MLSS 2016 – Practical

Machine Learning for Networks

Isabel Valera
Utkarsh Upadhyay

Max
Planck
Institute
for
Software Systems

MPI for Software Systems

MLSS 2016
CADIZ

Getting ready

1. Install CVX for MATLAB / cvxpy for Python

Mathematically

http://cvxr.com/cvx/download/

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

In CVX

```
cvx_begin
   variables x(n)
   minimize(f0(x))
   subject to
      f(x) <= 0
      A * x - b == 0
cvx_end
```

```
import cvxpy as CVX
x = CVX.Variable(n)
constraints = [
  f(x) <= 0,
  A * x - b == 0
]
prob = CVX.Problem(

CVX.Minimize(f0(x)),
        constraints)
prob.solve()
```

NB: f_0 and f_i must be convex and h_i must be affine

Outline

Point Processes → Simulation and estimation

Part I

Information Diffusion

NetRate

[Gomez Rodriguez et al., ICML 2011]

Part II

Social Activity

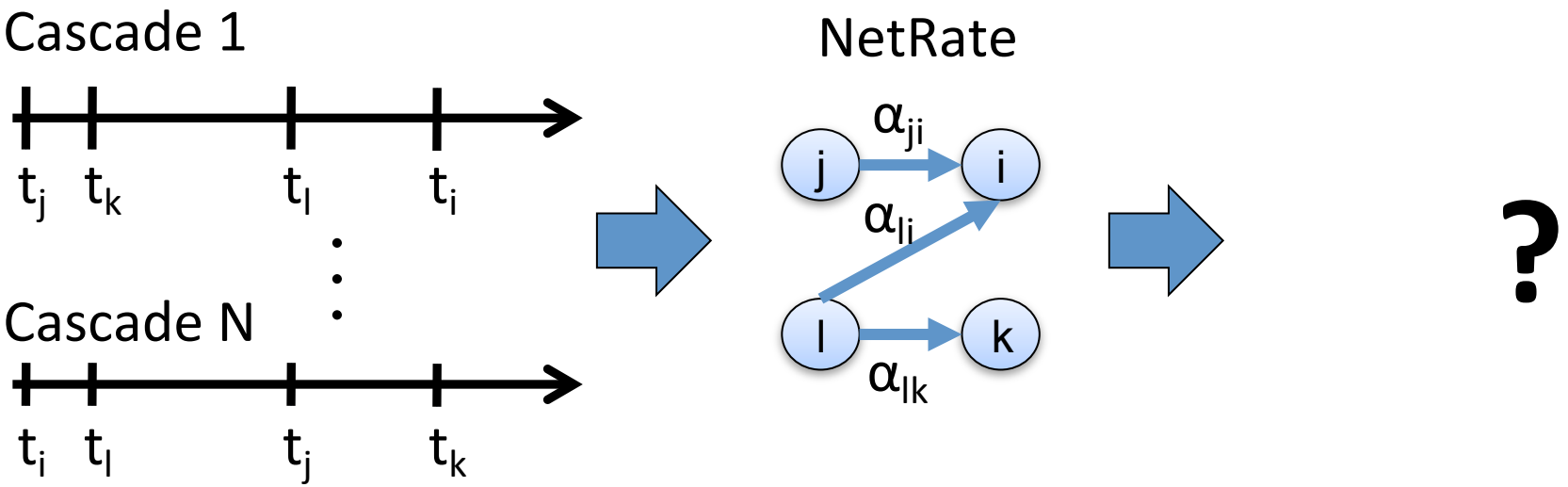
Hawkes Process

[Aalen et al., Springer 2008]

Task I

Estimate transmission rates in NetRate model with exponential transmission function.

Data **Model** **Estimation**

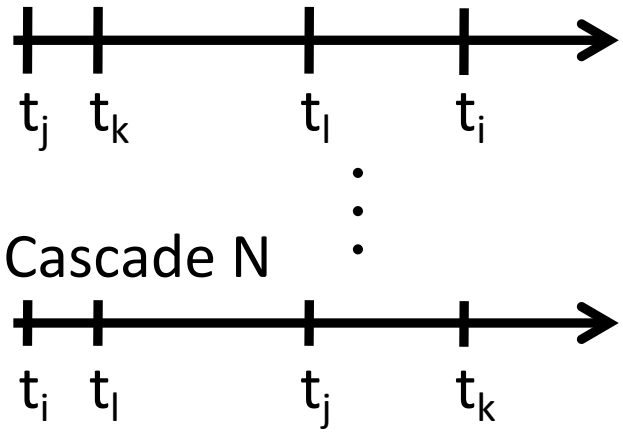


[Gomez Rodriguez et al., ICML 2011]

Task I

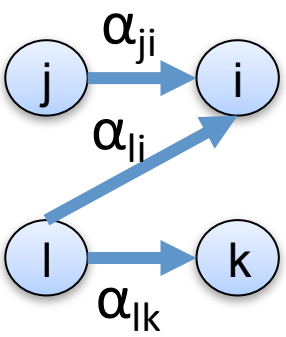
Estimate transmission rates in NetRate model with exponential transmission function.

Data



Model

NetRate



Estimation

MLE solution

$$\max_{\{\alpha_{ji}\}} \Sigma_c \log f(\{t_j\}^c ; \{\alpha_{ji}\})$$

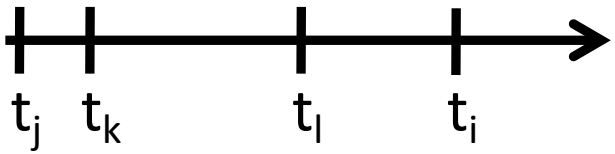
[Gomez Rodriguez et al., ICML 2011]

Task I

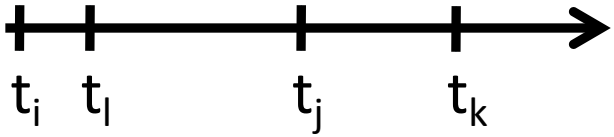
Estimate transmission rates in NetRate model with exponential transmission function.

Data

Cascade 1

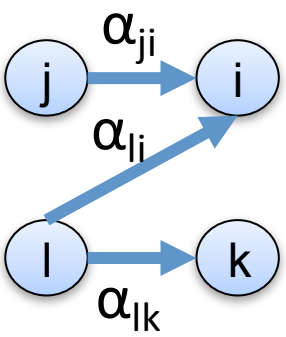


Cascade N



Model

NetRate



Estimation

MLE solution

$$\max_{\{\alpha_{ji}\}} \Sigma_c \log f(\{t_j\}^c ; \{\alpha_{ji}\})$$

Exercise 1

[Gomez Rodriguez et al., ICML 2011]

Task I

The **likelihood of a cascade** by time T is:

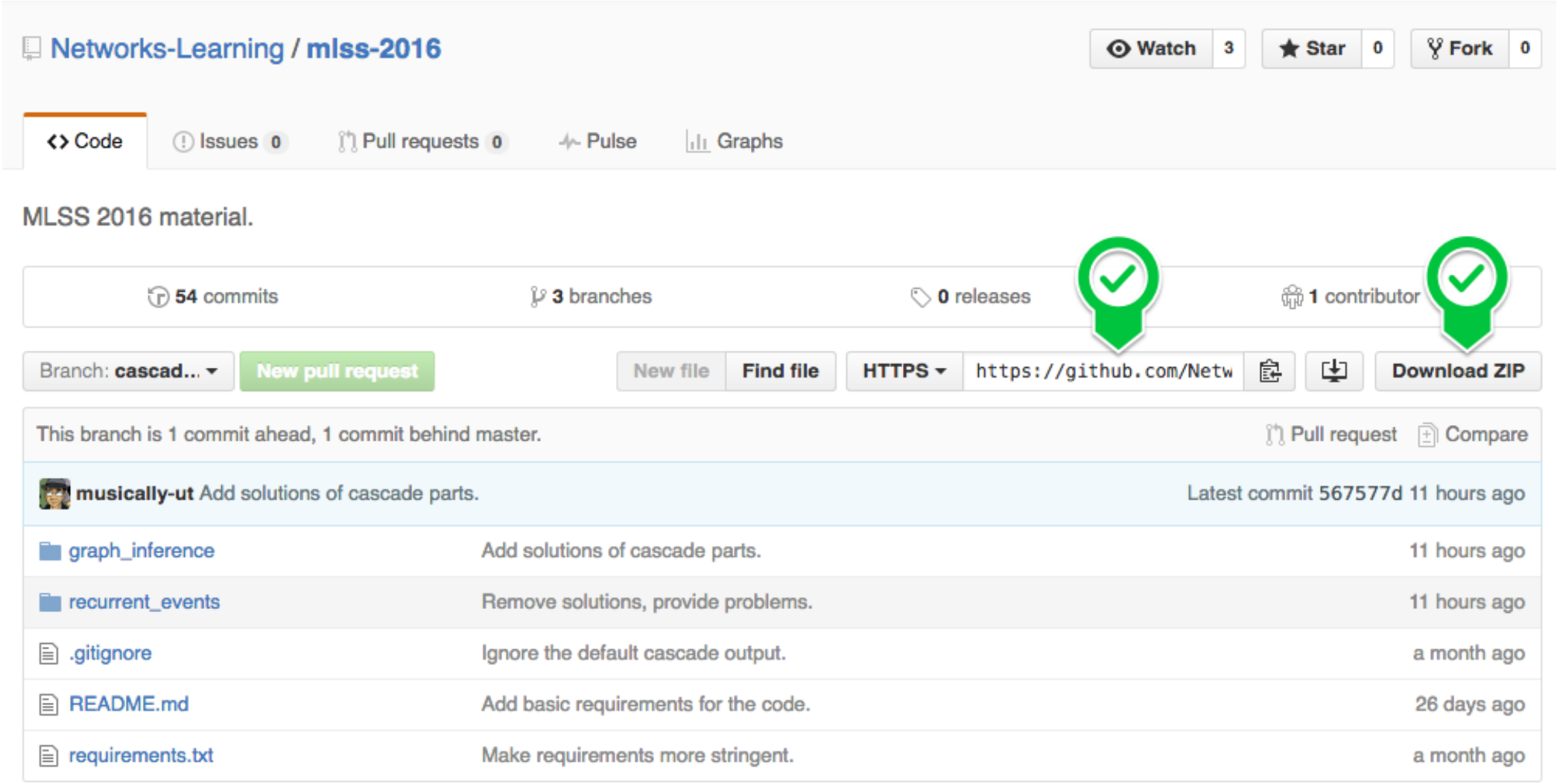
$$f(\mathbf{t};\mathbf{A}) = \prod_{i:t_i \le T} \prod_{m:t_m > T} S(T|t_i;\alpha_{i,m}) \times \prod_{k:t_k < t_i} S(t_i|t_k;\alpha_{k,i}) \left(\sum_{j:t_j < t_i} H(t_i|t_j;\alpha_{j,i}) \right)$$

MLE estimation:

$$\begin{array}{ll} \text{maximize}_{\mathbf{A}} & \sum_{c \in C} \log f(\mathbf{t}^c;\mathbf{A}) \\ \text{subject to} & \alpha_{j,i} \ge 0,\ i,j = 1,\ldots,N, i \ne j \end{array}$$

Task I

https://github.com/Networks-Learning/mlss-2016



OUTLINE

Point Processes → Simulation and estimation

Part I

Information Diffusion

NetRate

[Gomez Rodriguez et al., ICML 2011]

Part II

Social Activity

Hawkes Process

[Aalen et al., Springer 2008]

Task II

Generate samples from a (univariate) Hawkes Process

Model

Hawkes process

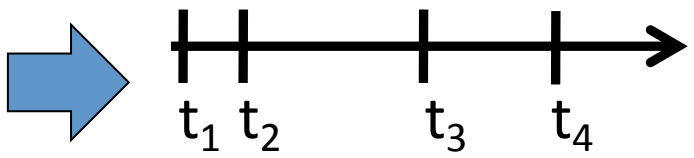
λ_0 = Base intensity

$$k_\alpha(t) = \alpha e^{-w(t-t_i)}$$

$$\lambda(t) = \lambda_0 + \sum_{i:t_i \in \mathcal{H}(t)} k_\alpha(t - ti)$$

Simulation

Ogata's algorithm



Estimation

MLE solution

$$\max_{\alpha,\lambda_0} \mathcal{L}(t_1, \ldots, t_n)$$

[Gomez Rodriguez et al., ICML 2011]

Task II

Generate samples from a (univariate) Hawkes Process

Model

Hawkes process

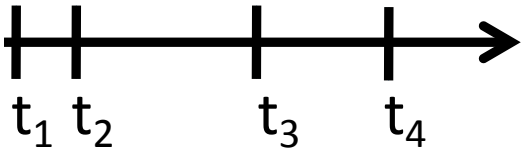
λ_0 = Base intensity

$$k_\alpha(t) = \alpha e^{-w(t-t_i)}$$

$$\lambda(t) = \lambda_0 + \sum_{i:t_i \in \mathcal{H}(t)} k_\alpha(t - ti)$$

Simulation

Ogata's algorithm



Exercise 2

Estimation

MLE solution

$$\max_{\alpha,\lambda_0} \mathcal{L}(t_1, \ldots, t_n)$$

[Gomez Rodriguez et al., ICML 2011]

Task II

Ogata's Algorithm:

1: **Input:** λ_0, α and T
2: Initialize: $n \leftarrow 1$, $\lambda^* \leftarrow \lambda_0$
Generate first event:
3: Generate $q \sim \mathcal{U}_{[0,1]}$ and $s \leftarrow -\frac{1}{\lambda^*}\ln(q)$
4: **if** $s > T$, **then** go to last step.
5: **else** Set $t_1 \leftarrow s$ and $n \leftarrow n+1$,
General subroutine:
6: **while** $s < T$ **do**
7: Update $\lambda^* \leftarrow \lambda(t_{n-1}) + \alpha$
8: Generate $q \sim \mathcal{U}_{[0,1]}$ and $s \leftarrow t_{n-1} - \frac{1}{\lambda^*}\ln(q)$
9: **if** $s > T$, **then** go to last step
10: **else Rejection Test:**
Sample $d \sim \mathcal{U}_{[0,1]}$
if $d \leq \frac{\lambda(s)}{\lambda^*}$, **then** $t_n \leftarrow s$ and $n \leftarrow n+1$
else update $\lambda^* \leftarrow \lambda(s)$.
end while
11: **Output:** Retrieve the events $(\{t_n\})$ on $[0, T]$

[Ogata, 1981]

Task II

Ogata's Algorithm:

Input: λ_0, α and T
Initialize: $n \leftarrow 1$, $\lambda^* \leftarrow \lambda_0$
Generate first event:
Generate $q \sim \mathcal{U}_{[0,1]}$ and $s \leftarrow -\frac{1}{\lambda^*}\ln(q)$
if $s > T$, **then** go to last step.
else Set $t_1 \leftarrow s$ and $n \leftarrow n+1$,
General subroutine:
while $s < T$ **do**
  Update $\lambda^* \leftarrow \lambda(t_{n-1}) + \alpha$
  Generate $q \sim \mathcal{U}_{[0,1]}$ and $s \leftarrow t_{n-1} - \frac{1}{\lambda^*}\ln(q)$
  if $s > T$, **then** go to last step
  else Rejection Test:
    Sample $d \sim \mathcal{U}_{[0,1]}$
    if $d \leq \frac{\lambda(s)}{\lambda^*}$, **then** $t_n \leftarrow s$ and $n \leftarrow n+1$
    else update $\lambda^* \leftarrow \lambda(s)$.
end while
Output: Retrieve the events $(\{t_n\})$ on $[0, T]$

$$\lambda(t) = \lambda_0 + \sum_{i:t_i \in \mathcal{H}(t)} k_\alpha(t - ti)$$

History up to but not including t.

[Ogata, 1981]

Task II

Conditional intensity rate

$$\lambda_u^*(t)$$

Events likelihood

$$\mathfrak{L} = \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau$$

Maximum likelihood approach to find model parameters!

$$\max_{\alpha, \lambda_0} \mathcal{L}(t_1, \ldots, t_n)$$

[Farajtabar et al., NIPS 2014]

Downloads

NetRate:

Paper: http://www.mpi-sws.org/~manuelgr/pubs/netrate-icml11.pdf
Code: https://github.com/Networks-Learning/netrate

Hawkes Process:

Slides: http://lamp.ecp.fr/MAS/fiQuant/ioane_files/HawkesCourseSlides.pdf
Code: https://github.com/dunan/MultiVariatePointProcess

CVX:

Slides: http://web.stanford.edu/class/ee364a/lectures/cvx_tutorial.pdf
Code: http://cvxr.com/cvx/download/