Learning Stochastic Differential Equations

José Bento

Join work with Andrea Montanari and Morteza Ibraimi

NIPS 2015

Estimation

$$\mathbf{P}_\theta(x) \longrightarrow X_1^n = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \longrightarrow \hat{\theta} = \hat{\theta}(X_1^n)$$

$$\theta \overset{?}{\approx} \hat{\theta}$$

Estimation

$$\mathbf{P}_\theta(x) \longrightarrow X_1^n = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \longrightarrow \hat{\theta} = \hat{\theta}(X_1^n)$$

$$\theta \overset{?}{\approx} \hat{\theta}$$

Fundamental problem in many engineering apps.

- Signal processing
- Quality control
- Control theory

Estimation

$$\mathbf{P}_\theta(x) \longrightarrow X_1^n = \{x^{(1)}, x^{(2)}, \dots, x^{(n)}\} \longrightarrow \hat{\theta} = \hat{\theta}(X_1^n)$$

$$\theta \overset{?}{\approx} \hat{\theta}$$

Classical setting

- Data = i.i.d. samples from $\mathbf{P}_\theta(x)$
- θ = low-dimensional parameter
- n = large value compared to $\dim(\theta)$

Estimation

$$\mathbf{P}_\theta(x) \longrightarrow X_1^n = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\} \longrightarrow \hat{\theta} = \hat{\theta}(X_1^n)$$

$$\theta \overset{?}{\approx} \hat{\theta}$$

Classical setting

E.g.: Learn θ in $y_\ell = \langle x_\ell, \theta \rangle + \varepsilon_\ell$ from i.i.d. $\{(y_\ell, x_\ell)\}_{\ell=0}^n$

In this talk...

Data = trajectory of a stochastic diff. eq. (SDE)

In this talk...

Data = trajectory of a stochastic diff. eq. (SDE)

$$\{x(t)\}_{t=0}^{T}: \qquad \mathrm{d}x(t) = F(x(t);\theta)\mathrm{d}t + \mathrm{d}b(t)$$

In this talk...

Data = trajectory of a stochastic diff. eq. (SDE)

$$\{x(t)\}_{t=0}^{T}: \qquad \mathrm{d}x(t) = F(x(t);\theta)\mathrm{d}t + \mathrm{d}b(t)$$

θ = very large graph

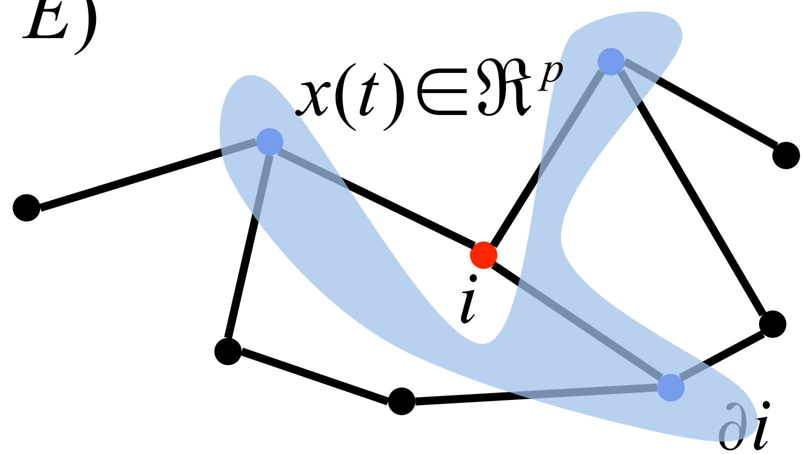
In this talk...

Data = trajectory of a stochastic diff. eq. (SDE)

$$\{x(t)\}_{t=0}^{T}: \qquad \mathrm{d}x(t) = F(x(t);\theta)\mathrm{d}t + \mathrm{d}b(t)$$

θ = very large graph, $G=(V,E)$

E.g.: G = interaction between components of x



$$\mathrm{d}x_i(t) = -mx_i(\mathrm{t})\mathrm{d}t + \sum_{j\in\partial i}(x_j(t)-x_i(t))\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

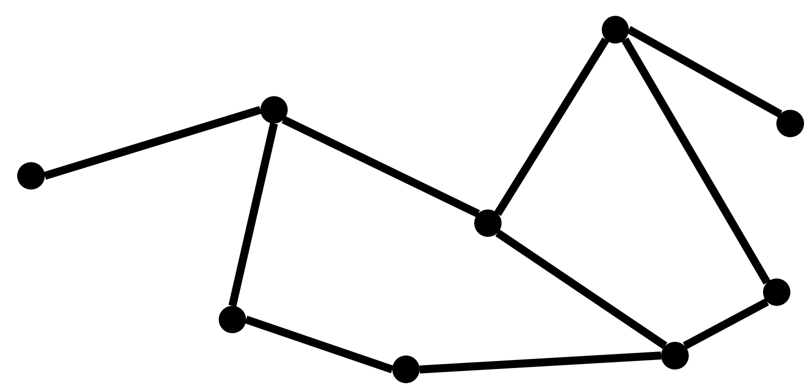
In this talk...

Data = trajectory of a stochastic diff. eq. (SDE)

$$\{x(t)\}_{t=0}^{T} : \qquad \mathrm{d}x(t) = F(x(t);\theta)\mathrm{d}t + \mathrm{d}b(t)$$

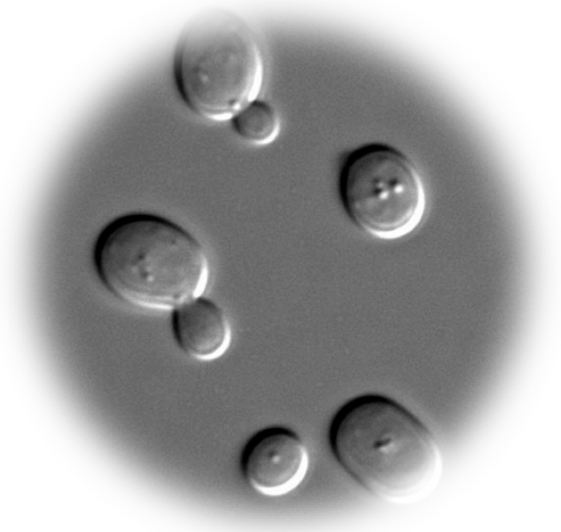
θ = very large graph

E.g.: G = interaction between components of x



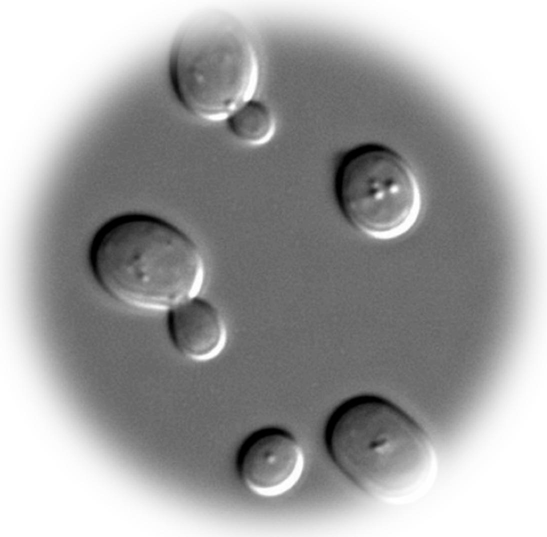
T = minimum value to recover graph with prob. ≈ 1

Graphs & SDEs
Example: Gene regulatory networks



Yeast

Example: Gene regulatory networks

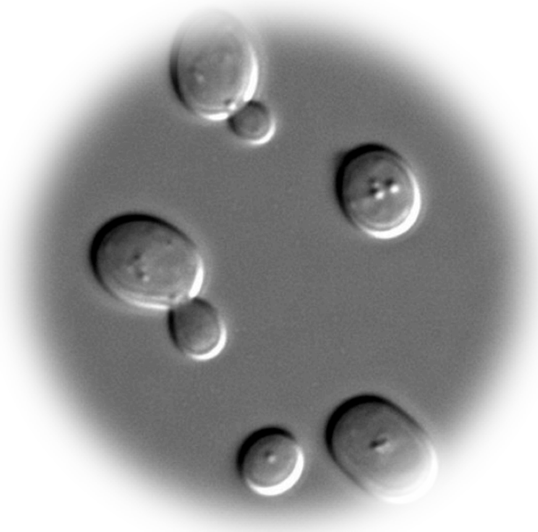


Yeast

'18srRnaa'	'biod5'	'THR5'	'YER022w5'	'YAL058W/CNE1'	'YAL042W/'	'YAL031C/FUN21'
'18srRnab'	'biodm'	'THR3'	'YER022wM'	'YAL058C-a/'	'YAL043C-a/'	'YAL030W/
'18srRnac'	'biod3'	'TRP5'	'YER022w3'	'YAL056W/'	'YAL041W/CDC24'	SNC1_ex1'
'18srRnad'	'cre5'	'TRPM'	'YAL069W/'	'YAL055W/'	'YAL040C/CLN3'	'YAL030W/
'18srRnae'	'crem'	'TRP3'	'YAL067C/SEO1'	'YAL054C/ACS1'	'YAL039C/CYC3'	SNC1_ex2'
'25srRnaa'	'cre3'	'DAP5'	'YAL066W/'	'YAL053W/'	'YAL038W/CDC19'	'YAL029C/MYO4'
'25srRnab'	'25srRnad'	'DAPM'	'YAL065C/'	'YAL051W/'	'YAL037W/'	'YAL028W/'
'25srRnac'	'25srRnae'	'DAP3'	'YAL065C-a/'	'YAL049C/'	'YAL036C/'	'YAL027W/'
'BIOB5'	'LYSA5'	'YFL039C5'	'YAL064W/FLO9_i'	'YAL048C/'	'YAL035W/FUN12'	'YAL026C/DRS2'
'biobm'	'LYSAM'	'YFL039CM'	'YAL063C/'	'YAL047C/'	'YAL034W-a/'	'YAL025C/MAK16'
'biob3'	'LYSA3'	'YFL039C3'	'YAL062W/GDH3'	'YAL046C/'	'YAL035C-a/'	'YAL024C/LTE1'
'bioc5'	'PHE5'	'YER148w5'	'YAL061W/'	'YAL045C/'	'YAL034C/FUN19'	'YAL023C/PMT2'
'biocm'	'PHEM'	'YER148wM'	'YAL060W/'	'YAL044C/GCV3'	'YAL033W/FUN53'	'YAL022C/'
'bioc3'	'PHE3'	'YER148w3'	'YAL059W/SIM1'	'YAL043C/PTA1'	'YAL032C/FUN20'	'YAL021C/CCR4'

≈ 6000 genes

Example: Gene regulatory networks

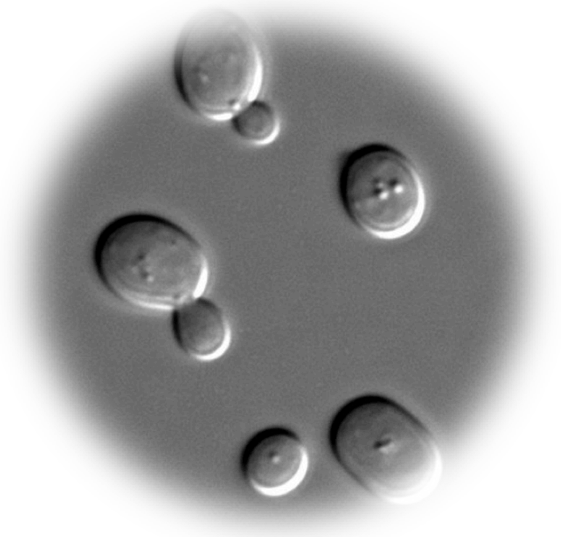


Yeast

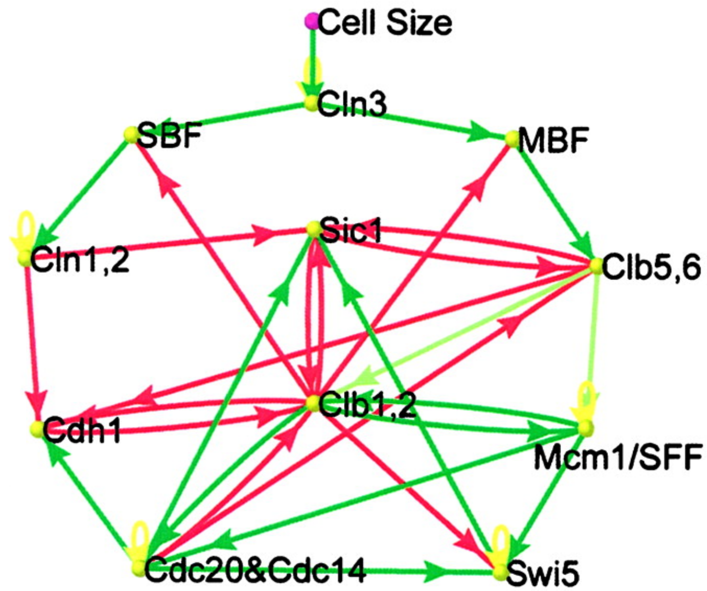
'18srRnaa'	'biod5'	'THR5'	'YER022w5'	'YAL058W/CNE1'	'YAL042W/'	'YAL031C/FUN21'
'18srRnab'	'biodm'	'THR3'	'YER022wM'	'YAL058C-a/'	'YAL043C-a/'	'YAL030W/
'18srRnac'	'biod3'	'TRP5'	'YER022w3'	'YAL056W/'	'YAL041W/CDC24'	SNC1_ex1'
'18srRnad'	'cre5'	'TRPM'	'YAL069W/'	'YAL055W/'	'YAL040C/CLN3'	'YAL030W/
'18srRnae'	'crem'	'TRP3'	'YAL067C/SEO1'	'YAL054C/ACS1'	'YAL039C/CYC3'	SNC1_ex2'
'25srRnaa'	'cre3'	'DAP5'	'YAL066W/'	'YAL053W/'	'YAL038W/CDC19'	'YAL029C/MYO4'
'25srRnab'	'25srRnad'	'DAPM'	'YAL065C/'	'YAL051W/'	'YAL037W/'	'YAL028W/'
'25srRnac'	'25srRnae'	'DAP3'	'YAL065C-a/'	'YAL049C/'	'YAL036C/'	'YAL027W/'
'BIOB5'	'LYSA5'	'YFL039C5'	'YAL064W/FLO9_i'	'YAL048C/'	'YAL035W/FUN12'	'YAL026C/DRS2'
'biobm'	'LYSAM'	'YFL039CM'	'YAL063C/'	'YAL047C/'	'YAL034W-a/'	'YAL025C/MAK16'
'biob3'	'LYSA3'	'YFL039C3'	'YAL062W/GDH3'	'YAL046C/'	'YAL035C-a/'	'YAL024C/LTE1'
'bioc5'	'PHE5'	'YER148w5'	'YAL061W/'	'YAL045C/'	'YAL034C/FUN19'	'YAL023C/PMT2'
'biocm'	'PHEM'	'YER148wM'	'YAL060W/'	'YAL044C/GCV3'	'YAL033W/FUN53'	'YAL022C/'
'bioc3'	'PHE3'	'YER148w3'	'YAL059W/SIM1'	'YAL043C/PTA1'	'YAL032C/FUN20'	'YAL021C/CCR4'

≈ 800 genes for
cell cycle
[Spellman 98]

Example: Gene regulatory networks



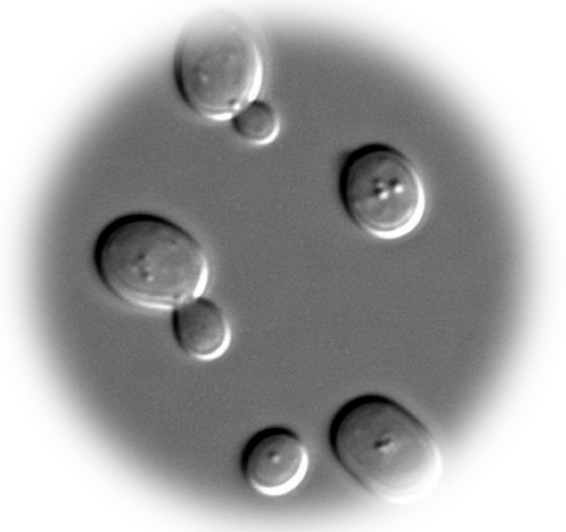
Yeast



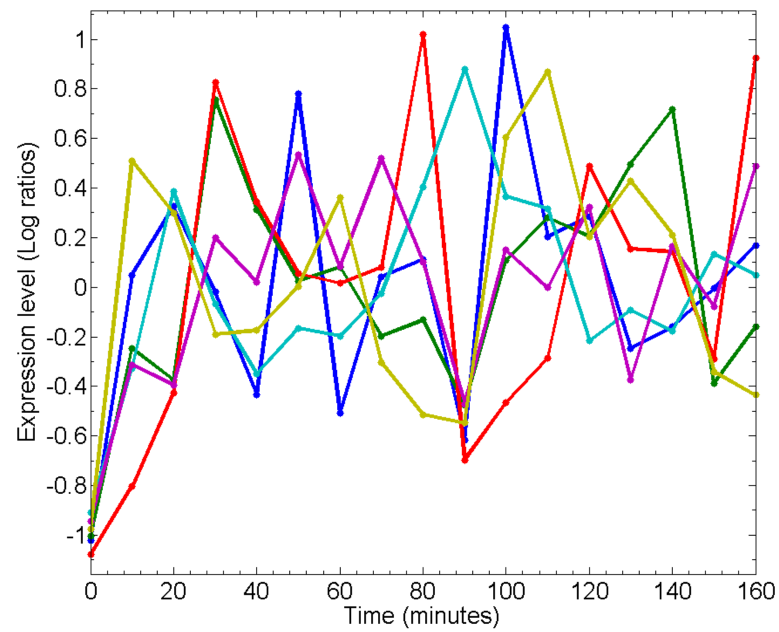
E.g. of cell cycle gene net.
[Li et al. 04]

Nodes = genes
Edges = interaction (activation/repression)

Example: Gene regulatory networks



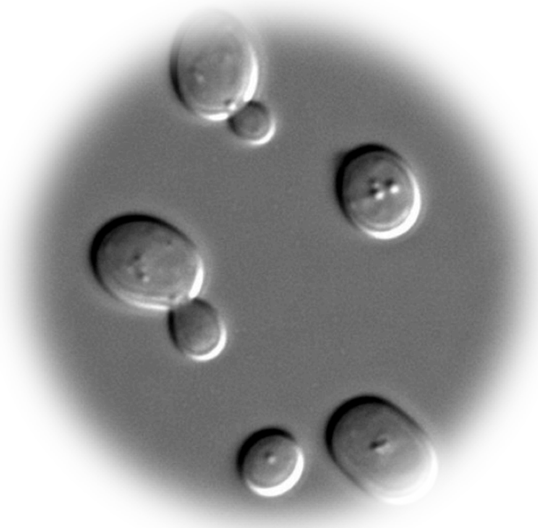
Yeast



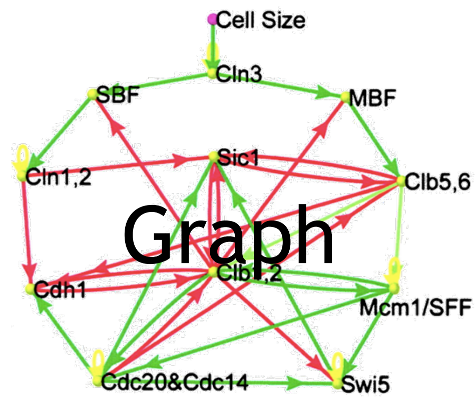
Gene expression t. series
[Stanford yeast cell cycle database]

$x_i(t)$ = i^{th} gene expression level at time t

Example: Gene regulatory networks

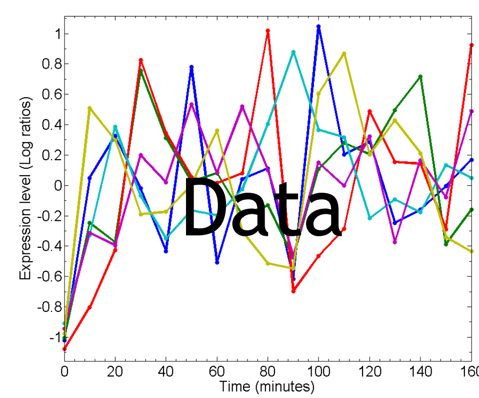


Yeast



Graph

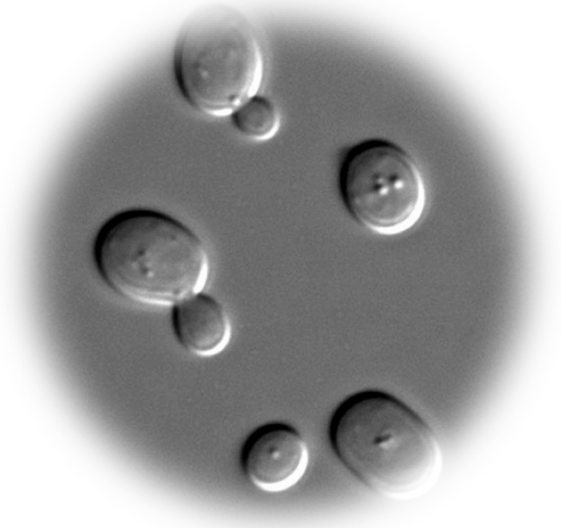
Model



Data

E.g.: Chen 05 et al. uses SDEs to study yeast cell-cycle

Example: Gene regulatory networks



Yeast

$$A + B \overset{k',k}{\Leftrightarrow} C + D$$

$$\mathrm{d}x_{\mathrm{A}} = \left(k' x_C x_D - k \ x_A x_B\right)\mathrm{d}t + \mathrm{d}b$$

Gene interaction

⇔

Bio-chemical reactions

Example: Gene regulatory networks

Simple model for gene interaction

$$dx_i = \begin{bmatrix} \theta_{i,0} & \theta_{i,1} & \theta_{i,2} & \dots & \theta_{i,12} & \theta_{i,23} & \dots \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \dots \\ x_1 x_2 \\ x_2 x_3 \\ \dots \end{bmatrix} dt + db_i$$

$\theta_{i,11} \neq 0 \Leftrightarrow$ **2** and **3** interact to regulate ***i***

Example: Gene regulatory networks

$$\mathrm{d}x_i = \begin{bmatrix} \theta_{i,0} & \theta_{i,1} & \theta_{i,2} & \dots & \theta_{i,12} & \theta_{i,23} & \dots \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \dots \\ x_1 x_2 \\ x_2 x_3 \\ \dots \end{bmatrix} \mathrm{d}t + \mathrm{d}b_i$$

Learning $\mathrm{supp}(\theta)$

$\Updownarrow$

Learning network of gene interactions

Example: Gene regulatory networks

$$\mathrm{d}x_i = \begin{bmatrix} \theta_{i,0} & \theta_{i,1} & \theta_{i,2} & \dots & \theta_{i,12} & \theta_{i,23} & \dots \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \dots \\ x_1 x_2 \\ x_2 x_3 \\ \dots \end{bmatrix} \mathrm{d}t + \mathrm{d}b_i$$

Problem:

1. How to fit this (or other SDEs models) to data?
2. How much data do we need?

Our problem

$$\mathrm{d}x(t) = F(x(t);\theta)\mathrm{d}t + \mathrm{d}b(t)$$

Large graph, G

$$\{x(t)\}_{t=0}^{T} \Rightarrow \boxed{\text{Alg}} \Rightarrow \hat{G} \overset{?}{=} G$$

Quantity of interest

$$T_{\mathrm{Alg},G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\}$$

Our problem

Quantity of interest

$$T_{Alg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim 0 \ ??$$

Our problem

Quantity of interest

$$T_{A\lg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim 0 \text{ ??}$$

$$T_{A\lg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim p \text{ ??}$$

Our problem

Quantity of interest

$$T_{A\lg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim 0 \ ??$$

$$T_{A\lg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim p \ ??$$

Fact: Learning a sparse θ in $y_\ell = \left\langle x_\ell, \theta \right\rangle + \varepsilon_\ell$ from

i.i.d. pairs $\{y_\ell, x_\ell\}_{\ell=0}^{n}$ is possible if $n \sim \log p$

Our problem

Quantity of interest

$$T_{Alg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim 0 \text{ ??}$$

$$T_{Alg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim p \text{ ??}$$

$$T_{Alg,G} = \inf\left\{T : \mathbf{P}(\hat{G} = G) > 1 - \delta\right\} \sim \log p \text{ ??}$$

Linear SDEs & graphical models

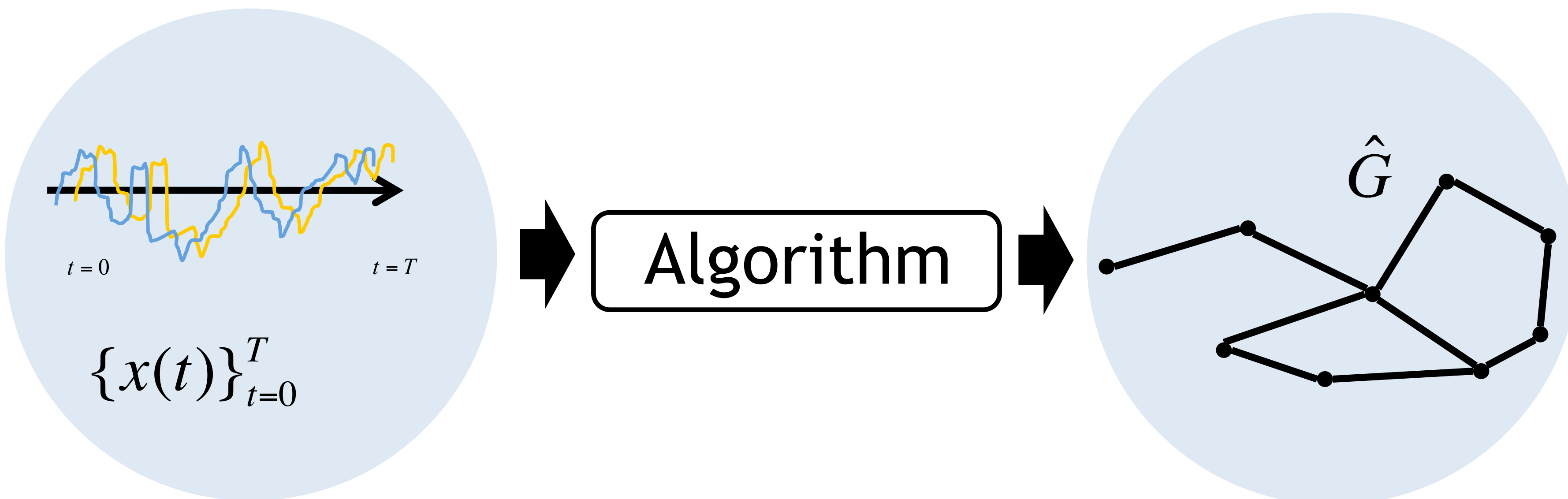
In linear SDEs,

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_\mathrm{i}(t)$$

θ encodes the interaction among components of x

Linear SDEs & graphical models

Learning SDEs is related to the broader problem of learning a graphical model from data



Prior work: Graphical Models

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

Stationary distribution

$$x(t) \sim N(0, \Sigma)$$

$$\theta\Sigma + \Sigma\theta^T + I = 0$$

Prior work: Graphical Models

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

Stationary distribution

$$x(t) \sim N(0, \Sigma)$$

$$\theta\Sigma + \Sigma\theta^T + I = 0$$

Given n i.i.d samples from $N(0, \Sigma)$ estimate

$$\hat{\mathrm{G}} = \mathrm{supp}(\Sigma^{-1})$$

Prior work: Graphical Models

Given n i.i.d samples from $N(0, \Sigma)$ estimate

$$\hat{G} = \mathrm{supp}(\Sigma^{-1})$$

Learning Gaussian graphical models

- Friedman, Hastie, & Tibshirani 08
- Meinshausen & Buhlmann 06

Prior work: Graphical Models

Given n i.i.d samples from $N(0, \Sigma)$ estimate

$$\hat{G} = \mathrm{supp}(\Sigma^{-1})$$

Why is our work different?

- Σ has less information than θ
- We have $\{x(t)\}$, not i.i.d. samples

Prior work: Graphical Models

Given n i.i.d samples from $N(0, \Sigma)$ estimate

$$\hat{G} = \mathrm{supp}(\Sigma^{-1})$$

Σ has less information than θ

$$\Theta_1 = \begin{bmatrix} -2 & 1 & 0 \\ -1 & -2 & 1 \\ 0 & -1 & -2 \end{bmatrix} \qquad \Theta_2 = \begin{bmatrix} -2 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

Prior work: Graphical Models

The system of SDEs is linear in θ

$$\mathrm{d}x_i(t) = \sum_j \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_i(t)$$

$$Y = X\theta_i \qquad \varepsilon$$

$$Y = \begin{bmatrix} \Delta x_i^{(1)} \\ \vdots \\ \Delta x_i^{(n)} \end{bmatrix} \qquad X = \begin{bmatrix} x_1^{(1)} & \cdots & x_p^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_p^{(n)} \end{bmatrix}$$

Prior work: Graphical Models

The system of SDEs is linear in θ

$$\mathrm{d}x_i(t) = \sum_j \theta_{ij} x_j(t)\,\mathrm{d}t + \mathrm{d}b_\mathrm{i}(t)$$

$$Y = X\theta_i \quad \varepsilon$$

Given Y, X satisfying $Y = X\,\theta_i + \varepsilon$ estimate

$$\partial \mathrm{i} = \mathrm{supp}(\theta_i)$$

Prior work: Graphical Models

Given Y, X satisfying $Y = X\theta_i + \varepsilon$ estimate

$$\partial \mathrm{i} = \mathrm{supp}(\theta_i)$$

L_1 – regularized least squares

- Tibshirani 96
- Zhao & Yu 06
- Wainwright 09

Prior work: Graphical Models

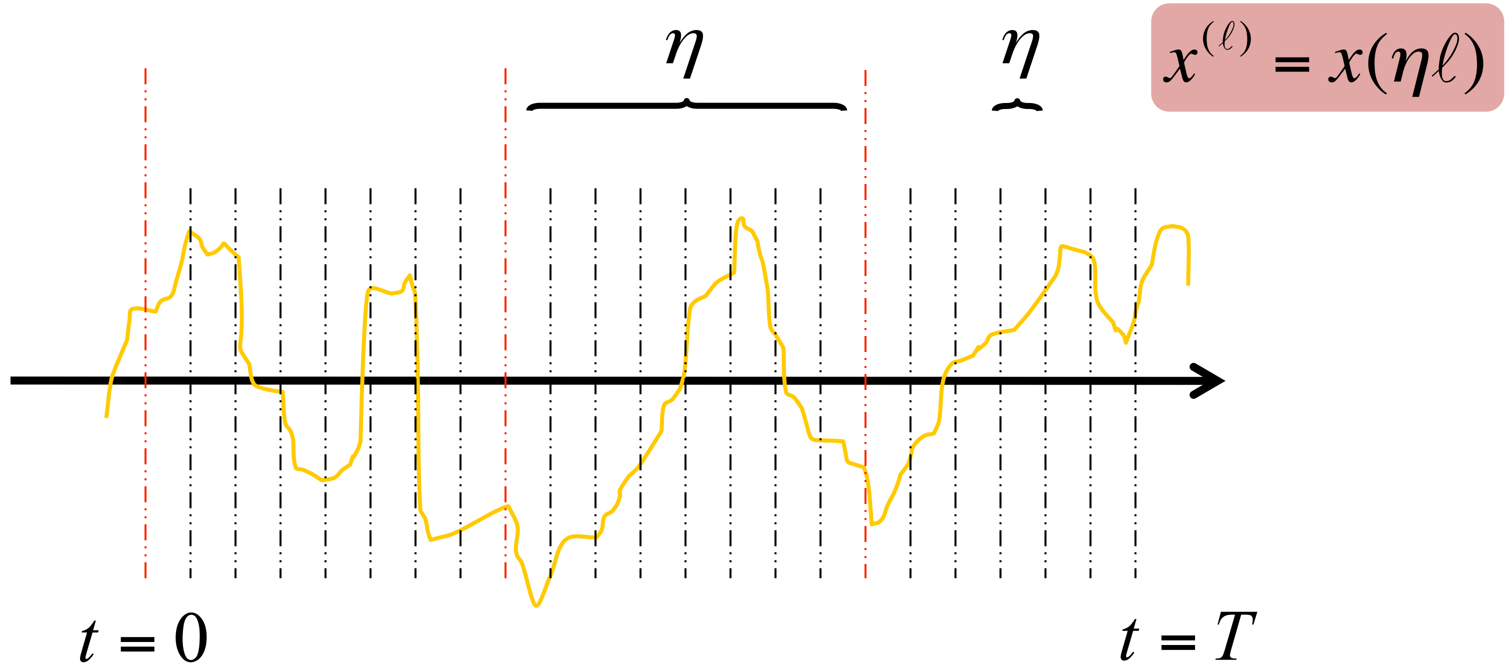
Given Y, X satisfying $Y = X\theta_i + \varepsilon$ estimate

$$\partial \mathrm{i} = \mathrm{supp}(\theta_i)$$

Why is our work different?

- Rows of Y and X obtained from $\{x(t)\}$ are not i.i.d.

Underlying challenge:
Spaced data or correlation?



	# samples	correlation
η decreases	$\uparrow$	$\uparrow$
η increases	$\downarrow$	$\downarrow$

Prior work: SDEs

Start with $$dx_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)dt + db_i(t)$$

Compute the likelihood function $$L(\Theta; \{x(t)\}_{t=0}^{T})$$

Estimate θ from $$\operatorname*{argmax}_{\Theta \in \Re^{p\times p}} L(\Theta; \{x(t)\}_{t=0}^{T})$$

ML methods for SDEs

- Basawa & Rao 80
- Kutoyants 04

Prior work: SDEs

Start with

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

Compute approx. likelihood function

$$L(\Theta; \{x(\eta\ell)\}_{\ell=1}^{n})$$

Estimate θ from

$$\underset{\Theta \in \Re^{p \times p}}{\operatorname{argmax}} L(\Theta; \{x(\eta\ell)\}_{\ell=1}^{n})$$

Approximate ML methods for **sampled SDEs**

- Dacunha-Castelle & Florens 86
- Pederson 95
- AitSahalia 02

Prior work: SDEs

Start with $$\mathrm{d}x_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_i(t)$$

Compute the likelihood function $$L(\Theta;\{x(t)\}_{t=0}^{T})$$

Estimate θ from $$\operatorname*{argmax}_{\Theta\in\Re^{p\times p}} L(\Theta;\{x(t)\}_{t=0}^{T})$$

Why is this work different?

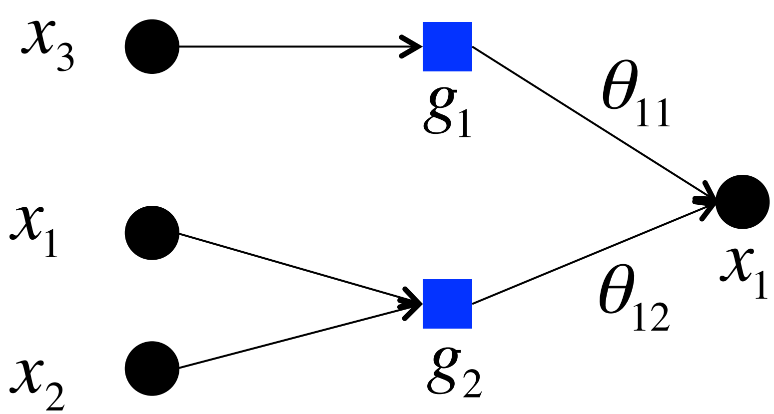
- Focus is on low dimensional setting
- ML is not a selection algorithm ($\theta \neq G$)
- Guarantees only hold asymptotically in T

The algorithm

Consider the non-linear SDE

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p'} \theta_{ij} g_j(x(t))\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

Example



$$\mathrm{d}x_1 = \big(\theta_{11} g_1(x_3) + \theta_{12} g_2(x_1, x_2)\big)\mathrm{d}t + \mathrm{d}b_1$$

The algorithm

The SDE is linear in θ : use a regression

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p'} \theta_{ij} g_j(x(t))\mathrm{d}t + \mathrm{d}b_\mathrm{i}(t)$$

The algorithm

The SDE is linear in θ : use a regression

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p'} \theta_{ij} g_j(x(t))\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

$$\Downarrow$$

Naïve derivation

$$\hat{\theta}_i = \underset{\Theta \in \mathbf{R}^p}{\operatorname{argmin}} \int_0^T \left(\mathrm{d}x_i(t) - \sum_{j=1}^{p'} \Theta_j g_j(x(t))\mathrm{d}t \right)^2 + \lambda \|\Theta\|_1$$

The algorithm: RLS(λ)

The SDE is linear in θ : use a regression

$$\mathrm{d}x_i(t) = \sum_{j=1}^{p'} \theta_{ij} g_j(x(t))\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$$

$$\hat{\theta}_i = \underset{\Theta \in \mathbf{R}^p}{\operatorname{argmin}} \int_0^T \left(\mathrm{d}x_i(t) - \sum_{j=1}^{p'} \Theta_j g_j(x(t))\mathrm{d}t \right)^2 + \lambda \|\Theta\|_1$$

Doing the math right $(\mathrm{d}b)^2 \to \mathrm{d}t$

$$\underset{\Theta \in \mathbf{R}^p}{\operatorname{argmin}} \frac{1}{2T} \int_0^T \left(\sum_{j=1}^{p'} \Theta_j g_j(x) \right)^2 \mathrm{d}t - \frac{1}{T} \int_0^T \sum_{j=1}^{p'} \Theta_j g_j(x) \mathrm{d}x_i + \lambda \|\Theta\|_1$$

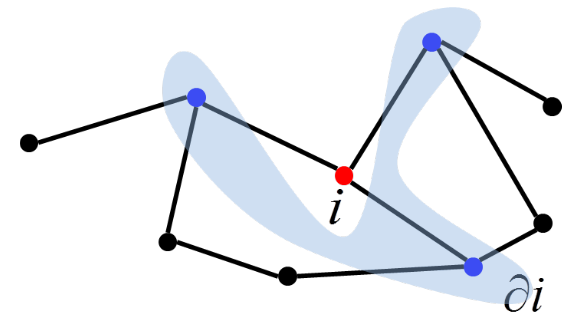
The algorithm: RLS(λ)

$$\hat{\theta}_i = \underset{\Theta \in \mathbf{R}^p}{\operatorname{argmin}} \underbrace{\frac{1}{2T}\int_0^T \left(\sum_{j=1}^{p'} \Theta_j g_j(x)\right)^2 \mathrm{d}t - \frac{1}{T}\int_0^T \sum_{j=1}^{p'} \Theta_j g_j(x) \mathrm{d}x_i}_{\mathcal{L}_i(\Theta; \{x(t)\}_{t=0}^{T})} + \lambda \|\Theta\|_1$$

What is the sample-complexity?

$$T_{\mathrm{RLS},G} = \inf\left\{T : \mathbf{P}(\mathrm{supp}(\hat{\theta}) = \mathrm{supp}(\theta)) > 1 - \delta\right\}$$

Characterization of T_{RLS}



Given $G = (V, E)$ define

$$\mathrm{d}x_i = -m x_i \mathrm{d}t + \sum_{j \in \partial i} (x_j - x_i)\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}$$

Let G be sparse of maximum degree Δ

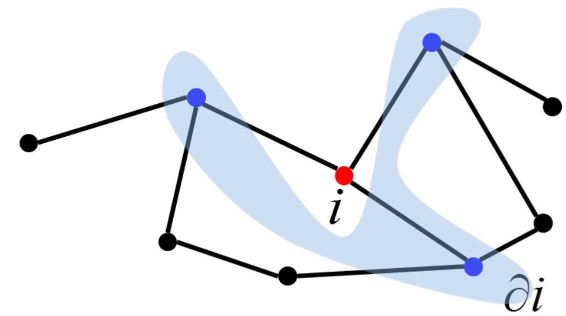
Theorem

If $p, m >> \Delta$ then

$$C(\Delta) m \log p < T_{\mathrm{RLS},G} < C'(\Delta) m^2 \log p$$

-Bento, Ibrahimi, Montanari, Proceedings of Neural Information Processing Systems (NIPS), 2010
-Bento, Ibrahimi, Montanari, IEEE International Symposium on Information Theory (ISIT), 2011

Characterization of T_{RLS}



Given $G = (V, E)$ define

$$\mathrm{d}x_i = -m x_i \mathrm{d}t + \sum_{j \in \partial i} (x_j - x_i)\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}$$

Now, G can be dense $(\Delta \sim \mathrm{p})$

Theorem

If $p, m >> 1$ then

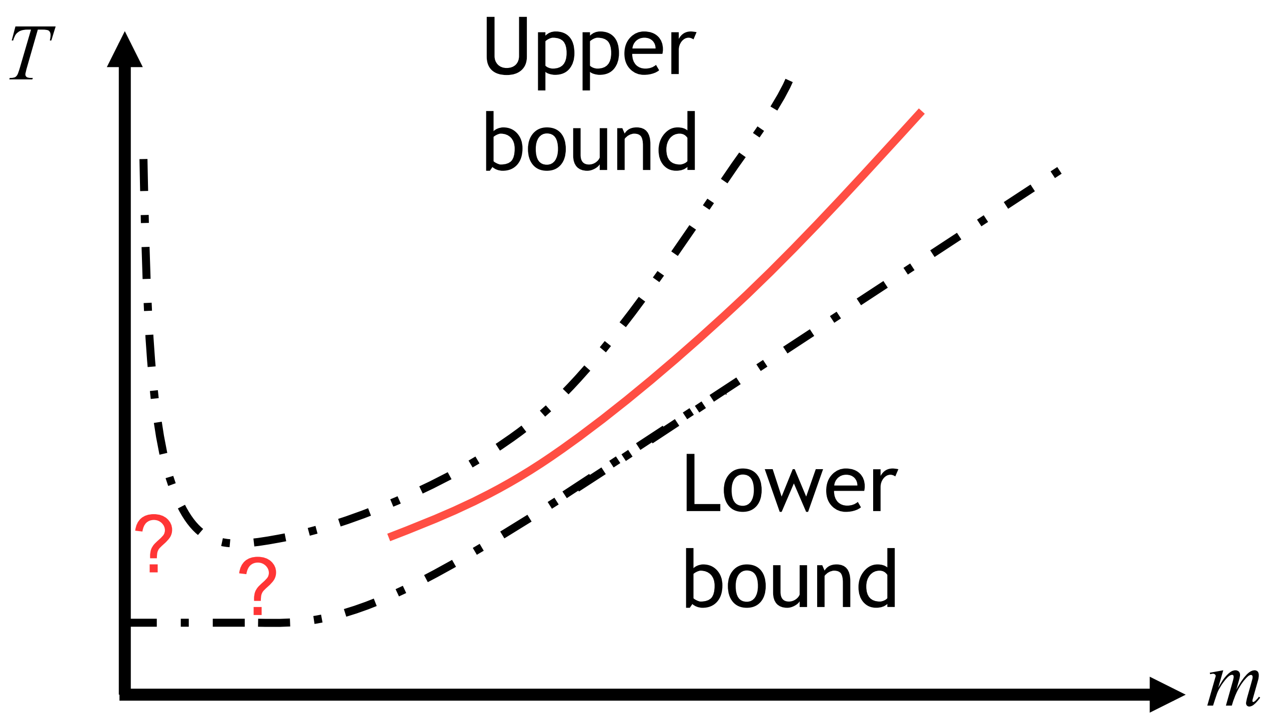
$$Cmp < T_{\mathrm{RLS},G} < C' mp$$

-Bento, Ibrahimi, Montanari, IEEE Transactions in information Theory, 2013
-Bento, Ibrahimi, Montanari, IEEE International Symposium on Information Theory (ISIT), 2011

General behaviour

- $T \sim \log p$: Large sparse systems can be learned from few samples

- $T \sim m$: Fast systems are harder to learn



General characterization of T_{RLS}

General theorem for $\mathrm{d}x_i(t) = \sum_{j=1}^{p} \theta_{ij} x_j(t)\mathrm{d}t + \mathrm{d}b_{\mathrm{i}}(t)$ requires some assumptions

In particular, it requires assumptions similar to the ones in Zhao & Yu 06 for sparse linear regression

- Entries smallest value is lower bounded
- Strong stability assumption
- Restricted convexity assumption
- Irrepresentable condition

Important ideas behind proof

1) Work with discretized SDE and take limits
2) Proof structure similar to Zhao & Yu 06 requires two new concentration bounds

$$\mathbb{P}\left(\left|\hat{G}_{ij}\right| > \epsilon\right) \leq C_1 e^{-C_2 n \eta \epsilon^2}$$

$$\hat{G}_{ij} = \frac{\mathrm{d}\mathcal{L}(\theta_i; \{x(k\eta)\}_{k=0}^{n})}{\mathrm{d}\theta_{ij}}$$

$$\mathbb{P}\left(\left|\hat{Q}_{ijk} - Q^0\right| > \epsilon\right) \leq C_1' e^{-C_2' n \eta \epsilon^2}$$

$$\hat{Q}_{ijk} = \frac{\mathrm{d}^2\mathcal{L}(\theta_i; \{x(k\eta)\}_{k=0}^{n})}{\mathrm{d}\theta_{ij}\mathrm{d}\theta_{ik}}$$

unlike in Zhao & Yu 06, G and Q are not of the form $C^\dagger Z$ for a random vector Z with i.i.d. entries but are of the form $Z^\dagger R Z$ for some matrix R.

Follow up work

- Autoregressive processes

$$x(t+1) = \sum_{r=0}^{k-1} \Theta(r) x(t-r) + u(t)$$

Bolstad et al. 2011 (group Lasso + tailored incoherence condition)

- Hidden variables (observe x)

$$x(t+1) = \Theta_0 x(t) + \Theta_1 h(t) + u_1(t)$$
$$h(t+1) = \Theta_2 x(t) + \Theta_3 h(t) + u_2(t)$$

A. Jalali & S. Sanghavi 2012 (only want Θ_0 + L1 + nuclear norm penalties + two incoherence cond.)

Follow up work

- Non-parametric

Ruttor et al. 2013 (estimate drift + sampled paths + Gaussian Process + approximate EM)

$$\mathrm{d}x(t) = F(x(t))\mathrm{d}t + D^{1/2}\mathrm{d}b(t)$$

Jung et al. 2014 (learn dependencies in GP + Blackman-Tukey estimator + group Lasso OR graphical Lasso + thresholding); (IT lower-bound)

$$(k,l) \notin E \text{ iff } [\mathbf{S}^{-1}(\omega)]_{k,l} = 0 \;\forall\, \omega \in [0,1)$$

$$\mathbf{S}(\omega) = \sum_{i=-\infty}^{\infty} R(t) e^{-j2\pi t\omega}; R(t) = \mathbb{E}(x(t)x^{\dagger}(0))$$

Extension:
Non-linear SDEs

1. Algorithm analysis only applies for linear SDEs

2. Some non-linear SDEs can be described as a linear combination of a set of basis functions

Extension:
Non-linear SDEs

1. Algorithm analysis only applies for linear SDEs

2. Some non-linear SDEs can be described as a linear combination of a set of basis functions

Open problem

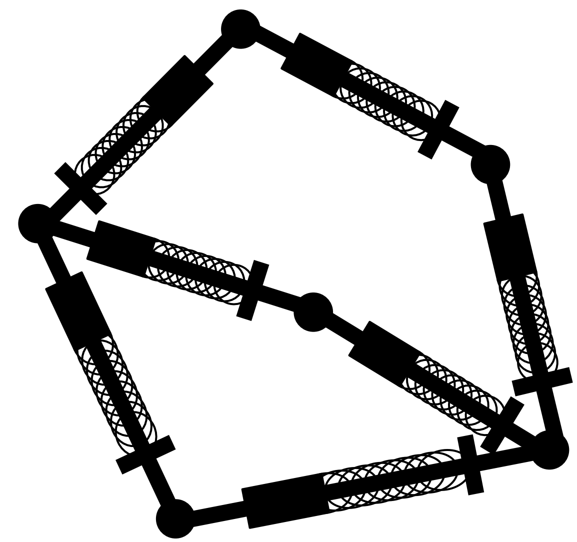
Does the same analysis hold for non-linear SDEs?

Numerical experiment
Learning mass-spring networks

$$\mathrm{d}v(t) = -\gamma v(t)\mathrm{d}t - \nabla U(q(t))\mathrm{d}t + \sigma \mathrm{d}b(t)$$

$$dq(t) = v(t)\mathrm{d}t$$

$$U(q) \equiv \frac{1}{2}\sum_{(i,j)} C_{ij}^{0}\left(\left\|q^{(i)} - q^{(j)}\right\| - D_{ij}^{0}\right)^{2}$$

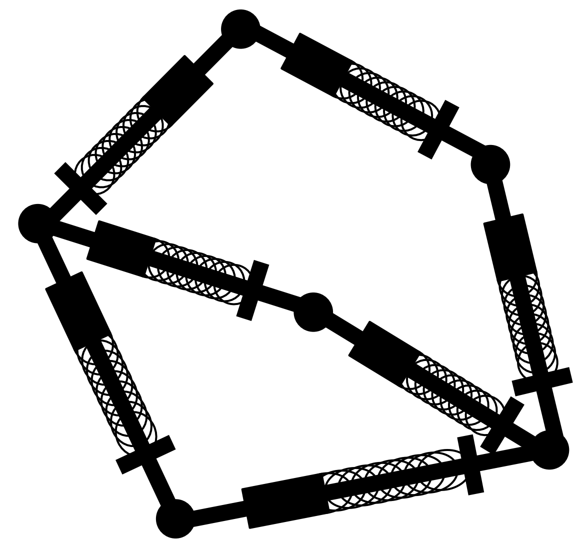


Numerical experiment
Learning mass-spring networks

$$\mathrm{d}v(t) = -\gamma v(t)\mathrm{d}t - \nabla U(q(t))\mathrm{d}t + \sigma \mathrm{d}b(t)$$

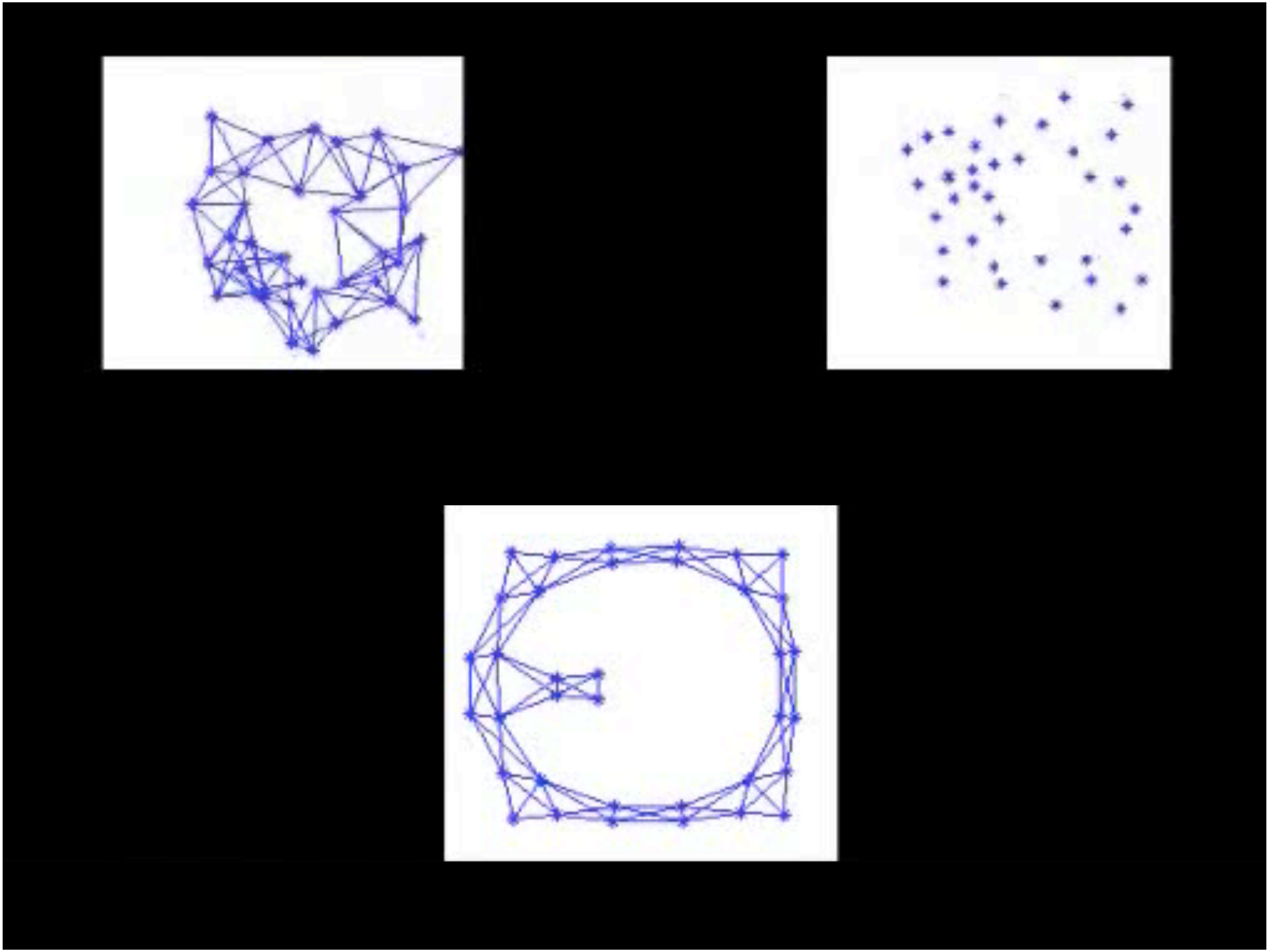
$$dq(t) = v(t)\mathrm{d}t$$

$$U(q) \equiv \frac{1}{2}\sum_{(i,j)} C_{ij}^{0}\left(\left\|q^{(i)} - q^{(j)}\right\| - D_{ij}^{0}\right)^{2}$$



Drift = linear combination of basis functions

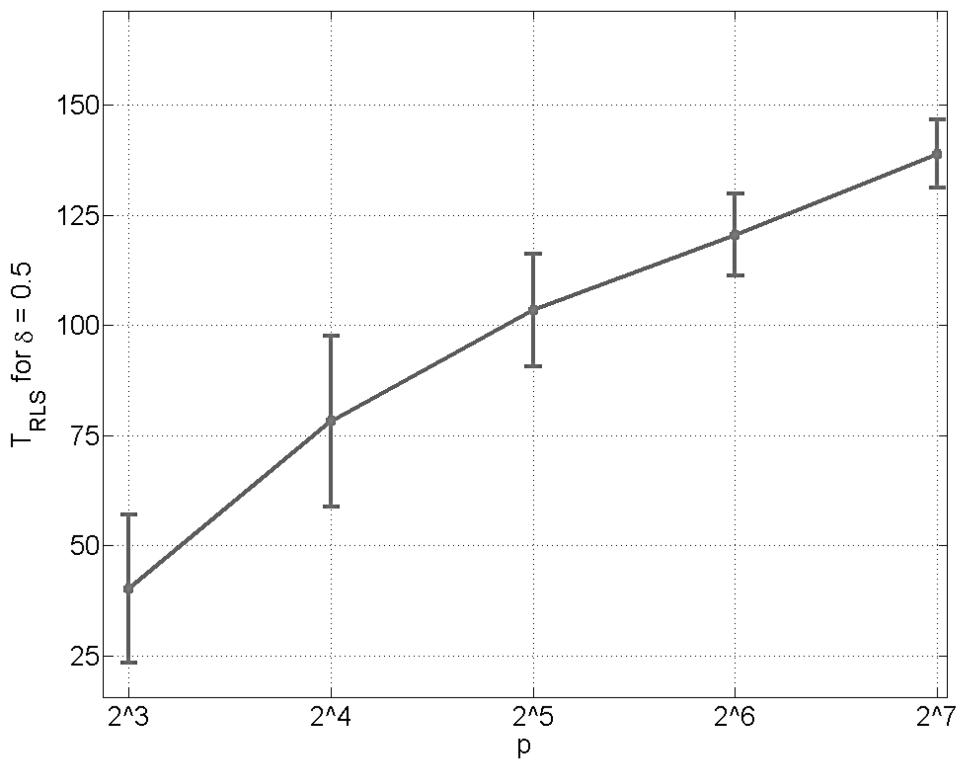
$$\left\{v_i(t)\right\}_{i\in[p]}, \left\{q_i(t) - q_j(t)\right\}_{i,j\in[p]}, \left\{\frac{q_i(t) - q_j(t)}{\left\|q_i(t) - q_j(t)\right\|}\right\}_{i,j\in[p]}$$



Numerical experiment
Learning mass-spring networks

Sample-complexity for learning regular graphs of different sizes p

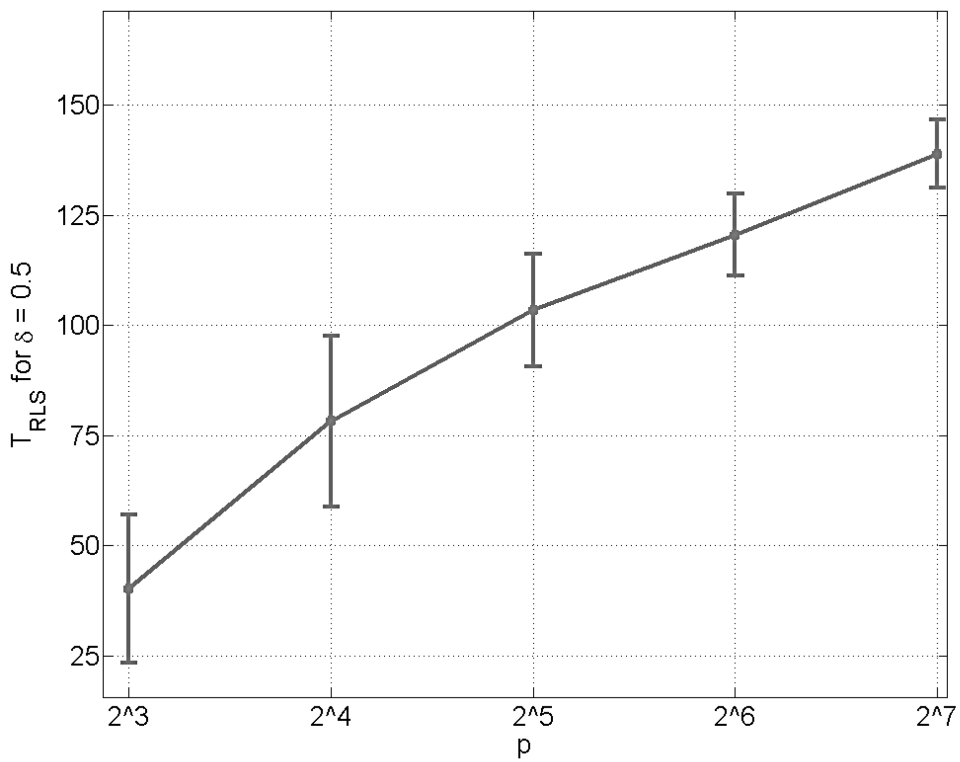
$$T_{RLS} \sim \log p$$



Numerical experiment
Learning mass-spring networks

Sample-complexity for learning regular graphs of different sizes p

$$T_{RLS} \sim \log p \qquad (T_{RLS} > C \log p)$$



By the general lower bound

Summary

1. For fast (large m) linear SDEs, RLS has optimal sample complexity ($\log p$ or p)

2. Empirical results suggest RLS has optimal scaling of T_{RLS} with p for sparse non-linear SDEs

3. Upper bound on T_{RLS} suggests that performance of RLS degrades for slow (small m) linear SDEs

4. For fast linear SDEs $T_{\mathrm{RLS}} \sim$ between m and m^2

Thank you