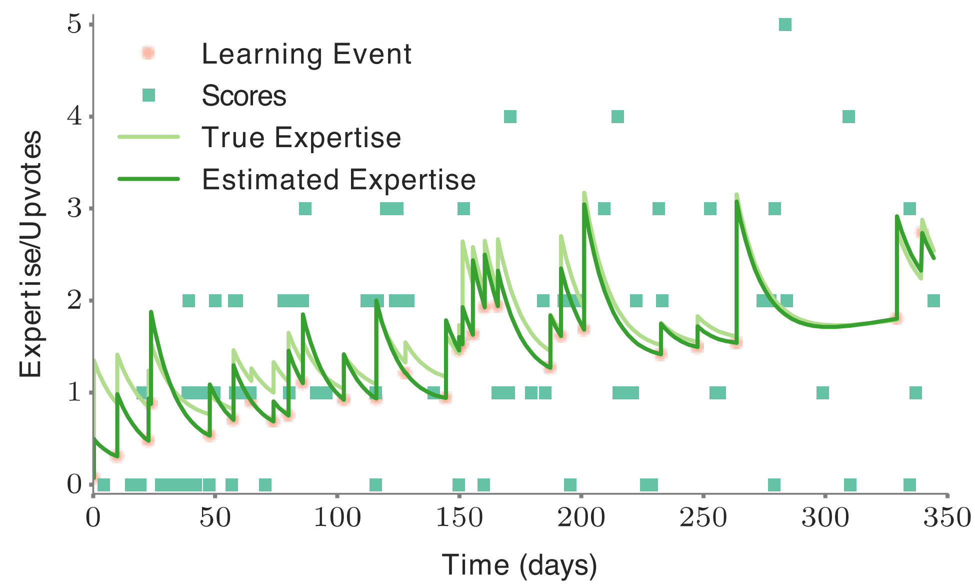


Expertise: $$e^*_{ua}(t) := \overbrace{\alpha_{ua}}^{\text{initial expertise}} + \underbrace{\mu_{ua} \cdot t}_{\text{off-site learning}} + \overbrace{\sum_{i:q_i \in \mathcal{H}^l_u(t)} k_{q_i a} \cdot \underbrace{\kappa_\omega (t - t_i)}_{\textbf{forgetting}}}^{\text{on-site learning}}$$

[Upadhyay et al., WSDM 2017]

Stochastic process for expertise (II)

Score a user's contribution receiv **at time t** depends o her expertise e^*_{ua}(t



Key Idea

Use recorded scores to uncover:

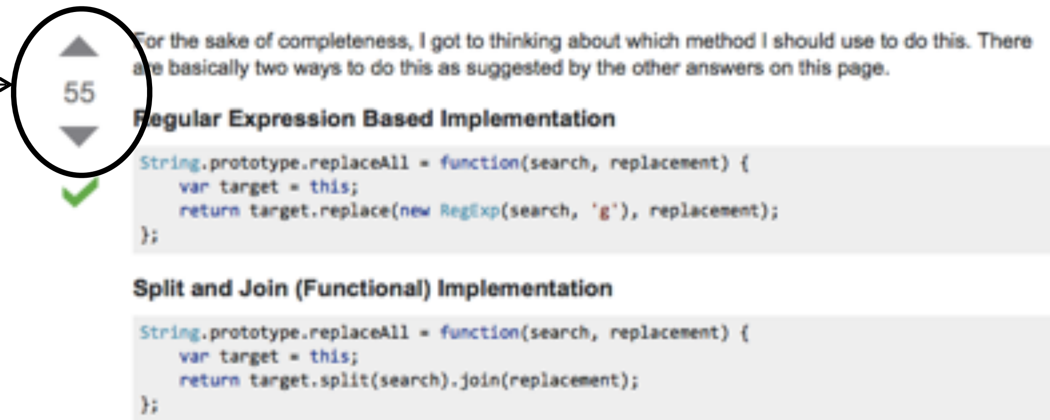
- Her expertise $e^*_{ua}(t)$
- Knowledge values k_{qa} of the items she learns

[Upadhyay et al., WSDM 2017]

Score distribution

The particular **choice** of score distribution **depends on the recorded data**, e.g.,

55 people **learned** from this answer


For the sake of completeness, I got to thinking about which method I should use to do this. There are basically two ways to do this as suggested by the other answers on this page.

Regular Expression Based Implementation

```
String.prototype.replaceAll = function(search, replacement) {
    var target = this;
    return target.replace(new RegExp(search, 'g'), replacement);
};
```

Split and Join (Functional) Implementation

```
String.prototype.replaceAll = function(search, replacement) {
    var target = this;
    return target.split(search).join(replacement);
};
```

Discrete non-negative scores

Contribution topics (tags)

$$p(s|\mathcal{A}_q, \boldsymbol{e}_u^*(t)) \sim \text{Poisson}\left(\frac{\boldsymbol{w}_q^T \boldsymbol{e}_u^*(t)}{\boldsymbol{w}_q^T \mathbf{1}}\right)$$

User's expertise per topic (or tag)

[Upadhyay et al., WSDM 2017]

Maximum Likelihood Estimation (MLE)

Given a collection of (million of) learning and contributing events:

$$\underbrace{\underset{\boldsymbol{\alpha}\geq 0,\boldsymbol{\mu}\geq 0,\boldsymbol{k}\geq 0}{\text{maximize}} \sum_{(u,t,q,s)\in\mathcal{H}^c(T)} s\cdot\log\left(\frac{\boldsymbol{w}_q^T\boldsymbol{e}_u^*(t)}{\boldsymbol{w}_q^T\mathbf{1}}\right) - \frac{\boldsymbol{w}_q^T\boldsymbol{e}_u^*(t)}{\boldsymbol{w}_q^T\mathbf{1}}}_{\textbf{Maximum likelihood estimation (MLE)}}$$

Initial expertise — Off-site learning — Knowledge items — Contributions

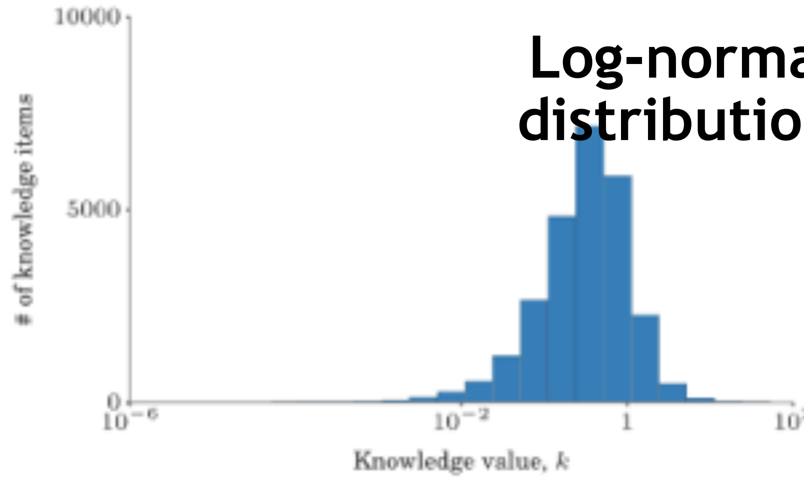
The problem is **jointly convex** on all parameters → Convex solver finds optimal parameters **efficiently**

[Upadhyay et al., WSDM 2017]

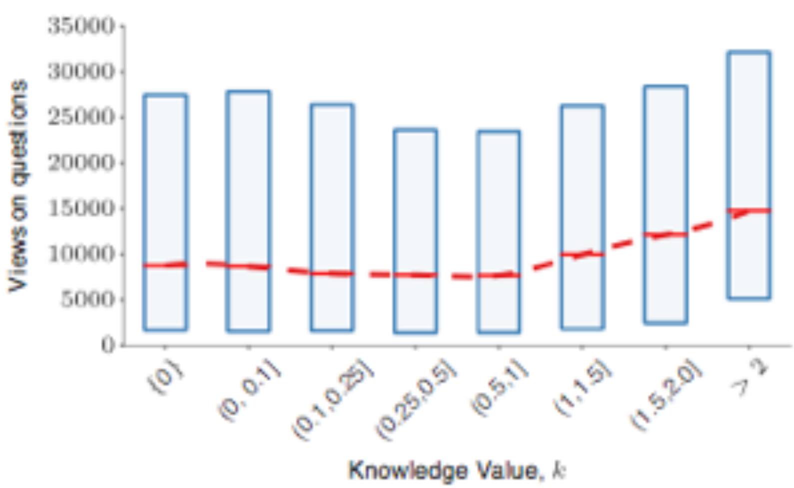
Distribution of knowledge

How is knowledge distributed?

10% of the items account for 60% of the knowledge!



Log-normal distribution



Does popularity mean high knowledge?

Not always - Two different knowledge regimes:

median k

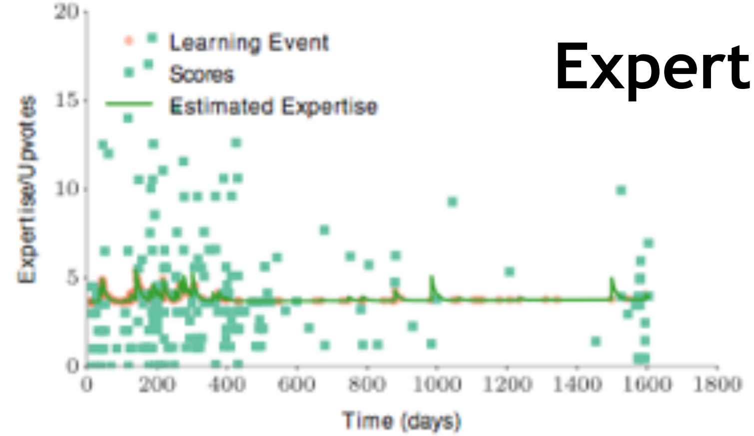
popularity is constant

it increases at a steady rat

[Upadhyay et al., WSDM 2017]

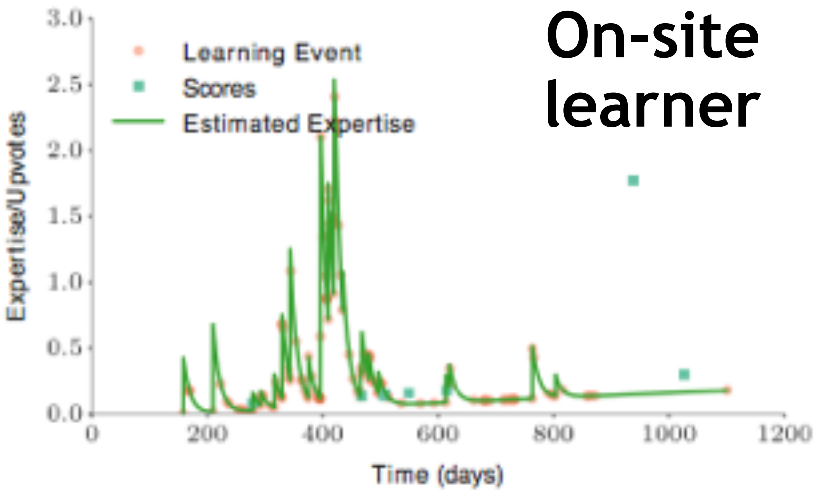
Types of learners: examples

Average learner

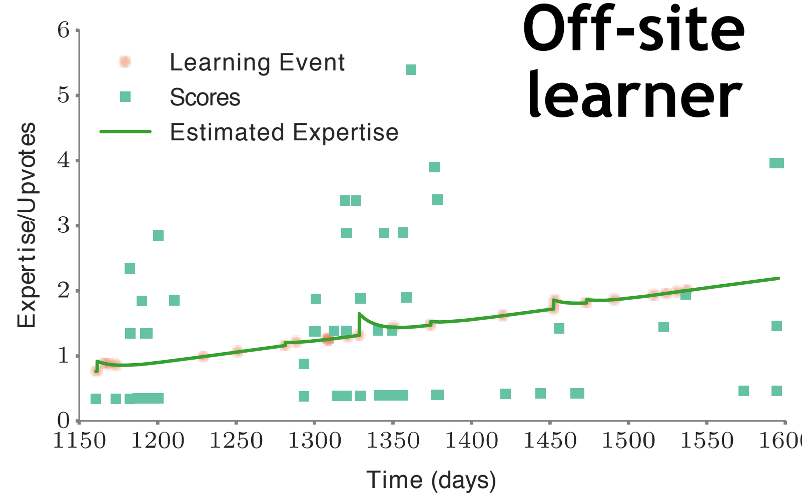


Expert

Avg. knowledge/contribution: 0.005 ⟷ Avg. knowledge/contribution: 0.034



On-site learner



Off-site learner

On-site learning 55% ⟷ On-site learning: 0.4%

[Upadhyay et al., WSDM 2017]

Interplay between on-site and off-site learning

Higher on-site learning, higher off-site learning

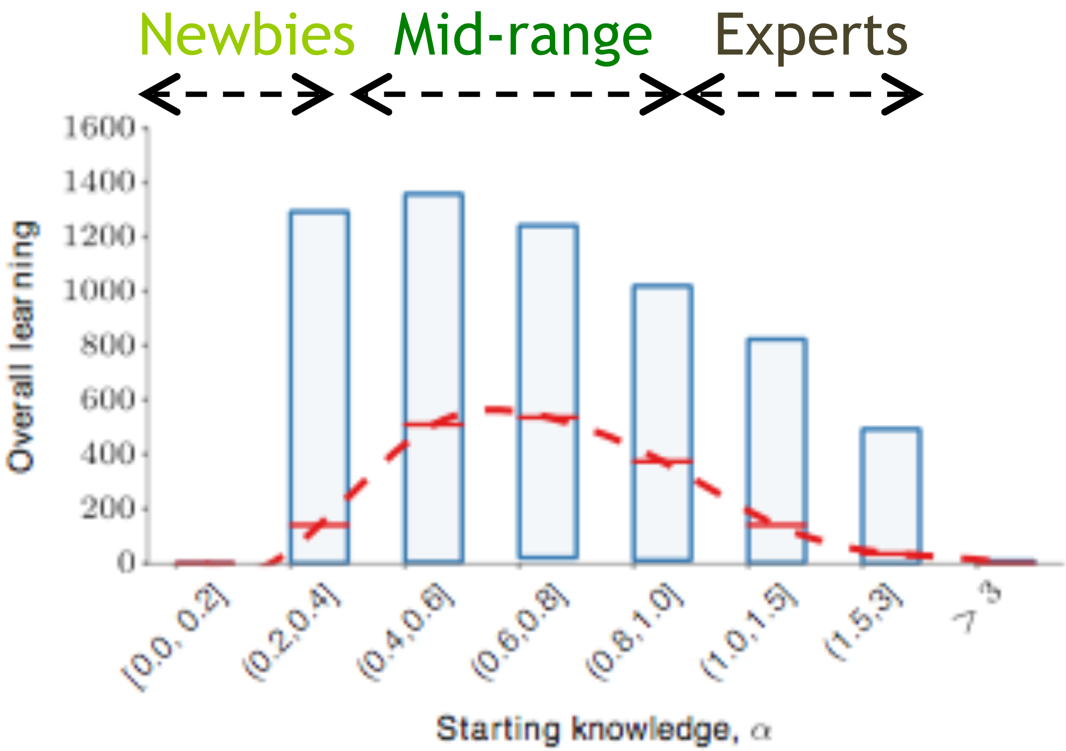
Off-site learning becomes more dominant

Quick learners rely less on on-site learning, in relative terms

Learning: aggregate number of up-votes users would have received if they were posting answers only due to either their on or off-site learning

[Upadhyay et al., WSDM 2017]

Which users learn the most?



Newbies and experts tend to increase their knowledge the least

Mid-range users tend to increase it the most

Coherent with previous research:

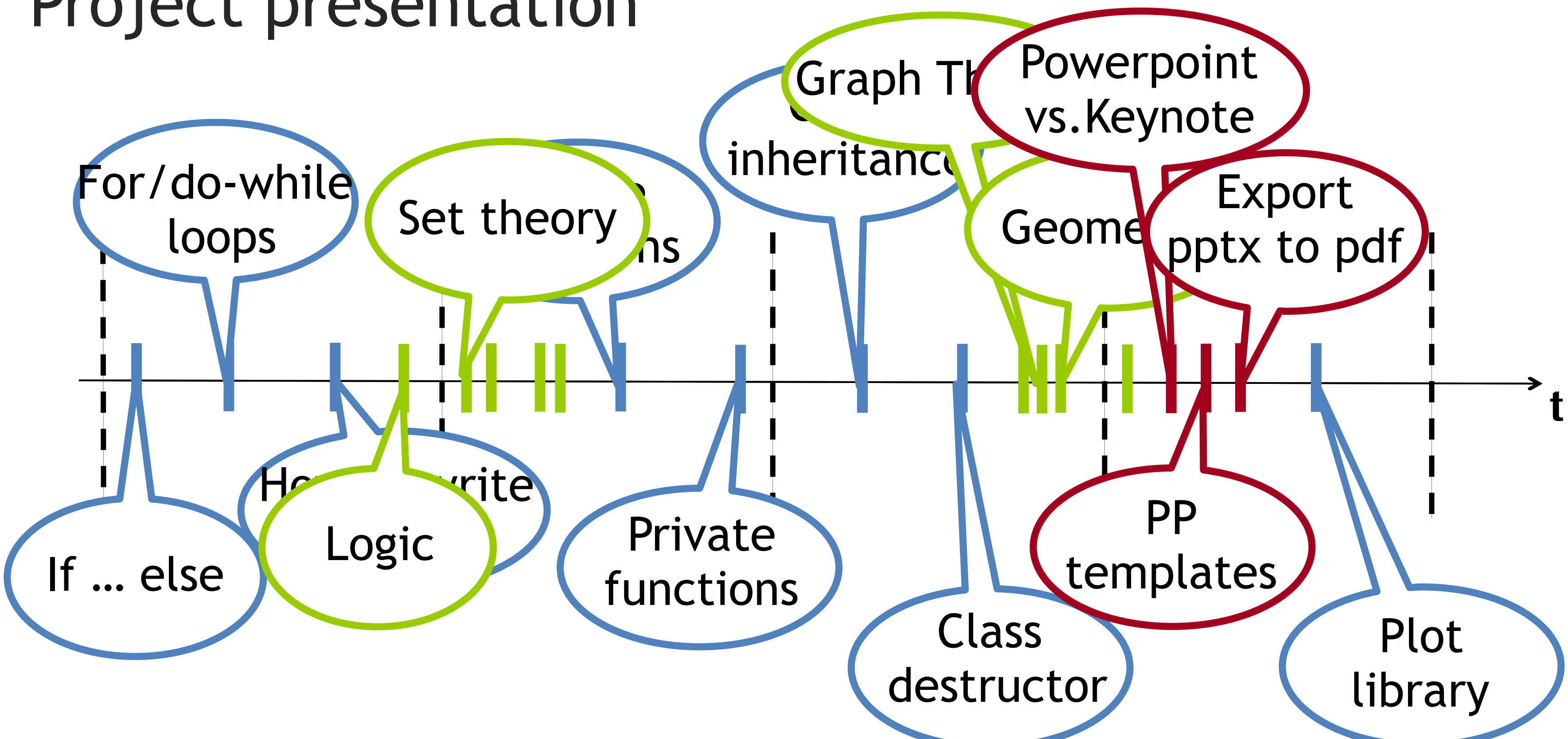
If there is only positive reinforcement

Gain in expertise: sigmoidal shape

[Upadhyay et al., WSDM 2017]

1st year computer science student

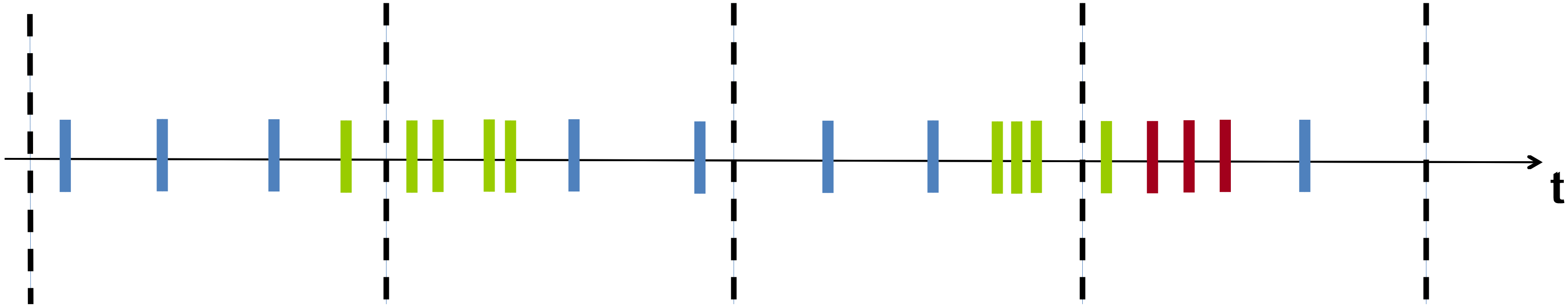
- Introduction to programming
- Discrete math
- Project presentation



Learning patterns: content + dynamics

1st year computer science student

- Introduction to programming
- Discrete math
- Project presentation



Content + Dynamics = ***Learning pattern***

programming + semester

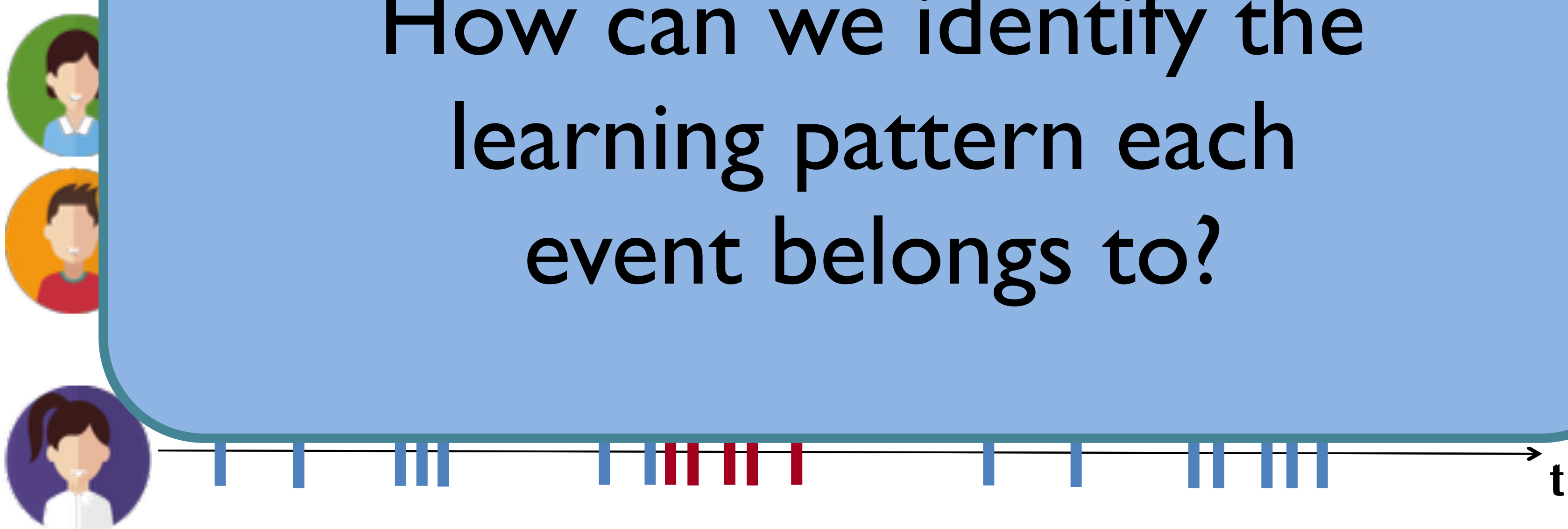
math + semester

presentation + week

People share same learning patterns

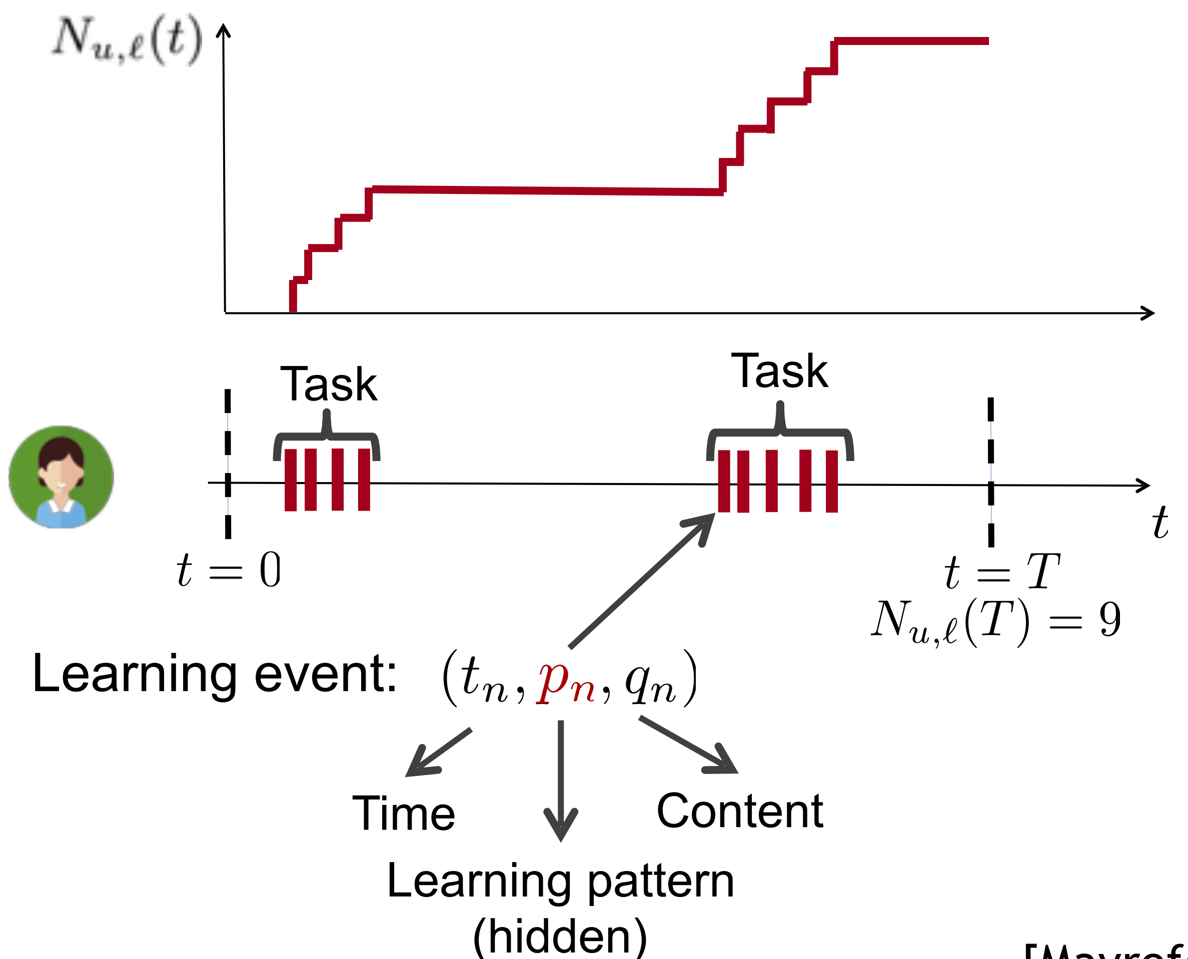
- Introduction to programming
- Discrete math
- Pr

How can we identify the learning pattern each event belongs to?



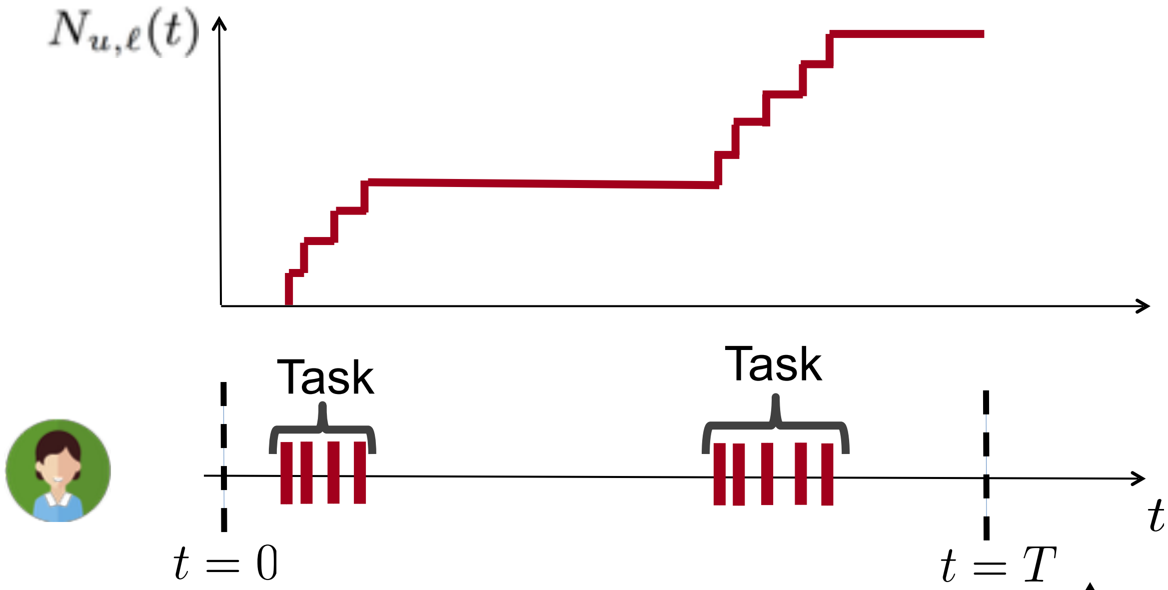
Learning events representation

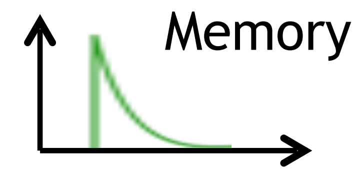
We represent the learning events using **marked temporal point processes:**



[Mavroforakis et al., WWW 2017]

Learning pattern intensity





New task rate

Learning pattern popularity

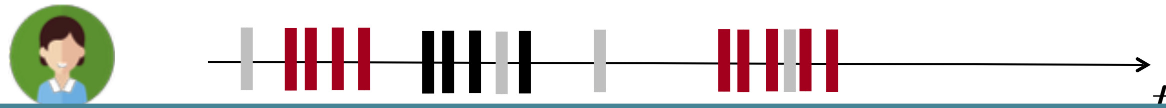
$$\underbrace{\lambda^*_{u,\ell}(t)}_{\substack{\textbf{Intensity} \\ \textbf{or rate} \\ \textbf{(events / hour)}}} = \underbrace{\mu_u \pi_\ell}_{\textbf{New task}} \underbrace{\qquad\qquad\qquad\qquad}_{\textbf{Follow-up}}$$

Hawkes process

[Mavroforakis et al., WWW 2017]

User learning events intensity

Users adopt more than one learning pattern:



# of learning patterns is **infinite**.

Efficient model inference using Sequential Montecarlo!

Details in the reference below!

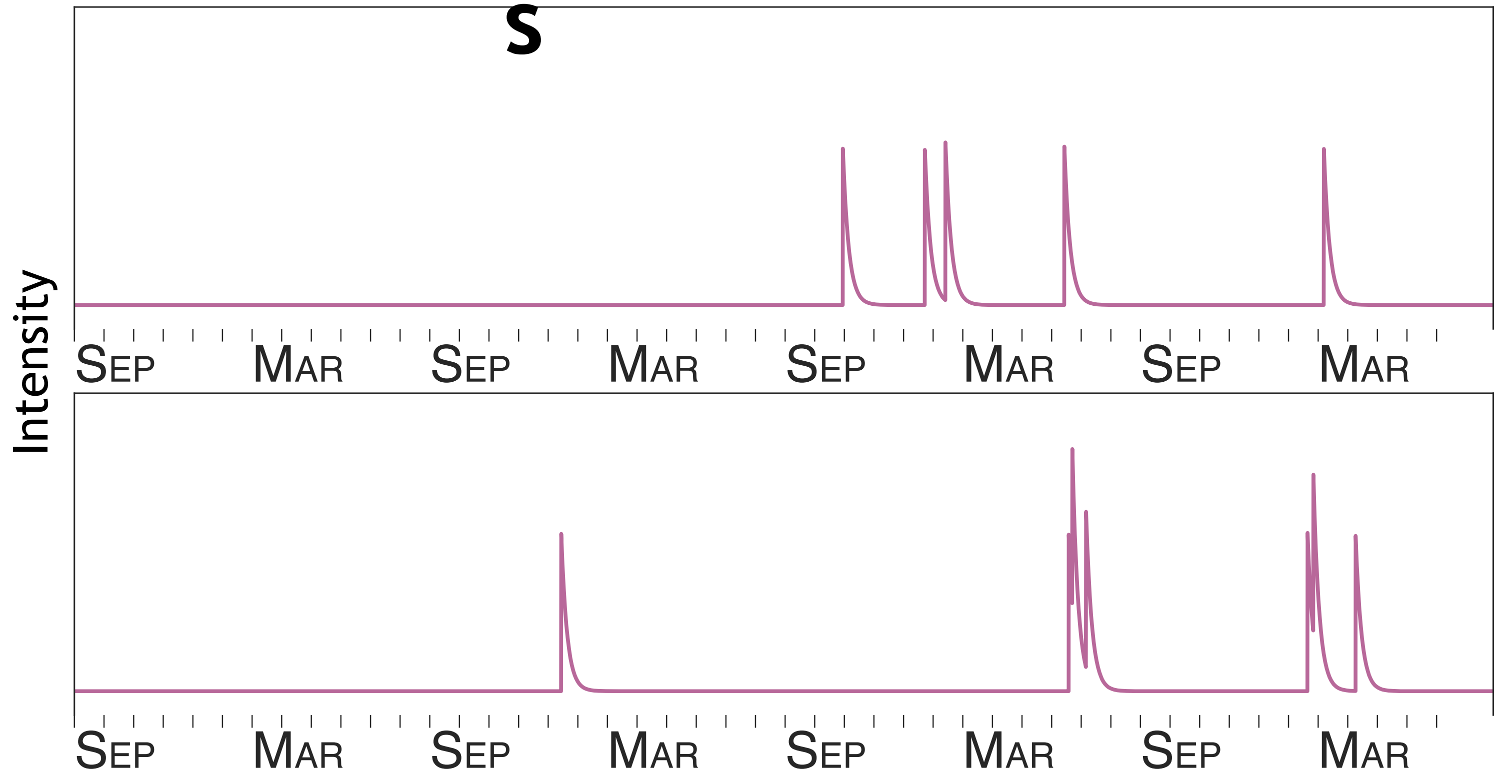
Content → $q_n = \boldsymbol{\omega} \quad \omega_j \sim Multinomial(\boldsymbol{\theta}_p)$

[Mavroforakis et al., WWW 2017]

Learning pattern (I): Version Control

Conten
t

Intensitie
s

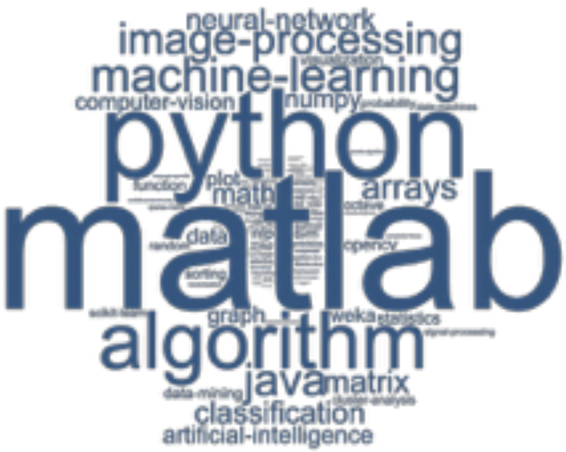


Version control tasks tend to be specific, quickly solved after performing few questions

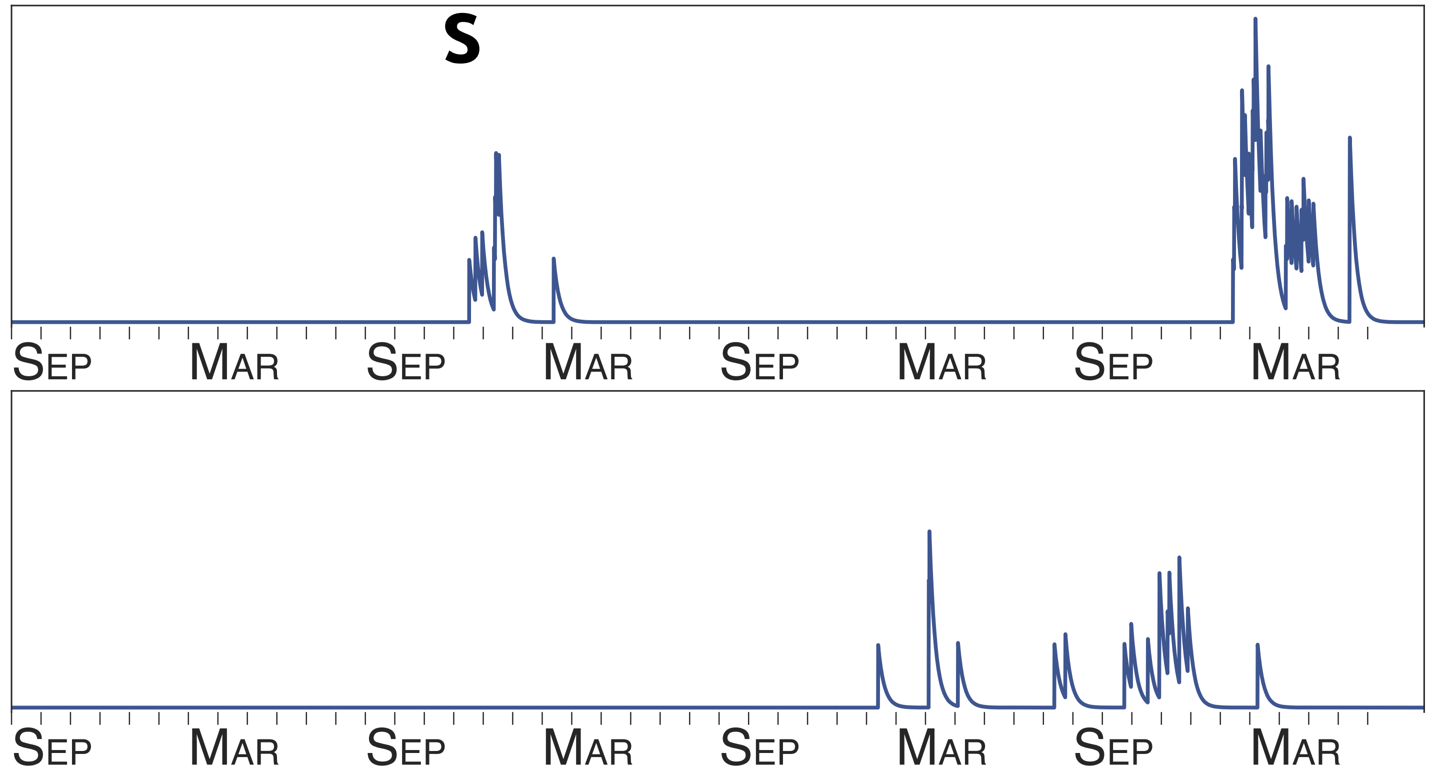
[Mavroforakis et al., WWW 2017]

Learning pattern (II): Machine learning

**Conten
t**



**Intensitie
s**

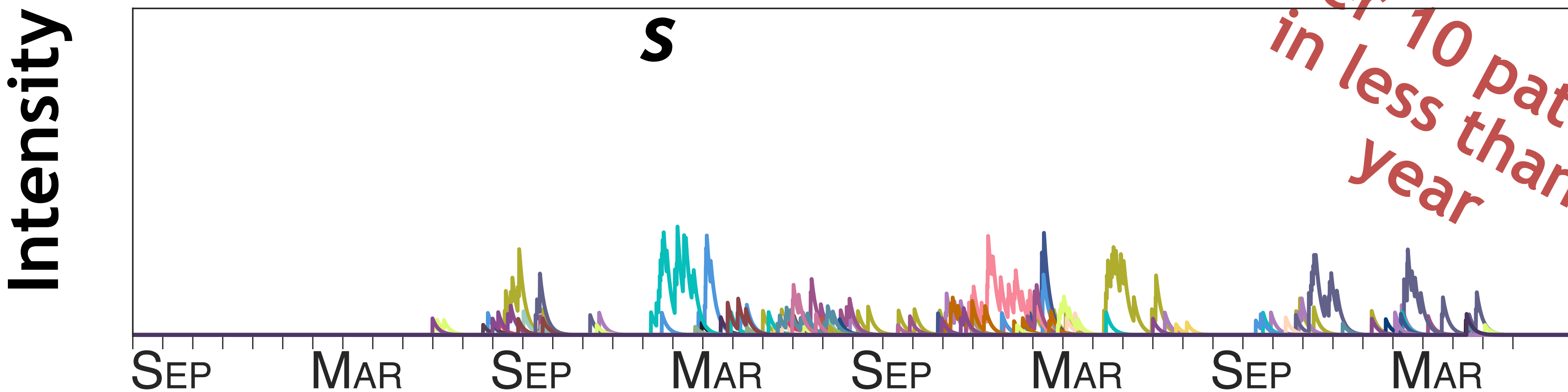


Machine learning tasks tend to be more complex and require asking more questions

[Mavroforakis et al., WWW 2017]

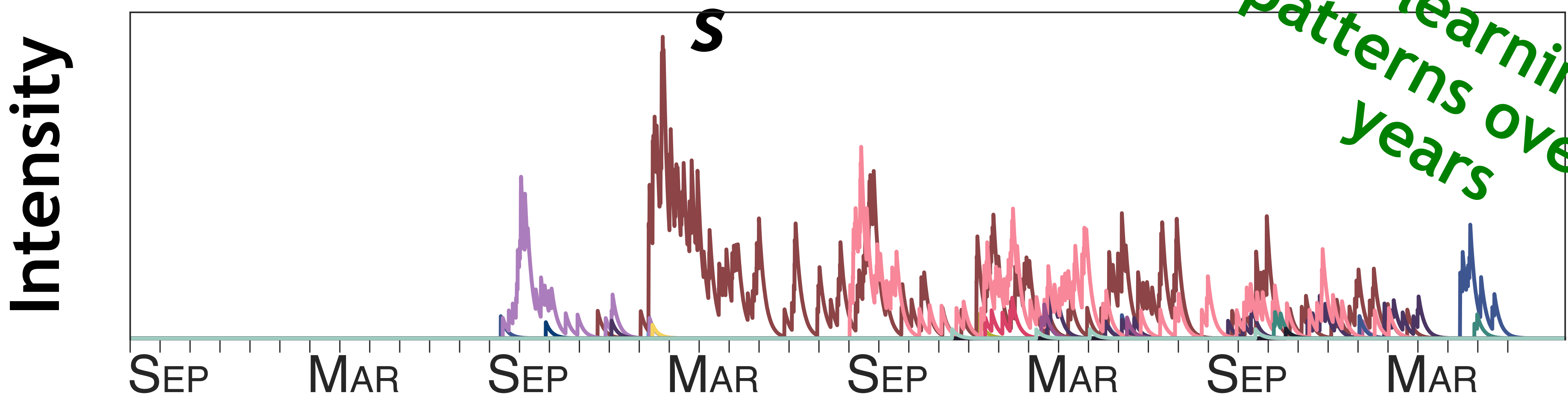
Types of users

Explorers



Over 10 patterns in less than a year

Loyals



2 learning patterns over 4 years

[Mavroforakis et al., WWW 2017]

Representation: Temporal Point processes

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

Applications: Models

1. Information propagation
2. Opinion dynamics
3. Information reliability
4. Knowledge acquisition

This lecture

Applications: Control

1. Influence maximization
2. Activity shaping
3. When to post
4. When to fact check

Next lecture