Statistical learning with Hawkes processes and new matrix concentration inequalities

Emmanuel Bacry[1], Stéphane Gaïffas[1], Jean-Francois Muzy[1,2]

Winter 2016

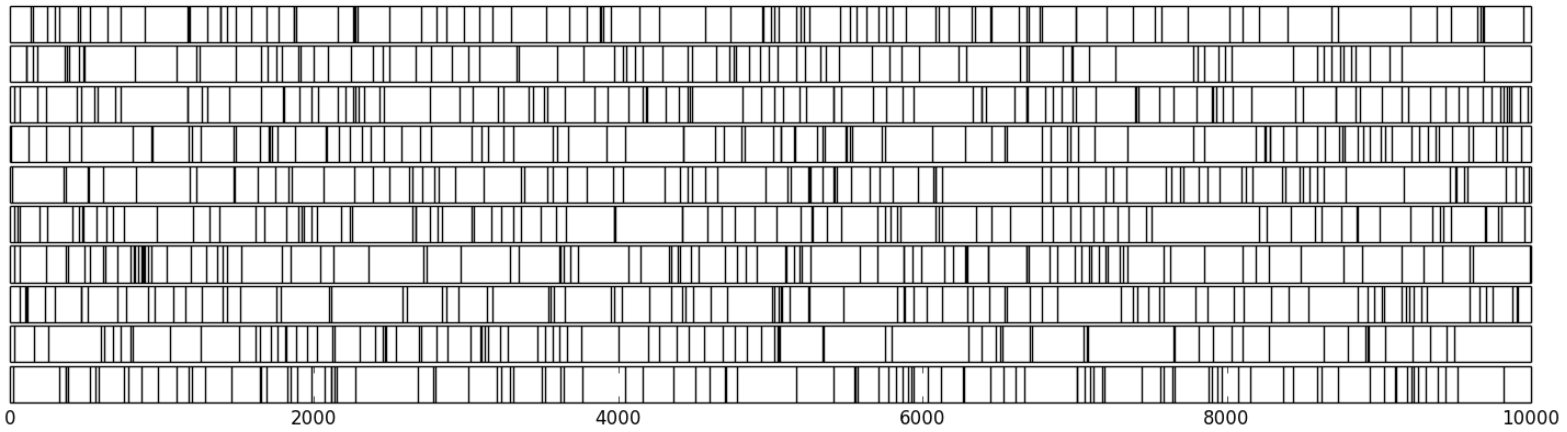
[1]École Polytechnique and CNRS
[2]CNRS, Université de Corse

Introduction

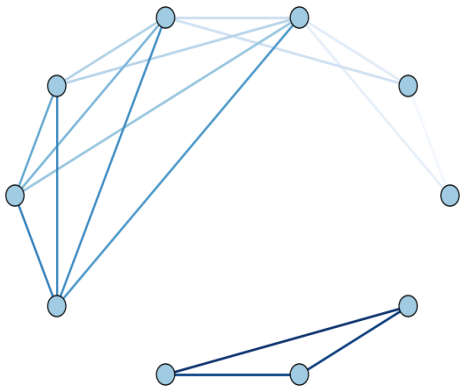
- You have users of a system (a social network, an e-commerce platform, etc.)
- You want to quantify the level of interaction between users
- You don't want to use only *declared* interactions, such as "friendship" or "likes". This information is often deprecated, and not really related to the activity of users
- You want levels of interaction driven by user's actions, using the timestamps' patterns of actions

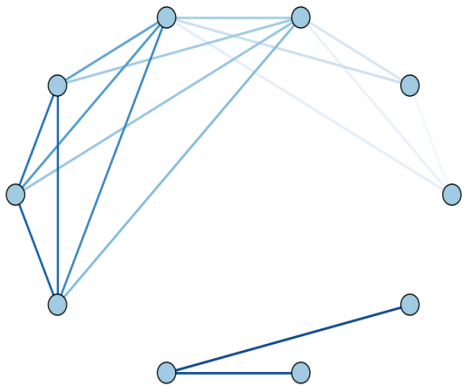
Introduction

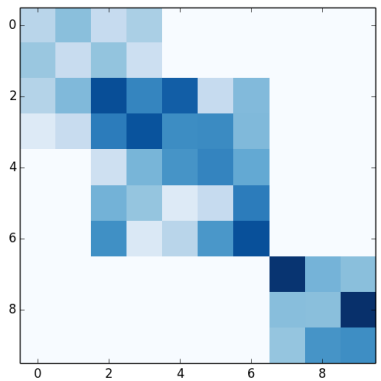
From:



We want to quantify interactions between users:







Model: Multivariate Hawkes Process (MHP)

- A d-dimensional counting process $N = [N_1, \ldots, N_d]^\top$
- d is "large"
- Observed on $[0, T]$. "Asymptotics" in $T \to +\infty$
- N_j has intensity λ_j, namely

$$\mathbb{P}\big(N_j \text{ has a jump in } [t, t+dt] \mid \mathcal{F}_t\big) = \lambda_j(t)dt$$

for $j = 1, \ldots, d$ where $\mathcal{F}_t$ some filtration

Model: Multivariate Hawkes Process (MHP)

- MHP assumes the following autoregressive structure:

$$\lambda_j(t) = \mu_j(t) + \int_{(0,t)} \sum_{k=1}^{d} \varphi_{j,k}(t-s) dN_k(s),$$

- $\mu_j(t) \geq 0$ baseline intensity of the j-th coordinate
- $\varphi_j : \mathbb{R}^+ \to \mathbb{R}^+$ self-exciting component
- Write this in matrix form

$$\lambda(t) = \boldsymbol{\mu} + \int_{(0,t)} \boldsymbol{\varphi}(t-s) dN(s),$$

with $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^\top$ and $\boldsymbol{\varphi}(t) = [\varphi_{j,k}(t)]_{1 \leq j,k \leq d}$.

- Notation:

$$\int_{(0,t)} \varphi(t-s) dN_k(s) = \sum_{i:0<T_{i,k}<t} \varphi(t - T_{i,k})$$

A brief history of MHP

Introduced by Hawkes in 1971

- **Earthquakes and geophysics** : Kagan and Knopoff (1981), Zhuang, Harte, Werner, Hainzl and Zhou (2012)
- **Genomics** : Reynaud-Bouret and Schbath (2010)
- **High-frequency Finance** : Bacry Delattre Hoffmann and Muzy (2013)
- **Terrorist activity** : Porter and White (2012)
- **Neurobiology** : Hansen, Reynaud-Bouret and Rivoirard (2012)
- **Social networks** : Carne and Sornette (2008), Simma and Jordan (2010), Zhou Song and Zha (2013)
- And even **FPGA-based implementation** : Guo and Luk (2013)

A brief history of MHP

THE GENESIS BLOCK

Digital currency research and data

HOME | NEWS | MINING | TRADING | ECONOMICS | REGULATION | BUSINESSES | BI

Home / Bitcoin 201 / Analyzing Trade Clustering To Predict Price Movement In Bitcoin Trading

Analyzing Trade Clustering To Predict Price Movement In Bitcoin Trading

Sep 19, 2013 Posted By Jonathan Heusser In Bitcoin 201, Economics, Featured, News, Trading Tagged Analysis, Bitcoin Trading, Hawkes Process, Jonathan Heusser, London, Price, Trading

Estimation for MHP: some references

Parametric estimation (Maximum likelihood)

- First work : Ogata 78
- Simma and Jordan (2010), Zhou Song and Zha (2013)
  $\rightarrow$ Expected Maximization (EM) algorithms, with priors

Non parametric estimation

- Marsan Lengliné (2008), generalized by Lewis, Mohler (2010)
  $\rightarrow$ EM for penalized likelihood function
  $\rightarrow$ Monovariate Hawkes processes, Small amount of data, No theoretical results
- Reynaud-Bouret and Schbath (2010)
  $\rightarrow$ Developed for small amount of data (Sparse penalization)
- Bacry and Muzy (2014)
  $\rightarrow$ Larger amount of data

What do we want to do with this?

- Do inference directly from **actions** of users
- Understand the community structure of users underlying the actions
- Exploit the hidden lower-dimensional structure of the network for inference/prediction

MHP in large dimension

Dimension d is large:

- Need a simple parametric model on μ and φ
- For inference: we want a **tractable** and **scalable** optimization problem
- We want to encode some prior assumptions using (convex) penalization

A simple parametrization of the MHP

Simple parametrization:

- Constant baselines $\mu_j(\cdot) \equiv \mu_j$
- Take

$$\varphi_{j,k}(t) = a_{j,k} e^{-\alpha_{j,k} t}$$

- $a_{j,k} =$ level of interaction between nodes j and k
- $\alpha_{j,k} =$ lifetime of instantaneous excitation of node j by node k

The matrix

$$\boldsymbol{A} = [a_{j,k}]_{1 \leq j,k \leq d}$$

is understood has a **weighted adjacency matrix** of mutual excitement of nodes $\{1, \ldots, d\}$

- $\boldsymbol{A}$ is non-symmetric: "oriented graph"

A simple parametrization of the MHP

We end up with intensities

$$\lambda_{j,\theta}(t) = \mu_j + \int_{(0,t)} \sum_{k=1}^{d} a_{j,k} e^{-\alpha_{j,k}(t-s)} dN_k(s)$$

for $j \in \{1, \dots, d\}$ where

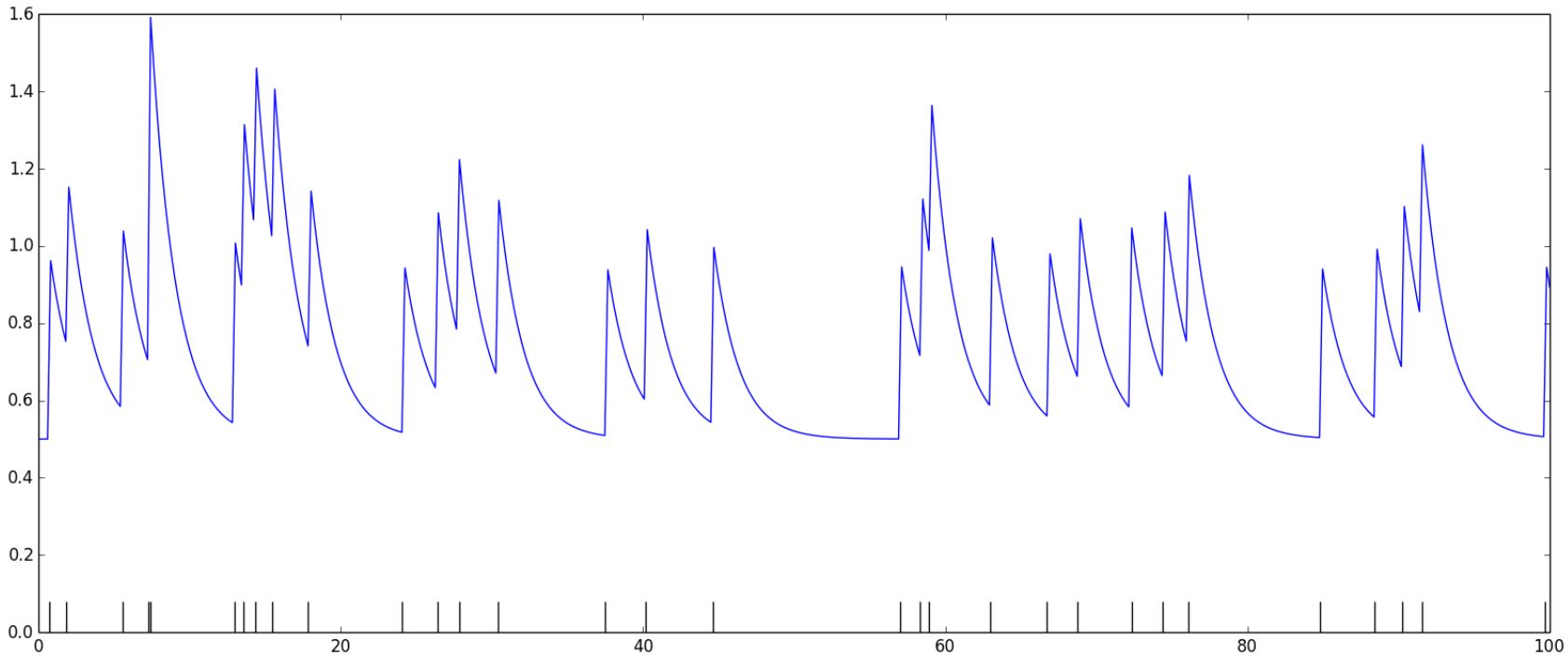
$$\theta = [\mu, \boldsymbol{A}, \boldsymbol{\alpha}]$$

with

- baselines $\mu = [\mu_1, \dots, \mu_d]^\top \in \mathbb{R}_+^d$
- interactions $\boldsymbol{A} = [a_{j,k}]_{1 \le j,k \le d} \in \mathbb{R}_+^{d \times d}$
- decays $\boldsymbol{\alpha} = [\alpha_{j,k}]_{1 \le j,k \le d} \in \mathbb{R}_+^{d \times d}$

A simple parametrization of the MHP

For $d = 1$, intensity λ_θ looks like this:



Minus log-**likelihood**

$$-\ell_T(\theta) = \sum_{j=1}^{d} \Big\{ \int_0^T (\lambda_{j,\theta}(t) - 1) dt - \int_0^T \log \lambda_{j,\theta}(t) dN_j(t) \Big\}$$

Least-squares

$$R_T(\theta) = \sum_{j=1}^{d} \Big\{ \int_0^T \lambda_{j,\theta}(t)^2 dt - 2 \int_0^T \lambda_{j,\theta}(t) dN_j(t) \Big\}$$

with

$$\lambda_{j,\theta}(t) = \mu_j + \sum_{k=1}^{d} a_{j,k} \int_{(0,t)} \exp\big(-\alpha_{j,k}(t-s)\big) dN_k(s)$$

where $\theta = [\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{\alpha}]$ with $\boldsymbol{\mu} = [\mu_j]$, $\boldsymbol{A} = [a_{j,k}]$, $\boldsymbol{\alpha} = [\alpha_{j,k}]$

A simple framework

Put $\|\lambda_\theta\|_T^2 = \langle \lambda_\theta, \lambda_\theta \rangle_T$ with

$$\langle \lambda_\theta, \lambda_{\theta'} \rangle_T = \frac{1}{T} \sum_{j=1}^{d} \int_{[0,T]} \lambda_{j,\theta}(t) \lambda_{j,\theta'}(t) dt.$$

so that least-squares writes

$$R_T(\theta) = \|\lambda_\theta\|_T^2 - \frac{2}{T} \sum_{j=1}^{d} \int_{[0,T]} \lambda_{j,\theta}(t) dN_j(t)$$

It is natural: if N has ground truth intensity λ^* then

$$\mathbb{E}[R_T(\theta)] = \mathbb{E}\|\lambda_\theta\|_T^2 - 2\mathbb{E}\langle \lambda_\theta, \lambda^* \rangle_T = \mathbb{E}\|\lambda_\theta - \lambda^*\|_T^2 - \|\lambda^*\|_T,$$

where we used "signal + noise" decomposition (Doob-Meyer):

$$dN_j(t) = \lambda^*(t) dt + dM_j(t)$$

where M_j martingale

A simple framework

A strong assumption: assume that

$$\varphi_{j,k}(t) = a_{j,k} h_{j,k}(t)$$

for **known** $h_{j,k}$ meaning that

$$\lambda_{j,\theta}(t) = \mu_j + \int_{(0,t)} \sum_{k=1}^{d} a_{j,k} h_{j,k}(t-s) dN_k(s),$$

where $\theta = [\mu, \boldsymbol{A}]$ with $\mu = [\mu_1, \ldots, \mu_d]^\top$ and $\boldsymbol{A} = [a_{j,k}]_{1 \leq j,k \leq d}$

A simple framework

Ways of cheating:

- approximate $h_{j,k}$ using a known dictionary $\{h_1, \ldots, h_L\}$and learn the coefficients

$$\lambda_{j,\theta}(t) = \mu_j + \int_{(0,t)} \sum_{k=1}^{d} \sum_{l=1}^{L} a_{j,k,l} h_l(t-s) dN_k(s),$$

  (but then use group Lasso $\sum_{1 \leq j,k \leq d} \|h_{j,k,\bullet}\|_2$)
- Don't estimate the kernels... (ongoing work)

But let's stick with the simple framework

Prior encoding by penalization

Prior assumptions

- Some users are basically inactive and react only if stimulated:

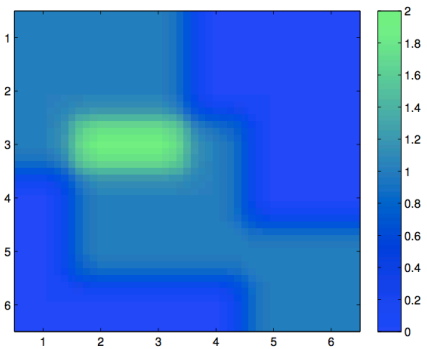
$$\mu \text{ is sparse}$$

- Everybody does not interact with everybody:

$$\boldsymbol{A} \text{ is sparse}$$

- Interactions have community structure, possibly overlapping, a small number of factors explain interactions:

$$\boldsymbol{A} \text{ is low-rank}$$



Prior encoding by penalization

Standard convex relaxations [Tibshirani (01), ..., Srebro et al. (05), Bach (08), Candès & Recht (08), ...]

- Convex relaxation of $\|\boldsymbol{A}\|_0 = \sum_{j,k} \mathbf{1}_{\boldsymbol{A}_{j,k}>0}$ is ℓ_1-norm:

$$\|\boldsymbol{A}\|_1 = \sum_{j,k} |\boldsymbol{A}_{j,k}|$$

- Convex relaxation of rank is trace-norm:

$$\|A\|_* = \sum_j \sigma_j(A) = \|\sigma(A)\|_1$$

where $\sigma_1(A) \geq \cdots \geq \sigma_d(A)$ singular values of $\boldsymbol{A}$

Prior encoding by penalization

So, we use the following penalizations

- Use ℓ_1 penalization on $\boldsymbol{\mu}$
- Use ℓ_1 penalization on $\boldsymbol{A}$
- Use trace-norm penalization on $\boldsymbol{A}$

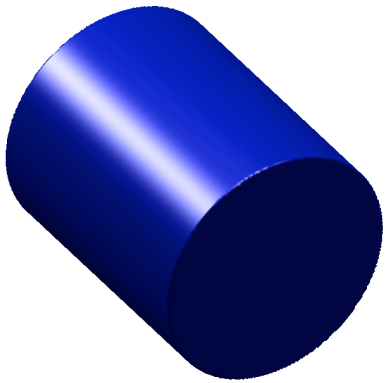
[but other choices might be interesting...]

NB1: to induce **sparsity AND low-rank** on $\boldsymbol{A}$, we use the mixed penalization

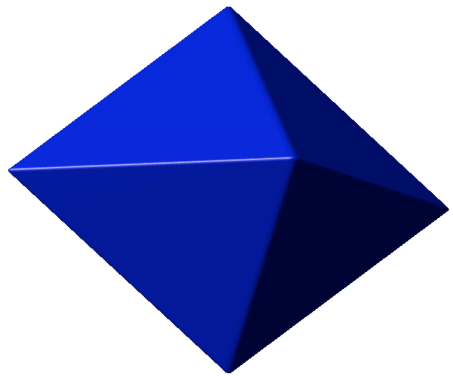
$$\boldsymbol{A} \mapsto w_* \|\boldsymbol{A}\|_* + w_1 \|\boldsymbol{A}\|_1$$

NB2: recent work by Richard et al (2013): much better way to induce sparsity and low-rank than this (but no theory)

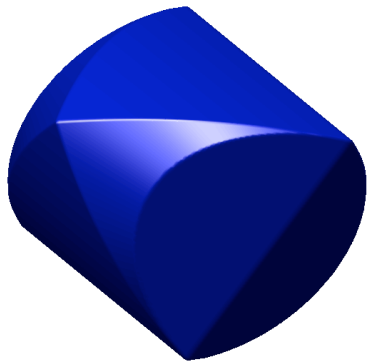
Sparse and low-rank matrices



$\{\boldsymbol{A} : \|A\|_* \leq 1\}$



$\{\boldsymbol{A} : \|\boldsymbol{A}\|_1 \leq 1\}$



$\{\boldsymbol{A} : \|\boldsymbol{A}\|_1 + \|\boldsymbol{A}\|_* \leq 1\}$

The balls are computed on the set of 2×2 symmetric matrices, which is identified with $\mathbb{R}^3$.

Algorithm

We end up with the problem

$$\hat{\theta} \in \underset{\theta \in \mathbb{R}_+^d \times \mathbb{R}_+^{d \times d}}{\operatorname{argmin}} \left\{ R_T(\theta) + \operatorname{pen}(\theta) \right\},$$

with mixed penalizations

$$\operatorname{pen}(\theta) = \tau_1 \|\mu\|_1 + \gamma_1 \|\boldsymbol{A}\|_1 + \gamma_* \|\boldsymbol{A}\|_*$$

But there is the "**features scaling**" problem

- Features scaling is necessary for "linear approaches" in supervised learning
- No features and labels here!

$\Rightarrow$ Can be solved here by fine tuning of the penalization terms

Algorithm

Consider instead

$$\hat{\theta} \in \underset{\theta \in \mathbb{R}_+^d \times \mathbb{R}_+^{d \times d}}{\operatorname{argmin}} \left\{ R_T(\theta) + \operatorname{pen}(\theta) \right\},$$

where this time

$$\operatorname{pen}(\theta) = \|\mu\|_{1,\hat{w}} + \|\boldsymbol{A}\|_{1,\hat{\boldsymbol{W}}} + \hat{w}_* \|\boldsymbol{A}\|_*$$

- Penalization tuned by data-driven weights $\hat{w}$, $\hat{\boldsymbol{W}}$ and $\hat{w}_*$ to solve the "scaling" problem
- Comes from sharp controls of the noise terms, using new probabilistic tools

Convex optimization – numerical aspects

- Can be solved using first-order routines
- Gradient of $R_T(\theta)$ using a recursion formula [Ogata (1988)]

  $\rightarrow$ When carefully done complexity of one gradient is $O(nd)$ (instead of $O(n^2 d)$ for the naive approach), where $n =$ number of events (very large)

  $\rightarrow$ The gradient on each node $j \in \{1, \ldots, d\}$ can be computed in **parallel**
- Computation bootleneck is the heavy use of exp and log [accelerated using some ugly hacking]
- Proximal of trace norm requires many truncated SVD: we use the default's Lanczos's implementation of Python (fast enough)

Machine Learning (*not only*) with Point Processes

`MLPP` is yet another machine learning library for `Python`, with a particular emphasis on point processes (Poisson processes, Hawkes processes, Cox regression), but includes also generalized linear models (GLM). It comes with optimization algorithms for inference and provides tools for simulation of datasets. A particular focus is on optimization: an extensive optimization toolbox is proposed, with recent state-of-the-art stochastic solvers.

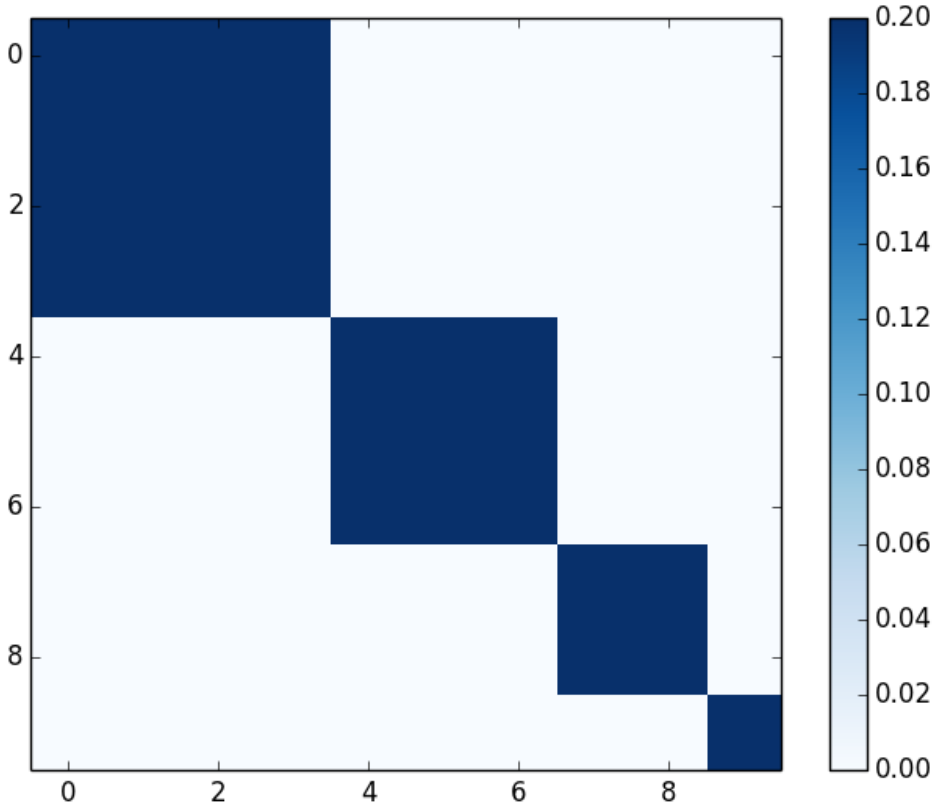
Main highlights

- **Python 3 only**!
- Fast, most computation are done in C++11 (including multi-threading)
- Contains an optimization toolbox: solvers (optimization algorithms), models (for computing gradient among other things) and prox (penalization) classes can be combined interactively in `iPython` for instance
- Features state-of-the-art stochastic optimization algorithms, with parallel implementations
- Support sparse datasets (sparse features matrices)
- Interplay between Python and C++ is done using swig

Optimization toolbox

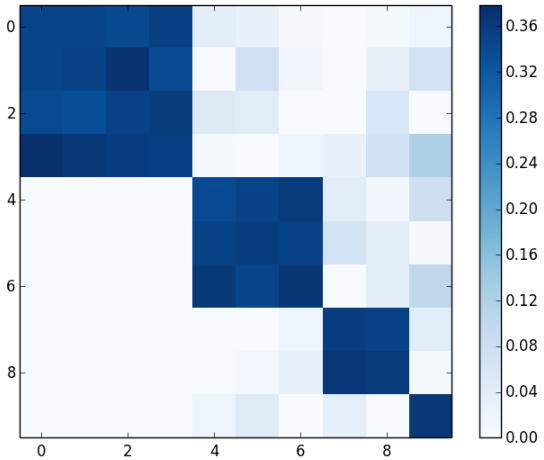
Numerical experiment

Toy example: take matrix $\boldsymbol{A}$ as

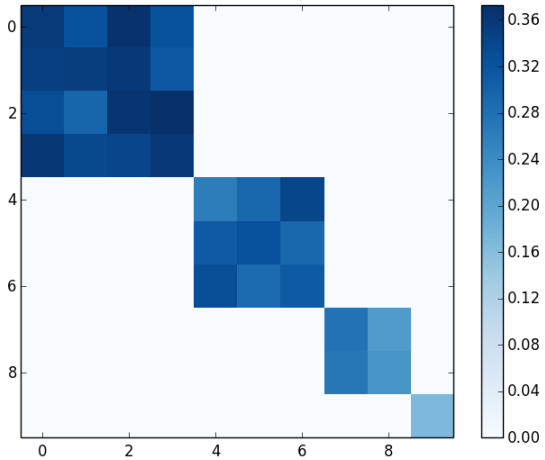


Numerical experiment: dimension 10, 210 parameters

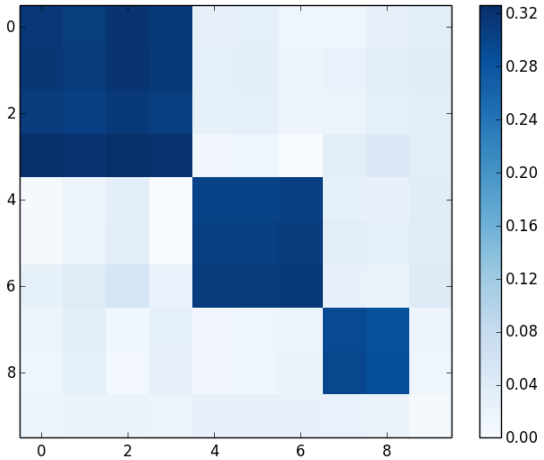
No penalization



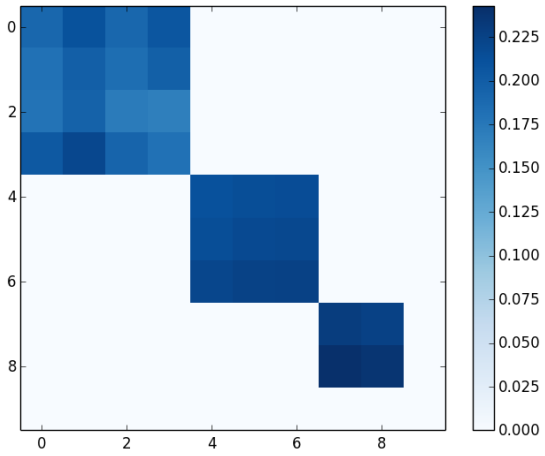
ℓ_1 penalization



trace-norm penalization

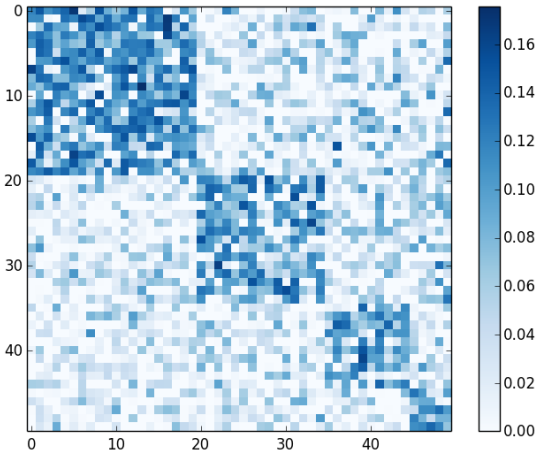


ℓ_1 + trace norm penalization

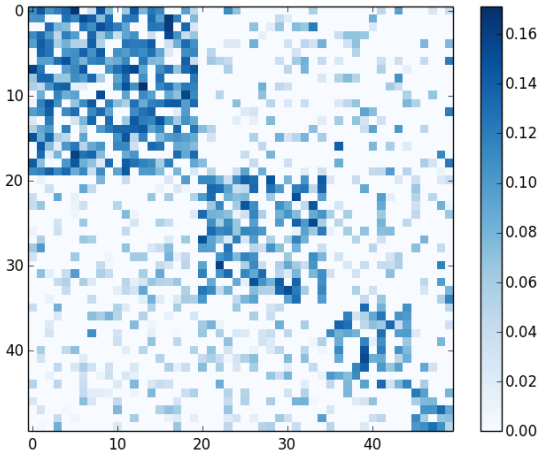


Numerical experiment: dimension 100, 20100 parameters

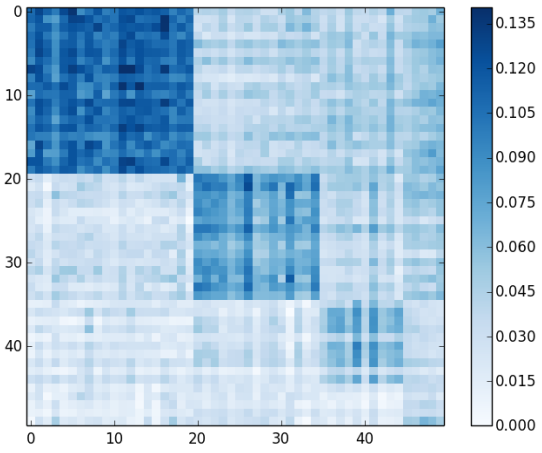
No penalization



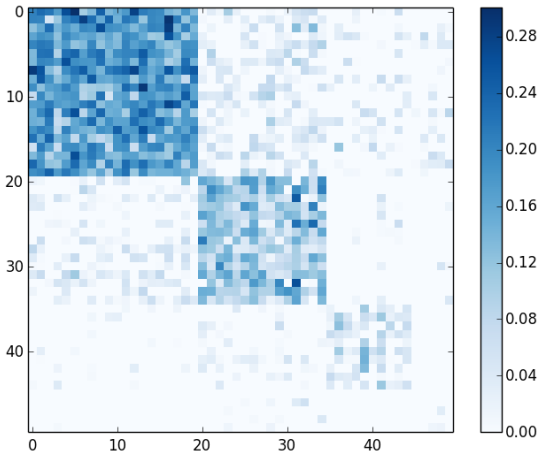
ℓ_1 penalization



trace-norm penalization

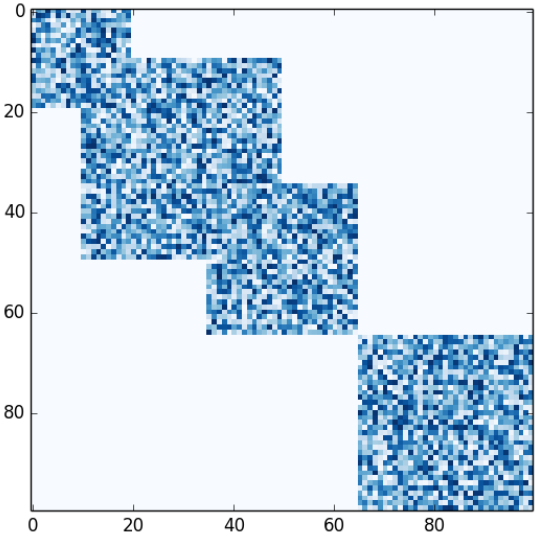


ℓ_1 + trace norm penalization

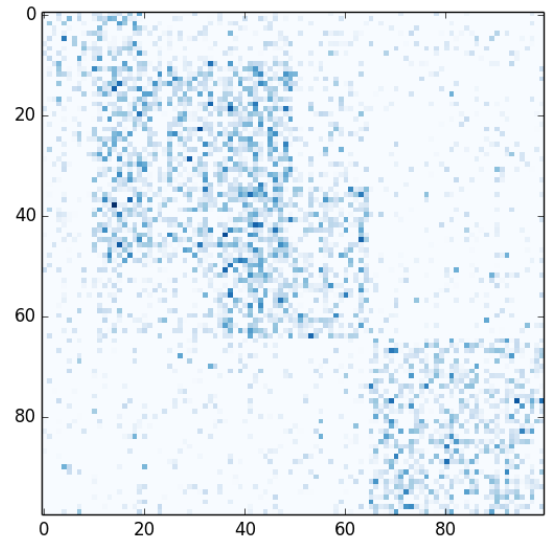


Numerical experiment: dimension 100, 20100 parameters

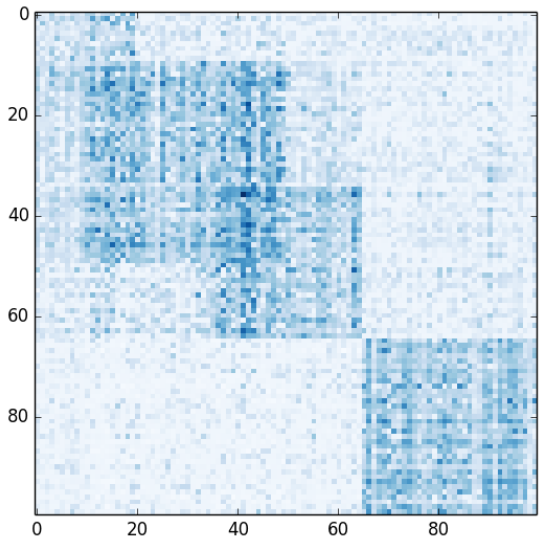
Ground truth



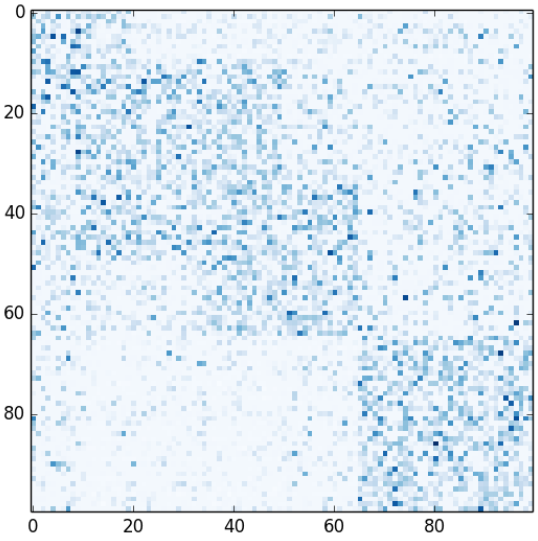
ℓ_1



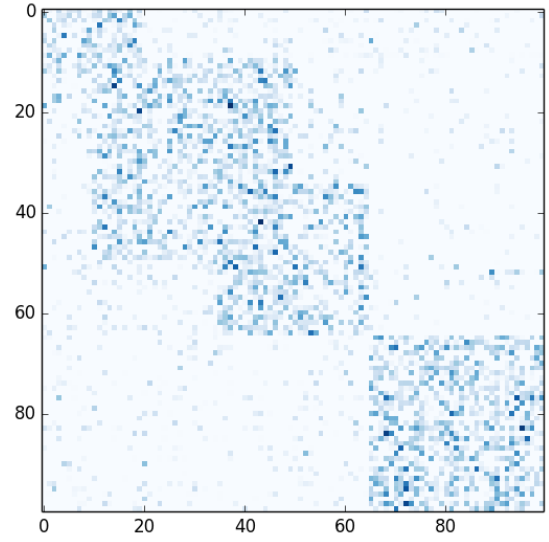
ℓ_1 + trace



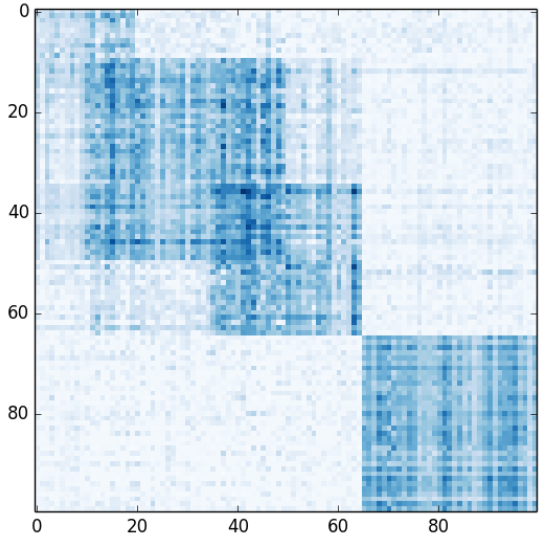
No pen



w-ℓ_1

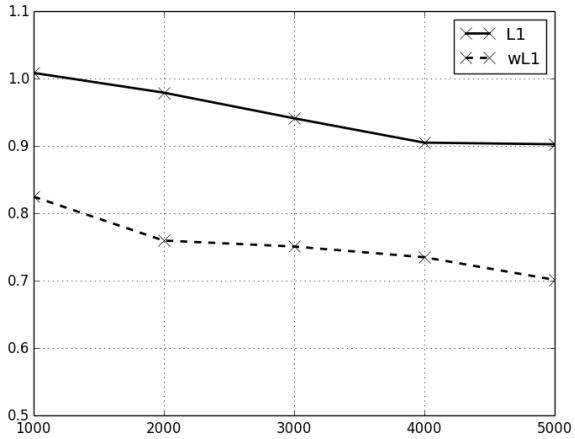


w-ℓ_1 + trace

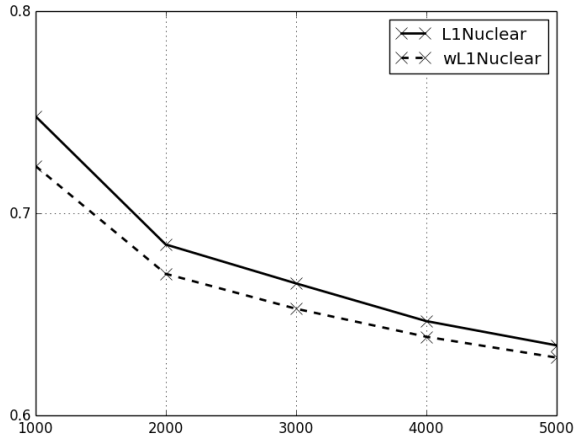


Estimation errors of $\boldsymbol{A}$ (measured by ℓ_2 norm)

ℓ_1 VS w-ℓ_1

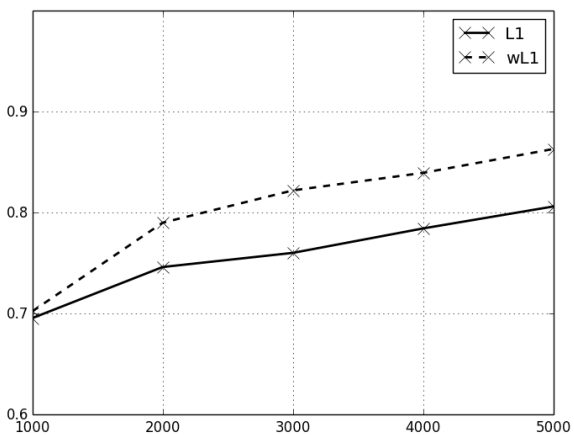


ℓ_1 + trace VS w-ℓ_1 + trace



AUC for support selection $\boldsymbol{A}$

ℓ_1 + trace VS w-ℓ_1 + trace



Theoretical guarantees: an oracle inequality

Towards a statistical guarantee: first order condition can be written as: for any θ

$$\|\lambda_{\hat{\theta}} - \lambda^*\|_T^2 + \|\lambda_{\hat{\theta}} - \lambda_\theta\|_T^2 - \|\lambda_\theta - \lambda^*\|_T^2$$
$$\leq -\langle \theta_\partial, \hat{\theta} - \theta \rangle + \frac{2}{T}\langle \hat{\mu} - \mu, \bar{M}_T \rangle + \frac{2}{T}\langle \hat{\boldsymbol{A}} - \boldsymbol{A}, \boldsymbol{Z}_T \rangle,$$

for $\theta_\partial \in \partial \,\mathrm{pen}(\theta)$ and we use $\frac{2}{T}\langle \hat{\boldsymbol{A}} - \boldsymbol{A}, \boldsymbol{Z}_T \rangle \leq \frac{2}{T}\|\hat{\boldsymbol{A}} - \boldsymbol{A}\|_* \|\boldsymbol{Z}_T\|_{\mathrm{op}}$

$\bar{M}_T = [\int_0^T dM_1(t) \cdots \int_0^T dM_d(t)]^\top$ and $\boldsymbol{Z}_t$ matrix martingale with entries

$$(\boldsymbol{Z}_t)_{j,k} = \int_0^t \int_{(0,s)} h_{j,k}(s-u) dN_k(u) dM_j(s), \tag{1}$$

or

$$\boldsymbol{Z}_t = \int_0^t \mathrm{diag}[dM_s] \boldsymbol{H}_s,$$

with $\boldsymbol{H}_t$ predictable process with entries

$$(\boldsymbol{H}_t)_{j,j'} = \int_{(0,t)} h_{j,j'}(t-s) dN_{j'}(s)$$

Noise term is a matrix-martingale in continuous time:

$$\frac{1}{T} \boldsymbol{Z}_T$$

wee need to control $\frac{1}{T} \|\boldsymbol{Z}_T\|_{\mathrm{op}}$

A consequence of our new concentration inequalities (more after):

$$\mathbb{P}\left[\frac{\|\boldsymbol{Z}_t\|_{\mathrm{op}}}{t} \geq \sqrt{\frac{2v(x+\log(2d))}{t}} + \frac{b(x+\log(2d))}{3t}, \right.$$
$$\left. b_t \leq b, \quad \lambda_{\max}(\boldsymbol{V}_t) \leq v \right] \leq e^{-x},$$

for any $v, x, b > 0$, where

$$\boldsymbol{V}_t = \frac{1}{t}\int_0^t \|\boldsymbol{H}_s\|_{2,\infty}^2 \begin{bmatrix} \mathrm{diag}[\lambda_s^*] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_s^\top \mathrm{diag}[\boldsymbol{H}_s\boldsymbol{H}_s^\top]^{-1} \mathrm{diag}[\lambda_s^*]\boldsymbol{H}_s \end{bmatrix} ds$$

and $b_t = \sup_{s\in[0,t]} \|\boldsymbol{H}_s\|_{2,\infty}$ ($\|\cdot\|_{2,\infty}$ = maximum ℓ_2 row norm)

Useless for statistical learning! Event $\lambda_{\max}(\boldsymbol{V}_t) \leq v$ is annoying and $\boldsymbol{V}_t$ is **not observable** (depends on λ^*)!

Theorem [Something better]. For any $x > 0$, we have

$$\frac{\|\boldsymbol{Z}_t\|_{\mathrm{op}}}{t} \leq 8\sqrt{\frac{(x + \log d + \hat{\ell}_{x,t})\lambda_{\max}(\hat{\boldsymbol{V}}_t)}{t}} + \frac{(x + \log d + \hat{\ell}_{x,t})(10.34 + 2.65 b_t)}{t}$$

with a probability larger than $1 - 84.9e^{-x}$, where

$$\hat{\boldsymbol{V}}_t = \frac{1}{t}\int_0^t \|\boldsymbol{H}_s\|_{2,\infty}^2 \begin{bmatrix} \mathrm{diag}[dN_s] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_s^\top \mathrm{diag}[\boldsymbol{H}_s\boldsymbol{H}_s^\top]^{-1} \mathrm{diag}[dN_s]\boldsymbol{H}_s \end{bmatrix} ds$$

and small ugly term:

$$\hat{\ell}_{x,t} = 4 \log\log\left(\frac{2\lambda_{\max}(\hat{\boldsymbol{V}}_t) + 2(4 + b_t^2/3)x}{x} \vee e\right) + 2\log\log\left(b_t^2 \vee e\right).$$

This is a non-commutative deviation inequality with **observable** variance

- These concentration inequalities leads to a data-driven tuning of penalization
- Solves the **"scaling" problem** in this context $\approx$ **features scaling** in supervised learning

Controls on $\|\boldsymbol{Z}_T\|_\infty = \max_{j,k} |A_{j,k}|$ and $\|\boldsymbol{Z}_T\|_{\text{op}}$ leads to the following tuning of the penalizations

For ℓ_1 penalization of μ: $\|\mu\|_{1,\hat{w}} = \sum_{j=1}^{d} \hat{w}_j |\mu_j|$ with

$$\hat{w}_j = 6\sqrt{2}\sqrt{\frac{(x + \log d + \hat{\ell}_{x,j,T}) N_j([0,T])/T}{T}} + 27.93 \frac{x + \log d + \hat{\ell}_{x,j,T}}{T}$$

where $N_j([0,T]) = \int_0^T dN_j(t)$, namely

$$\hat{w}_j \approx c\sqrt{\frac{N_j([0,T])/T}{T}}$$

- Each coordinate j of μ is penalized (roughly) by $N_j([0,T)]/T$: estimated average intensity of events of node j

For ℓ_1 penalization of $\boldsymbol{A}$: $\|\boldsymbol{A}\|_{1,\hat{\boldsymbol{W}}} = \sum_{1 \le j,k \le d} \hat{\boldsymbol{W}}_{j,k} |\boldsymbol{A}_{j,k}|$ with

$$\hat{\boldsymbol{W}}_{j,k} = 4\sqrt{2}\sqrt{\frac{(x + 2\log d + \hat{\ell}_{x,j,k,T})\hat{\boldsymbol{V}}_{j,k}(T)}{T}} + 18.62\frac{(x + 2\log d + \hat{\ell}_{x,j,k,T})\boldsymbol{B}_{j,k}(T)}{T}$$

where

$$\boldsymbol{B}_{j,k}(t) = \sup_{s \in [0,t]} \int_{(0,t)} h_{j,k}(t-s) dN_k(s)$$

$$\hat{\boldsymbol{V}}_{j,k}(t) = \frac{1}{t}\int_0^t \Big(\int_{(0,s)} h_{j,k}(s-u) dN_k(u) \Big)^2 dN_j(s)$$

namely

$$\hat{\boldsymbol{W}}_{j,k} \approx c\sqrt{\frac{\hat{\boldsymbol{V}}_{j,k}(T)}{T}}$$

$\hat{\boldsymbol{V}}_{j,k}(t)$ estimates the "variance" of self-excitements between nodes j and k

For trace-norm penalization of $\boldsymbol{A}$: $\hat{w}_* \|\boldsymbol{A}\|_*$ with

$$\hat{w}_* = 8\sqrt{\frac{(x + \log d + \hat{\ell}_{x,T})\lambda_{\max}(\hat{\boldsymbol{V}}_T)}{T}} + \frac{2(x + \log d + \hat{\ell}_{x,T})(10.34 + 2.65 b_t)}{T}$$

namely

$$\hat{w}_* \approx \sqrt{\frac{\lambda_{\max}(\hat{\boldsymbol{V}}_T)}{T}}$$

- Data-driven weights that comes from "empirical" Bernstein's inequalities, entrywise and for operator norm of $\boldsymbol{Z}_T$
- $\hat{\boldsymbol{V}}_{j,k}(t)$ and $\lambda_{\max}(\hat{\boldsymbol{V}}_t)$ are estimations (based on optional variation) of the variance terms from Bernstein's inequality
- $\boldsymbol{B}_{j,k}(t)$ and b_t are L^∞ terms (sub-exponential actually) from these Bernstein's inequalities
- Leads to a data-driven scaling of penalization: deals correctly with the inhomogeneity of information over nodes

A sharp oracle inequality

- Recall $\langle \lambda_1, \lambda_2 \rangle_T = \frac{1}{T} \sum_{j=1}^{d} \int_0^T \lambda_{1,j}(t)\lambda_{2,j}(t)dt$ and $\|\lambda\|_T^2 = \langle \lambda, \lambda \rangle_T$
- Assume RE in our setting (Restricted Eigenvalues), which is a standard assumption to obtain fast rates for the Lasso (and other convex-relaxation based procedures)

Theorem. We have

$$\|\lambda_{\hat{\theta}} - \lambda^*\|_T^2 \leq \inf_{\theta} \left\{ \|\lambda_\theta - \lambda^*\|_T^2 + \kappa(\theta)^2 \left(\frac{5}{4} \|(\hat{w})_{\mathsf{supp}(\mu)}\|_2^2 + \frac{9}{8} \|(\hat{\boldsymbol{W}})_{\mathsf{supp}(\boldsymbol{A})}\|_F^2 + \frac{9}{8} \hat{w}_*^2 \, \mathsf{rank}(\boldsymbol{A}) \right) \right\}$$

with a probability larger than $1 - 146e^{-x}$.

- Leading constant 1

Roughly, $\hat{\theta}$ achieves an optimal tradeoff between approximation and complexity given by

$$\begin{aligned}
&\frac{\|\mu\|_0(x + \log d)}{T} \max_j N_j([0, T])/T \\
&\quad + \frac{\|\boldsymbol{A}\|_0(x + 2\log d)}{T} \max_{j,k} \hat{\boldsymbol{V}}_{j,k}(T) \\
&\quad + \frac{\mathrm{rank}(A)(x + \log d)}{T} \lambda_{\max}(\hat{\boldsymbol{V}}_T)
\end{aligned}$$

- Complexity measured both by sparsity and rank
- Convergence has shape $(\log d)/T$, where $T =$ length of the observation interval
- These terms are balanced by "empirical variance" terms

Main tool: new concentration inequalities for matrix martingales in continuous time

Introduce

$$\boldsymbol{Z}_t = \int_0^t \boldsymbol{A}_s(\boldsymbol{C}_s \odot d\boldsymbol{M}_s)\boldsymbol{B}_s,$$

where $\{\boldsymbol{A}_t\}$, $\{\boldsymbol{C}_t\}$ and $\{\boldsymbol{B}_t\}$ predictable and where $\{\boldsymbol{M}_t\}_{t\geq 0}$ is a "white" matrix martingale, in the sense that $[\mathrm{vec}\boldsymbol{M}]_t$ is diagonal

NB: entries of $\boldsymbol{Z}_t$ are given by

$$(\boldsymbol{Z}_t)_{i,j} = \sum_{k=1}^{p}\sum_{l=1}^{q}\int_0^t (\boldsymbol{A}_s)_{i,k}(\boldsymbol{C}_s)_{k,l}(\boldsymbol{B}_s)_{l,j}(d\boldsymbol{M}_s)_{k,l}.$$

- $\langle \boldsymbol{M} \rangle_t$ = entrywise predictable quadratic variation, so that

$$\boldsymbol{M}_t^{\odot 2} - \langle \boldsymbol{M} \rangle_t$$

  martingale
- vectorization operator $\mathrm{vec} : \mathbb{R}^{p \times q} \to \mathbb{R}^{pq}$ stacks vertically the columns of $\boldsymbol{X}$
- $\langle \mathrm{vec} \boldsymbol{M} \rangle_t$ is the $pq \times pq$ matrix with entries that are all pairwise quadratic covariations, so that

$$\mathrm{vec}(\boldsymbol{M}_t)\mathrm{vec}(\boldsymbol{M}_t)^\top - \langle \mathrm{vec} \boldsymbol{M} \rangle_t$$

  is a martingale.

- $\boldsymbol{M}_t = \boldsymbol{M}_t^c + \boldsymbol{M}_t^d$, where $\boldsymbol{M}_t^c$ is a continuous martingale and $\boldsymbol{M}_t^d$ is a purely discountinuous martingale. Its (entrywise) quadratic variation is defined as

$$[\boldsymbol{M}]_t = \langle \boldsymbol{M}^c \rangle_t + \sum_{0 \le s \le t} (\Delta \boldsymbol{M}_t)^2, \tag{2}$$

and its quadratic covariation by

$$[\mathrm{vec}\boldsymbol{M}]_t = \langle \mathrm{vec}\boldsymbol{M}^c \rangle_t + \sum_{0 \le s \le t} \mathrm{vec}(\Delta \boldsymbol{M}_s)\mathrm{vec}(\Delta \boldsymbol{M}_s)^\top.$$

We say that $\boldsymbol{M}$ is *purely discontinuous* if the process $\langle \mathrm{vec}\boldsymbol{M}^c \rangle_t$ is identically the zero matrix.

Concentration for purely discountinuous matrix martingale:

- $\boldsymbol{M}_t$ is purely discountinuous and we have

$$\langle \boldsymbol{M} \rangle_t = \int_0^t \boldsymbol{\lambda}_s ds$$

  for a non-negative and predictable intensity process $\{\boldsymbol{\lambda}_t\}_{t \geq 0}$.

- Standard moment assumptions (subexponential tails)

Introduce

$$\boldsymbol{V}_t = \int_0^t \|\boldsymbol{A}_s\|_{\infty,2}^2 \|\boldsymbol{B}_s\|_{2,\infty}^2 \boldsymbol{W}_s ds$$

where

$$\boldsymbol{W}_t = \begin{bmatrix} \boldsymbol{W}_t^1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{W}_t^2 \end{bmatrix}, \tag{3}$$

$$\boldsymbol{W}_t^1 = \boldsymbol{A}_t \operatorname{diag}[\boldsymbol{A}_t^\top \boldsymbol{A}_t]^{-1} \operatorname{diag}\left[(\boldsymbol{C}_t^{\odot 2} \odot \boldsymbol{\lambda}_t)\mathbb{1}\right] \boldsymbol{A}_t^\top$$
$$\boldsymbol{W}_t^2 = \boldsymbol{B}_t^\top \operatorname{diag}[\boldsymbol{B}_t \boldsymbol{B}_t^\top]^{-1} \operatorname{diag}\left[(\boldsymbol{C}_t^{\odot 2} \odot \boldsymbol{\lambda}_t)^\top \mathbb{1}\right] \boldsymbol{B}_t$$

Introduce also

$$b_t = \sup_{s\in[0,t]} \|\boldsymbol{A}_s\|_{\infty,2}\|\boldsymbol{B}_s\|_{2,\infty}\|\boldsymbol{C}_s\|_{\infty}.$$

Theorem.

$$\mathbb{P}\Bigg[\|\boldsymbol{Z}_t\|_{\mathrm{op}} \geq \sqrt{2v(x+\log(m+n))} + \frac{b(x+\log(m+n))}{3},$$
$$b_t \leq b, \quad \lambda_{\max}(\boldsymbol{V}_t) \leq v\Bigg] \leq e^{-x},$$

- First result of this type for matrix-martingale in continuous time

Corollary. $\{\boldsymbol{N}_t\}$ a $p \times q$ matrix, each $(\boldsymbol{N}_t)_{i,j}$ is an independent inhomogeneous Poisson processes with intensity $(\boldsymbol{\lambda}_t)_{i,j}$. Consider the martingale $\boldsymbol{M}_t = \boldsymbol{N}_t - \boldsymbol{\Lambda}_t$, where $\boldsymbol{\Lambda}_t = \int_0^t \boldsymbol{\lambda}_s ds$ and let $\{\boldsymbol{C}_t\}$ be deterministic and bounded. We have

$$\begin{aligned}
&\left\| \int_0^t \boldsymbol{C}_s \odot d(\boldsymbol{N}_t - \boldsymbol{\Lambda}_t) \right\|_{\mathrm{op}} \\
&\quad \leq \sqrt{2\left(\left\| \int_0^t \boldsymbol{C}_s^{\odot 2} \odot \boldsymbol{\lambda}_s ds \right\|_{1,\infty} \vee \left\| \int_0^t \boldsymbol{C}_s^{\odot 2} \odot \boldsymbol{\lambda}_s ds \right\|_{\infty,1} \right)(x + \log(p+q))} \\
&\qquad + \frac{\sup_{s\in[0,t]} \|\boldsymbol{C}_s\|_\infty (x + \log(p+q))}{3}
\end{aligned}$$

holds with a probability larger than $1 - e^{-x}$.

Corollary. Even more particualar: $\boldsymbol{N}$ random matrix where $\boldsymbol{N}_{i,j}$ are independent Poisson variables with intensity $\boldsymbol{\lambda}_{i,j}$. We have

$$\|\boldsymbol{N} - \boldsymbol{\lambda}\|_{\mathrm{op}} \leq \sqrt{2(\|\boldsymbol{\lambda}\|_{1,\infty} \vee \|\boldsymbol{\lambda}\|_{\infty,1})(x + \log(p+q))} + \frac{x + \log(p+q)}{3}.$$

- Up to our knowledge, not previously stated in literature
- NB: In the Gaussian case: variance depends on maximum ℓ_2 norm of rows and columns (cf. Tropp (2011))

- We have as well a non-commutative Hoeffding's inequality when $\boldsymbol{M}_t$ has continuous paths (allowing Itô's formula...), with a similar variance term
- Tools from stochastic calculus, use of the dilation operator and some classical matrix inequalities about the trace exponential and the SDP order.

A difficult proposition: a control of the quadratic variation of the pure jump process

$$\boldsymbol{U}_t^u = \sum_{0 \leq s \leq t} \left(e^{u \Delta \mathscr{S}(\boldsymbol{Z}_s)} - u \Delta \mathscr{S}(\boldsymbol{Z}_s) - \boldsymbol{I} \right)$$

given by

$$\langle \boldsymbol{U}^{\xi} \rangle_t \preceq \int_0^t \frac{\varphi\left(\xi \|\boldsymbol{A}_s\|_{\infty,2} \|\boldsymbol{B}_s\|_{2,\infty} \|\boldsymbol{C}_s\|_{\infty}\right)}{\|\boldsymbol{C}_s\|_{\infty}^2} \boldsymbol{W}_s ds,$$

where $\varphi(x) = e^x - x - 1$.

Conclusion

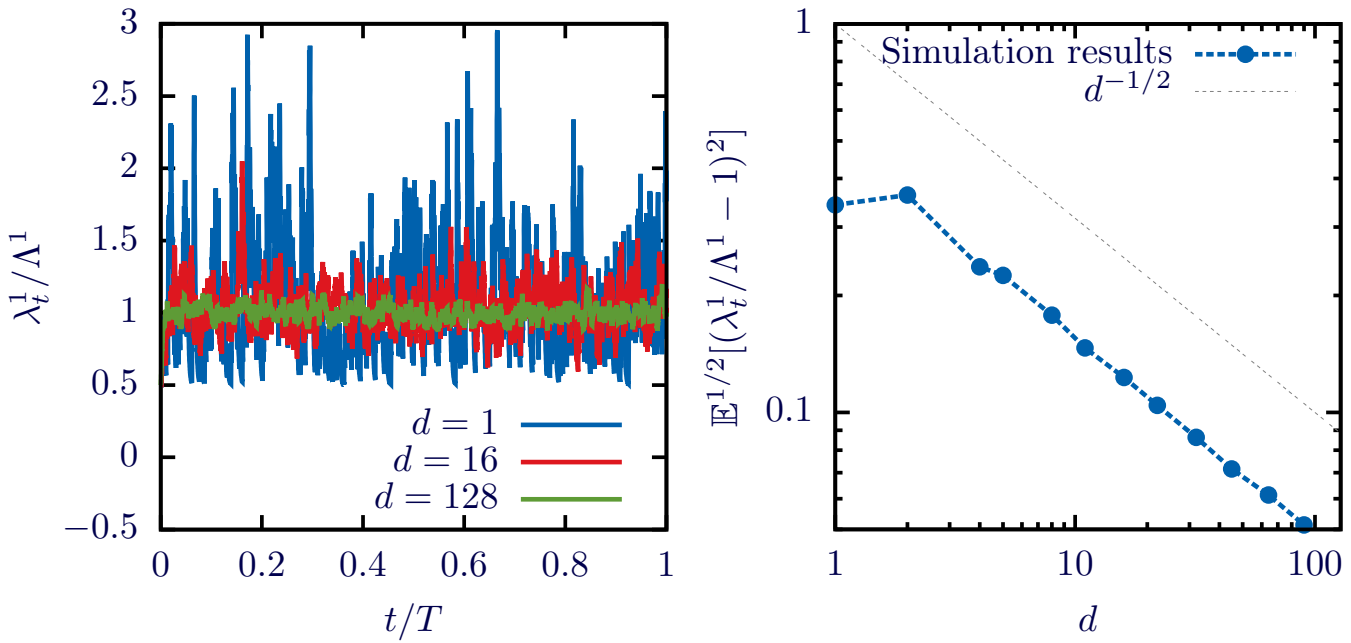
- Theoretical study of learning algorithms for "time-oriented" models needs new probabilistic results
- In our case new concentration results for matrix martingales in continuous time
- Leads to a better scaling of penalizations

A word about something else...

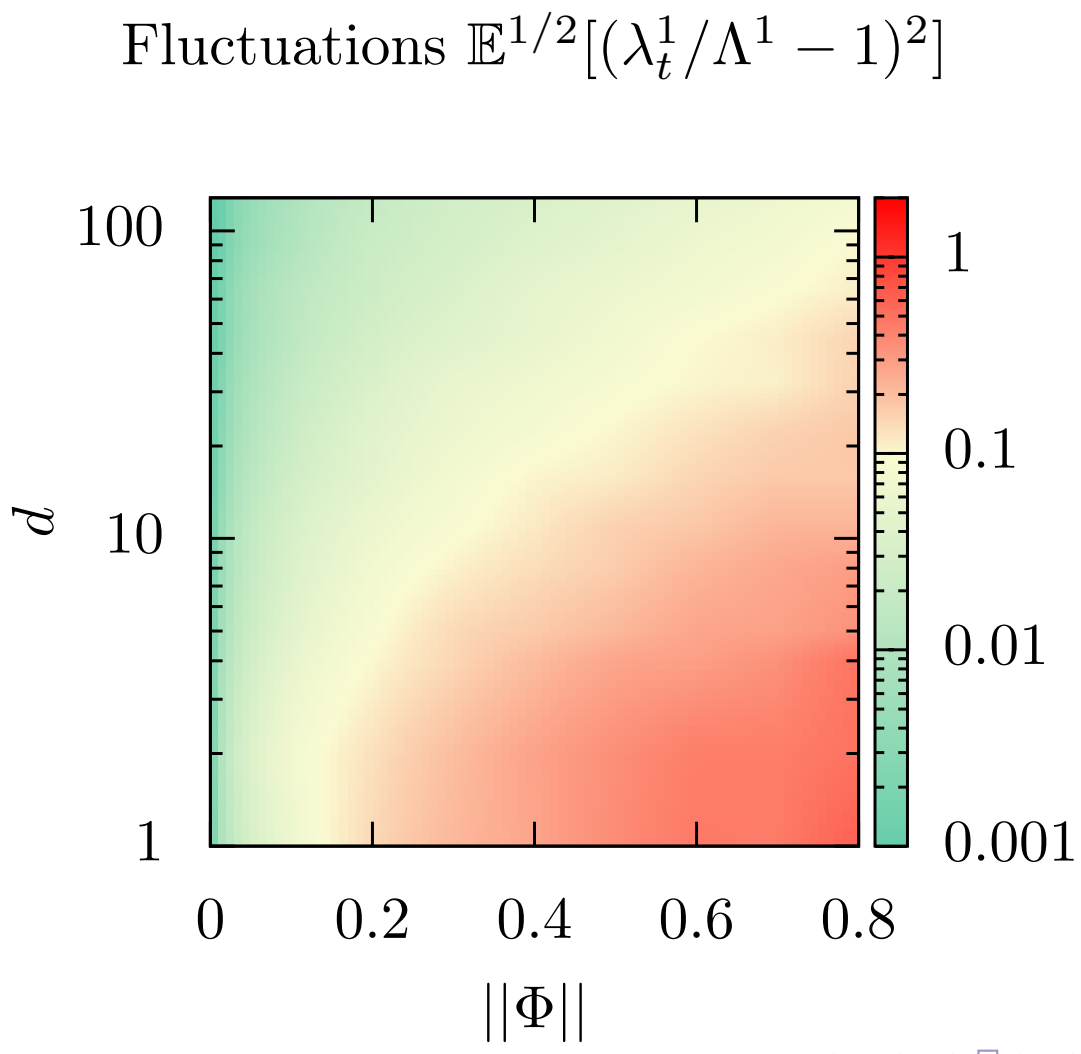
Going back to maximum-likelihood estimation, with d very large

- For inference, exploit the fact that d is large

$\Rightarrow$ use a Mean-Field approximation! (from Delattre et al. 2015)

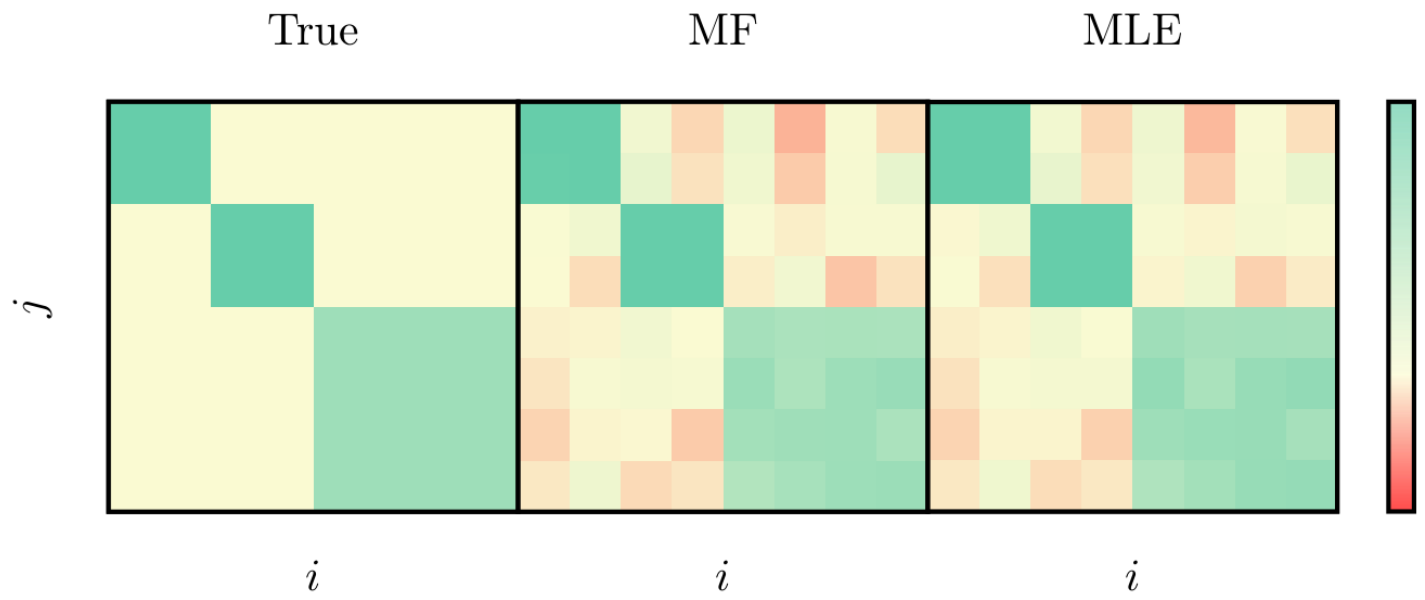


We don't understand perfectly why this works yet (proof on a toy example)

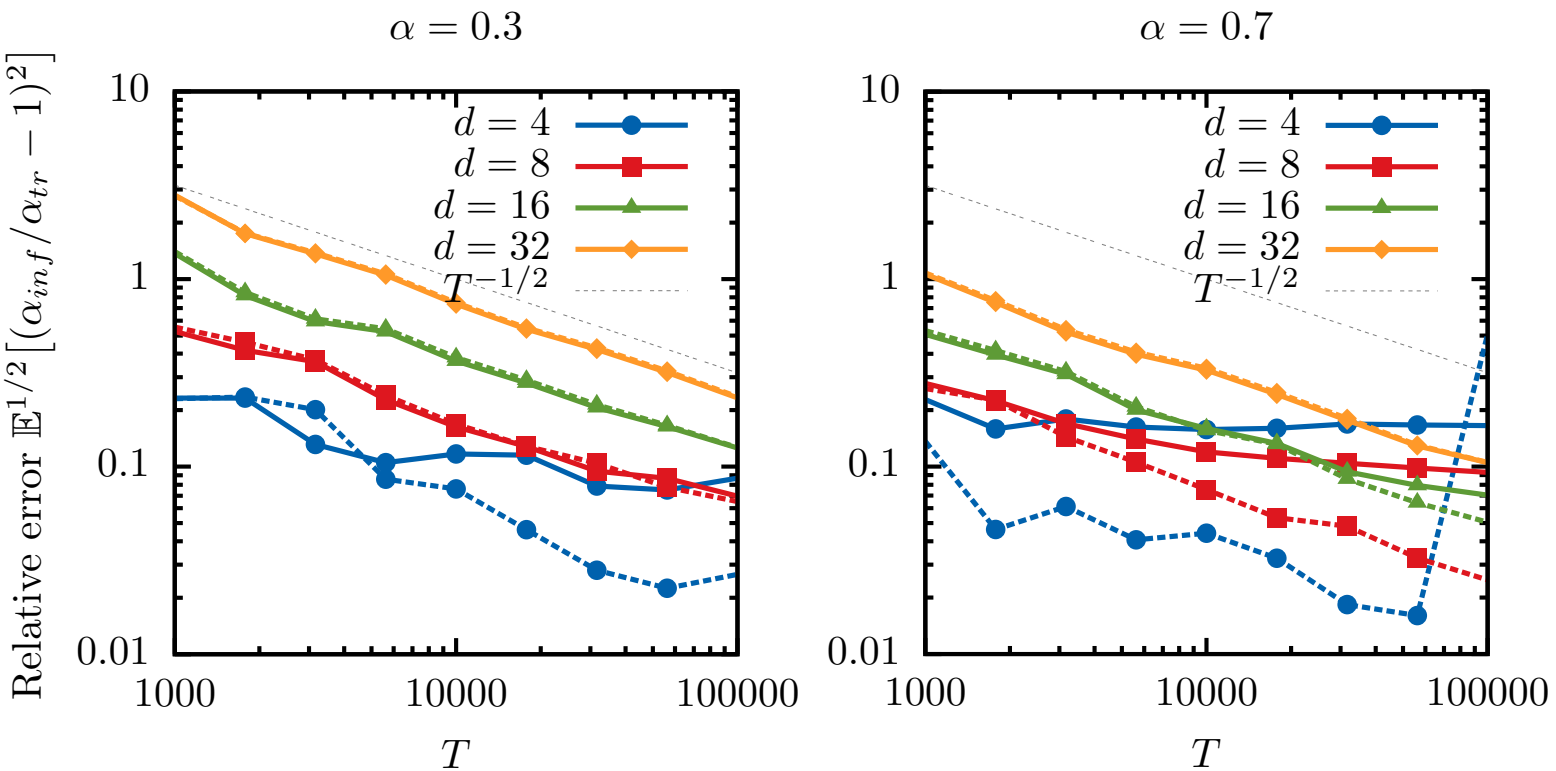


Mean-field inference for Hawkes

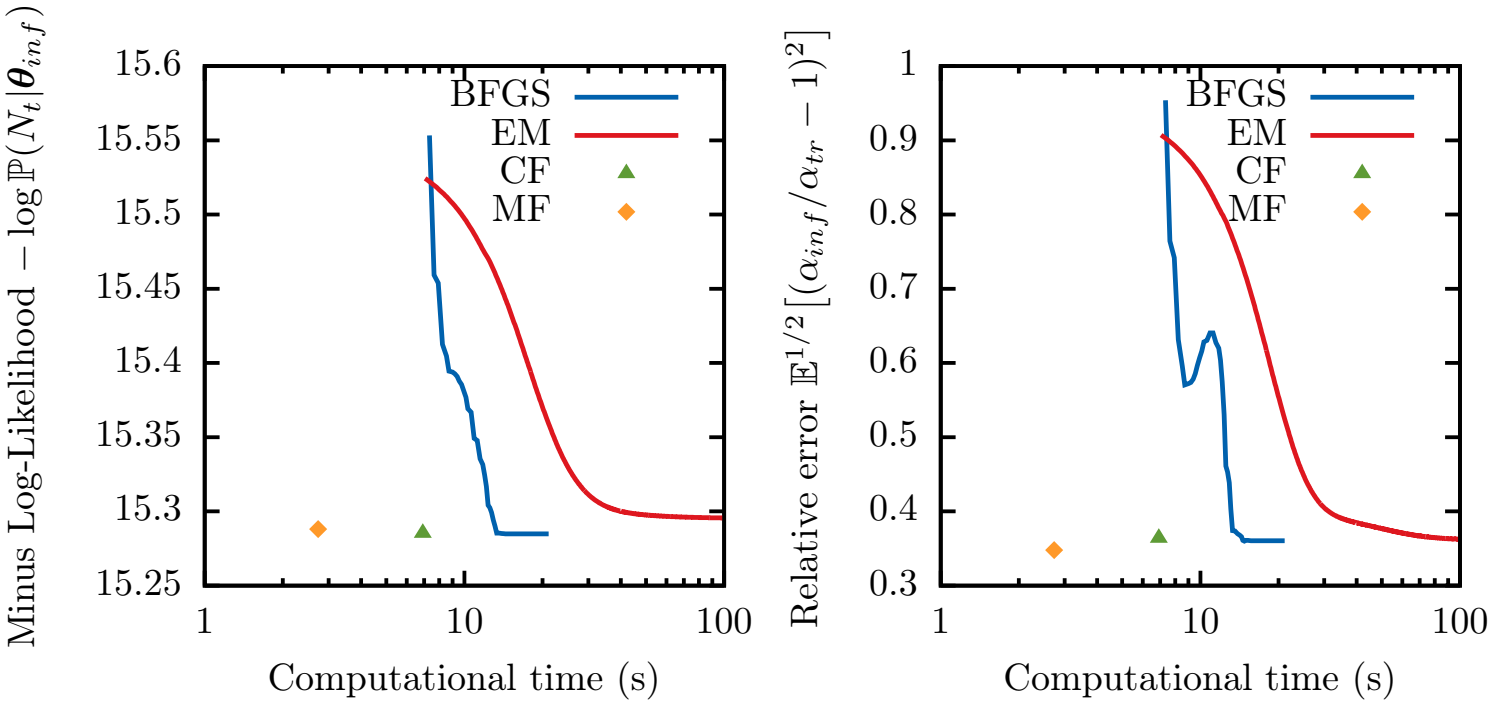
But it does very well empirically



But it does very well empirically



And it is faster by several order of magnitude than state-of-the-art solvers



Perspectives

- Better understanding of the Mean-Field based inference
- Quantifying influence without kernels estimation
- Experiments: ongoing project with Préfecture de l'Oise (car theft), High-frequency Finance
- Non-constant baseline (block-stationarity): block constant with total-variation penalization
- NMF based learning of the self-excitement matrix
- SBM prior for community detection results
- Even better optimization using recent stochastic gradient algorithms (but needs some tweaking...)