Where identification meets learning: Data-driven approaches to estimating properties of dynamical systems

Cristian R. Rojas

Division of Decision and Control Systems
KTH Royal Institute of Technology
Stockholm, Sweden

Grand séminaire MACS – Edition 2025
Paris, Oct 15-16

Outline

Introduction

Automatic control has been called the “hidden technology”…

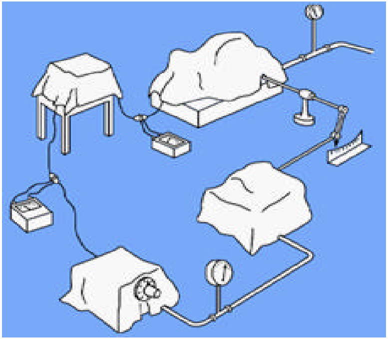
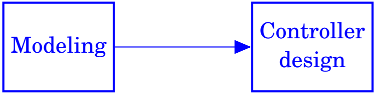


Image from slides of K.J. Åström, 2004

Modelling: hidden technology *behind* the hidden technology of control

Introduction (cont.)

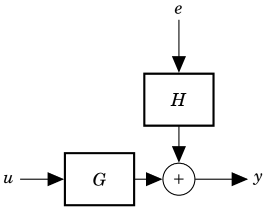
“Classical” control design workflow:



- $\sim 75\%$ of total cost (in money and time) of designing industrial controllers is due to modeling(*)
- However, for some applications one does not need a full model... $\Rightarrow$ (partially) skip the modeling step!
- In this talk, we will provide a few examples, where learning techniques come in handy

(*) Y. Zhu, “System Identification for Process Control: Recent Experiences and a Personal Outlook”. Plenary talk at SYSID, 2006.

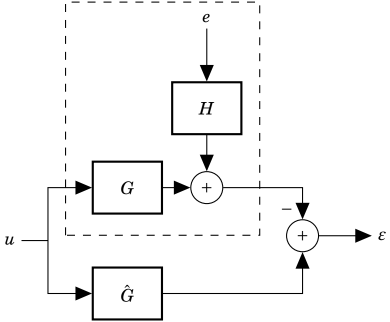
Motivation



- To design a controller for a linear process G, robust control approaches typically need a model $\hat{G}$ of G, as well as a bound on its error $\hat{G} - G$ (*e.g.*, its $\mathscr{H}_\infty$-norm)
- To estimate $\hat{G}$, one may impose several assumptions, *e.g.*,
  - model structure (ARX, ARMAX, BJ, ...)
  - number of poles and zeros
  - bounds on coefficients of the polynomials, ...
- Modeling error $\hat{G} - G$ consists of bias + variance errors, thus it is desirable to make *as few assumptions as possible* to bound it
- This motivates use of *data driven approaches* to estimate $\|\hat{G} - G\|_\infty$

Motivation (cont.)

Idea: Construct an artificial signal ε, the output of error $\hat{G} - G$



By working with batches of input-output samples (u_t, ε_t), where u_t is designed based on previous batches, we can construct an estimate of $\|\hat{G} - G\|_\infty$

Let us discuss some approaches to estimate the $\mathcal{H}_\infty$ norm of a system in this way...

Power Iterations

with H. Hjalmarsson (KTH), B. Wahlberg (KTH)
and T. Oomen (TU/e)

Introduction

Assume that at iteration t, the system is reset. If there is no noise, the output can be written as

$$\boldsymbol{y}^t = \boldsymbol{G}\boldsymbol{u}^t$$

where $\boldsymbol{u}^t = [u_1^t \ \cdots \ u_N^t]^T$, $\boldsymbol{y}^t = [y_1^t \ \cdots \ y_N^t]^T$, and

$$\boldsymbol{G} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_1 & g_0 \end{bmatrix} \qquad \text{(lower triangular Toeplitz matrix)}$$

If N is sufficiently large,

$$\|G\|_\infty^2 = \sup_{u \neq 0} \frac{\|G(q)u\|_{\ell_2}^2}{\|u\|_{\ell_2}^2} \approx \sup_{\boldsymbol{u}^t \neq \boldsymbol{0}} \frac{(\boldsymbol{u}^t)^T \boldsymbol{G}^T \boldsymbol{G} \boldsymbol{u}^t}{(\boldsymbol{u}^t)^T \boldsymbol{u}^t}.$$

Thus, $\|G\|_\infty^2$ is approx. largest eigenvalue of $\boldsymbol{G}^T\boldsymbol{G}$, so it can be estimated using the *power iterations method*!

Power iterations method

To estimate largest eigenvalue of $\boldsymbol{A} \in \mathbb{R}^{n \times n}$:

1. Choose $\boldsymbol{x}_0 \in \mathbb{R}^n$ at random, and normalize it so $\|\boldsymbol{x}_0\|_2 = 1$
2. Set $i \leftarrow 1$
3. Compute $\boldsymbol{x}_i \leftarrow \dfrac{1}{\|\boldsymbol{A}\boldsymbol{x}_{i-1}\|_2}\boldsymbol{A}\boldsymbol{x}_{i-1}$
4. Set $i \leftarrow i+1$ and go to Step 3

Largest eigenvalue can be estimated as $\|\boldsymbol{A}\boldsymbol{x}_{i-1}\|_2$, or as

$$\boldsymbol{x}_{i-1}^T \boldsymbol{A}\boldsymbol{x}_{i-1}$$

To apply this method to the estimation of $\|G\|_\infty$, we need to compute the action of $\boldsymbol{G}^T$ on a vector…

Power iterations method (cont.)

Due to Toeplitz structure of $\boldsymbol{G}$,

$$\boldsymbol{G}^T = \boldsymbol{\tau}\boldsymbol{G}\boldsymbol{\tau}$$

where $\boldsymbol{\tau}$ is a *time reversal operator*:

$$\boldsymbol{\tau} = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & 1 & 0 \\ \vdots & \iddots & & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}$$

so $\boldsymbol{\tau}[x_1 \ \cdots \ x_N]^T = [x_N \ \cdots \ x_1]^T$

Power iterations method (cont.)

Since $\boldsymbol{G}^T\boldsymbol{G} = \boldsymbol{\tau}\boldsymbol{G}\boldsymbol{\tau}\boldsymbol{G}$, we obtain

Power iterations method for $\mathscr{H}_\infty$-norm estimation:

1. Choose $\boldsymbol{u}^0 \in \mathbb{R}^n$ at random, and normalize it so $\|\boldsymbol{u}^0\|_2 = \sqrt{N}$
2. Set $i \leftarrow 1$
3. Apply $\boldsymbol{u}^{i-1}$ to the system, and record its output $\boldsymbol{y}^i \in \mathbb{R}^N$
4. Time reverse $\boldsymbol{y}^i$ and normalize it so its norm equals $\sqrt{N}$:

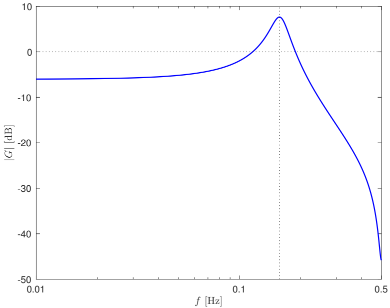
$$\boldsymbol{u}^i \leftarrow \frac{\sqrt{N}}{\|\boldsymbol{\tau}\boldsymbol{y}^i\|_2}\boldsymbol{\tau}\boldsymbol{y}^i$$

5. Set $i \leftarrow i+1$ and go to Step 3

An estimate of $\|G\|_\infty$ can be obtained as

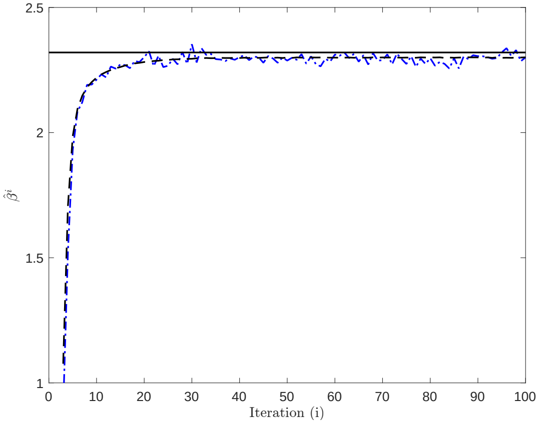
$$\hat{\boldsymbol{\beta}}^i = \frac{(\boldsymbol{u}^{i-1})^T\boldsymbol{\tau}\boldsymbol{y}^i}{\sqrt{N}}$$

Example



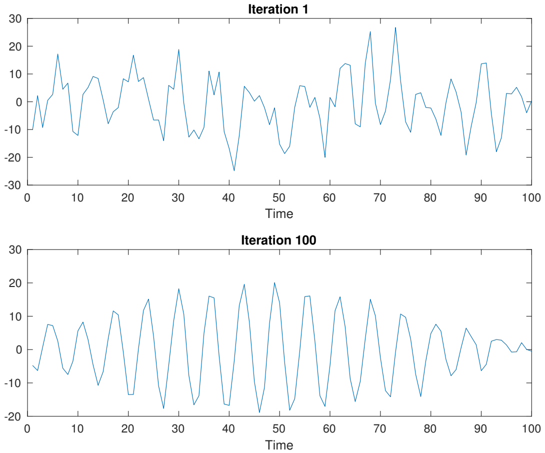
- $\mathscr{H}_\infty$-norm: 2.397, at peak frequency 0.159 Hz
- White noise of variance $\sigma_e^2 = 9$ added to the output
- Batches of $N = 100$ samples

Example (cont.)



Estimate of the $\mathscr{H}_\infty$-norm of G (blue, dash-dotted), and average of 1000 Monte Carlo simulations (black, dashed). Also shown is the actual $\mathscr{H}_\infty$-norm of G (black, solid).

Example (cont.)



Input signal generated by power iterations after iterations 1 (top) and 100 (bottom)

Summary of power iterations

- It converges very quickly (~ 30 iterations, in the example)
- The estimate is *negatively biased*, due to the presence of noise: this bias is roughly proportional to σ_e^2
- Input signal $\boldsymbol{u}^t$ converges to an eigenvector of $\boldsymbol{G}$, which resembles a "modulated" sinusoid whose frequency is approx. the peak frequency

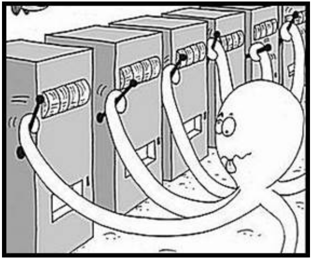
Details:

- H. Hjalmarsson, "From experiment design to closed-loop control". *Automatica*, 2005.
- C.R.R., T. Oomen, H. Hjalmarsson, B. Wahlberg, "Analyzing iterations in identification with application to nonparametric $\mathscr{H}_\infty$-norm estimation". *Automatica*, 2012.
- T. Oomen, R. van der Maas, C.R.R., H. Hjalmarsson, "Iterative data-driven $\mathscr{H}_\infty$ norm estimation of multivariable systems with application to robust active vibration isolation. *IEEE TCST*, 2014.

Bandit Algorithms

with M. I. Müller (KTH), P. Valenzuela (KTH), A. Proutiere (KTH),
S. Formentin (P. Milano) and T. Oomen (TU/e)

Introduction to multi-armed bandits



- Popular machine learning framework for adaptive control (but where the "plant" is static)
- Name coined in 1952 by H. Robbins, in the context of Sequential Design of Experiments
- Exploration vs exploitation dilemma
- First asymptotically optimal solution proposed in 1985 by T.L. Lai and H. Robbins

Basic setup (for stochastic MABs):

- There are K *arms* (slot machines) to choose from
- One can play one arm in each round
- Each arm j gives a reward $X_{j,t} \in \{0,1\}$, with mean μ_j (t: round) (**Obs** only the reward of the selected arm j is revealed)
- Problem: which machine should one play in each round?

Performance of a strategy measured in terms of *expected cumulative regret*:

$$R(T) = \sum_{i=1}^{T} (\mu^* - \mathrm{E}\{\mu_{a(i)}\})$$

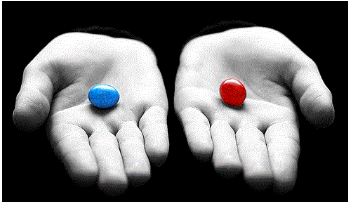
where: T: number of rounds
$\mu^* = \max_i \mu_i$: best reward
$a(i)$: arm chosen at round i

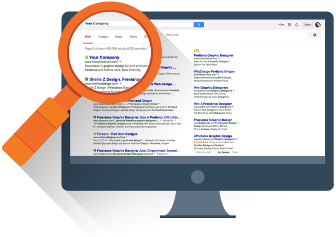
Introduction to multi-armed bandits (cont.)

Applications:

- Clinical trials
- Ad placement on webpages
- Recommender systems
- Computer game-playing
- …

2			4	5	2.94*
5		4			1
		5		2	2.48*
	1		5		4
		4			2
4	5		1		1.12*

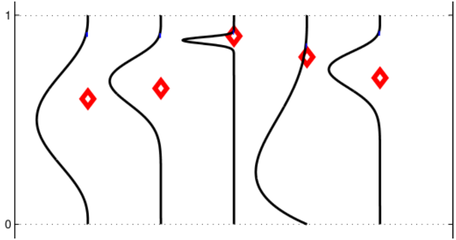




Optimal algorithms

Thompson sampling: (Thompson, 1933)

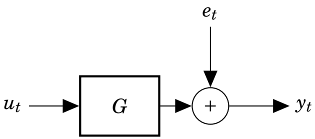
- First MAB algorithm, conceived for adaptive clinical trials
- Bayesian origin: Assume a uniform prior on μ_j for every j, and update the posterior p_{μ_j} based on samples up to round t
- At round t, sample $\hat{\mu}_j$ from posterior p_{μ_j}, and pick arm for which $\hat{\mu}_j$ is largest



- Empirically shown to have better finite sample mean performance than UCB, but its variance can be higher
- Kaufmann, Korda & Munos (ALT, 2012) showed that Thompson Sampling is asymptotically optimal

Application to $\mathscr{H}_\infty$-norm estimation

Setup

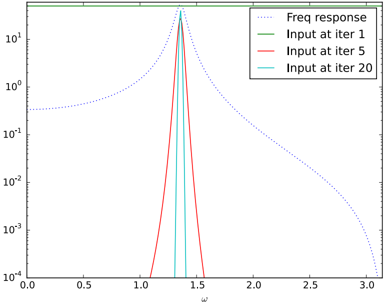


- Work with data batches, of length N
- At each iteration i, an input batch $\boldsymbol{u}^i = (u_1, \ldots, u_N)$ is designed and applied to the system
- The output of the system, $\boldsymbol{y}^i = (y_1, \ldots, y_N)$, is collected
- **Goal** Determine the $\mathscr{H}_\infty$ norm of the system, as accurately as possible

Application to $\mathscr{H}_\infty$-norm estimation (cont.)

Main Idea:

Design $\boldsymbol{u}^i$ in frequency domain, considering each freq. $2\pi k/N$ as an arm!

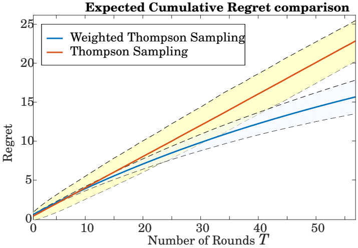


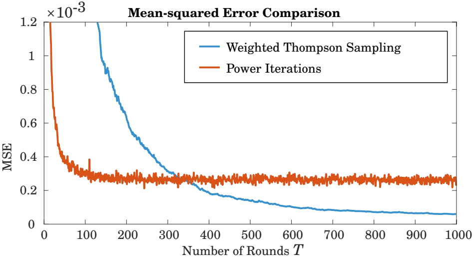
This is a standard MAB problem, except that:

- More than one arm can be pulled at once (in fact, we can choose a *distribution* over the arms!)
- The outcomes are complex-valued Gaussian distributed (variance inversely proportional to applied power)

Application to $\mathscr{H}_\infty$-norm estimation (cont.)

- Derived a lower bound for the problem, which shows that choosing only one freq. is not more restrictive (asymptotically in i) than a continuous spectrum for $\boldsymbol{u}^i$, *when variance of e is known*
- Proposed a *weighted Thompson sampling* algorithm with better regret than standard TS
- Still... power iterations has better initial transient than MAB algorithms!





Application to $\mathscr{H}_\infty$-norm estimation (cont.)

	Known variances $(\sigma^2,\ldots,\sigma^2)$	Unknown variances $(\sigma_1^2,\ldots,\sigma_N^2)$
Single freq.	$\sum_{k\neq k^\star} \frac{\sigma^2}{\Delta_k}$	$\sum_{k\neq k^\star} \frac{\Delta_k}{\ln\left(1+\frac{\Delta_k^2}{\sigma_k^2}\right)}$
Power spreading	$\sum_{k\neq k^\star} \frac{\sigma^2}{\Delta_k}$	$\sum_{k\neq k^\star} \frac{\sigma_k^2}{\Delta_k}$

Lower bounds on regret: $\liminf_{T\to\infty} \mathrm{E}\{R^\pi(T)\}/\ln T$, for known/unknown variance noise, and policies restricted to a single sinusoid ("single freq.") applied in each iteration, or to a multisine ("power spreading").

- When variances are different and unknown, lower bound at least as high as when they are equal and known,
- however, for power spreading strategies, lower bounds are equal!

Summary of MABs

- MABs are a useful approach to adaptive control
- Standard theory applicable to some problems of iterative identification and control
- A relevant example: $\mathscr{H}_\infty$-norm estimation
- Control applications require non-trivial extensions to basic MAB framework:

Interesting research directions!

Details:

- M.I. Müller, P.E. Valenzuela, A. Proutiere, C.R.R. "A stochastic multi-armed bandit approach to nonparametric $\mathscr{H}_\infty$-norm estimation". *CDC*, 2017.
- M.I. Müller, C.R.R. "Gain estimation of dynamical linear systems using Thompson Sampling". *AISTATS*, 2019.
- M.I. Müller, C.R.R. "Iterative $\mathscr{H}_\infty$-norm estimation using cyclic-prefixed signals'. *CDC*, 2020.
- M.I. Müller, C.R.R. "Asymptotically optimal bandits under weighted information". arXiv: 2105.14114.
- G. Rallo, S. Formentin, C.R.R., T. Oomen, S.M. Savaresi, "Data-driven $\mathscr{H}_\infty$-norm estimation via expert advice". *CDC*, 2017.

Extensions

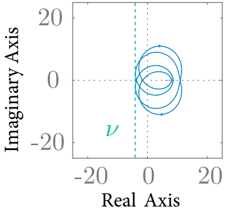
with M. I. Müller (KTH), M. A. Guerrero (KTH),
A. Koch (U. Stuttgart) and F. Allgöwer (U. Stuttgart)

Passivity index

- From classical nonlinear control theory, we know that feedback interconnection of a passive system and a strictly passive one is stable
- The *passivity index*

$$\nu = \min_{\omega\in[0,\pi]} \mathrm{Re}\, G(e^{i\omega})$$

measures distance to passivity



- The passivity index can be re-written as

$$\nu = \left\| b - \frac{1}{2}(G + G^{\sim}) \right\|_{\infty} - b$$

for $b > 0$ sufficiently large, to which power iterations can be applied

Details:

M. I. Müller, A. Koch, F. Allgöwer and C.R.R. "Data-driven input-passivity Estimation using power iterations", *SYSID*, 2021.

Vinnicombe ν-gap metric

- The *ν-gap metric* between two systems $P_1(q)$ and $P_2(q)$, is widely used in robust control, as it induces the weakest topology for which closed-loop stability is a robust property
- For SISO systems, under a winding number condition,

$$\delta_\nu(P_1,P_2) = \sup_{\omega\in(-\pi,\pi]} \frac{|P_1(e^{i\omega}) - P_2(e^{i\omega})|}{\sqrt{1+|P_1(e^{i\omega})|^2}\sqrt{1+|P_2(e^{i\omega})|^2}}$$

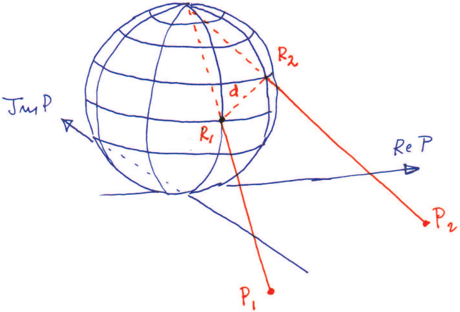


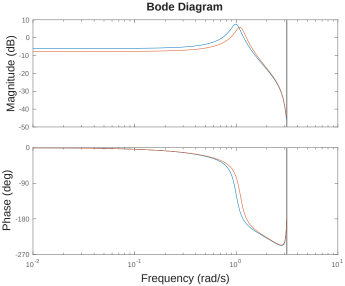
Image from slides of K.J. Åström, 2019

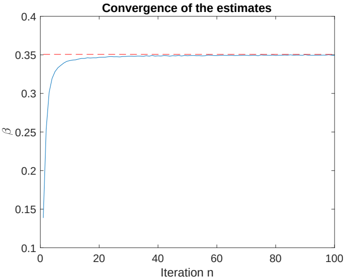
Vinnicombe ν-gap metric (cont.)

- If the outputs to P_1 and P_2, with input u, are y_1 and y_2, respectiv., then

$$\delta_\nu(P_1,P_2) = \sup_{\omega\in(-\pi,\pi]} \frac{|Y_1(e^{i\omega}) - Y_2(e^{i\omega})|\,|U(e^{i\omega})|}{\sqrt{|U(e^{i\omega})|^2 + |Y_1(e^{i\omega})|^2}\sqrt{|U(e^{i\omega})|^2 + |Y_2(e^{i\omega})|^2}}$$

- One can apply power iterations to this!





(work in progress...)

Connections to Cyber-Physical Security

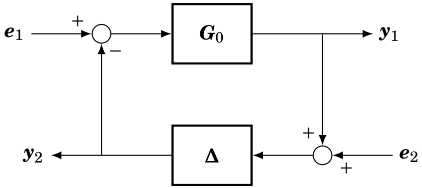
with M. A. Guerrero (KTH) and B. Lakshminarayanan (KTH)

Introduction

- *Identification of vulnerabilities*: Smallest perturbation to a critical infrastructure leading to destabilization or mis-behavior

- Given a stable interconnected system, a question is: what is the "smallest" perturbation that will render it unstable?

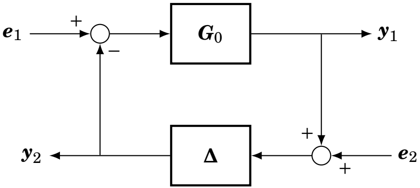
- The question is related to that of *robust stability*:



If $\boldsymbol{G}_0$ is a fixed stable MIMO system, and $\boldsymbol{\Delta}$ denotes a stable and "uncertain", but structured block, what is the smallest $\|\boldsymbol{\Delta}\|_\infty$ for which the interconnection $(\boldsymbol{G}_0, \boldsymbol{\Delta})$ is unstable?

- Classical solution: *structured singular value* $\mu_\Delta(\boldsymbol{G}_0)$

Infinity norm and structured singular values



If no assumptions, beyond stability, are made on Δ, robust stability is solved by the *small gain theorem*:

$$\inf_{\Delta \in \mathcal{H}_\infty :\ (G_0, \Delta) \text{ unstable}} \|\Delta\|_\infty = \frac{1}{\|G_0\|_\infty}$$

Infinity norm and structured singular values (cont.)

What if $\boldsymbol{\Delta}$ has a block diagonal structure?

$$\boldsymbol{\Delta} \in \Delta = \{\mathrm{diag}(\delta_1 \boldsymbol{I}_{r_1}, \dots, \delta_s \boldsymbol{I}_{r_s}, \boldsymbol{\Delta}_1, \dots, \boldsymbol{\Delta}_f)\}$$

$(\delta_1, \dots, \delta_s \in \mathbb{C};\ \boldsymbol{\Delta}_1 \in \mathbb{C}^{m_1 \times m_1}, \dots, \boldsymbol{\Delta}_f \in \mathbb{C}^{m_f \times m_f})$

Robust stability is guaranteed iff $\|\boldsymbol{\Delta}\|_\infty \mu_\Delta(\boldsymbol{G}_0) < 1$, where

$$\mu_\Delta(\boldsymbol{G}_0) := \frac{1}{\min\{\|\boldsymbol{\Delta}\|_\infty : \boldsymbol{\Delta} \in \Delta,\ \det[\boldsymbol{I} + \boldsymbol{G}_0(z)\boldsymbol{\Delta}(z)] = 0 \text{ for some } z \in \partial\mathbb{D}\}}$$

Structured singular value of $\boldsymbol{G}_0$!

Reduction to static matrices

μ_Δ can be written as $\mu_\Delta(\boldsymbol{G}_0) = \sup_{\omega\in[-\pi,\pi)} \mu'_\Delta(\boldsymbol{G}_0(e^{i\omega}))$, where

$$\mu'_\Delta(\boldsymbol{M}) := \frac{1}{\min\{\bar{\sigma}(\boldsymbol{\Delta}):\ \boldsymbol{\Delta}\in\Delta,\ \det[\boldsymbol{I}+\boldsymbol{M}\boldsymbol{\Delta}]=0\}}$$

Computing μ'_Δ is NP-hard, but the following bounds are known:

$$\max_{\boldsymbol{Q}\in\mathcal{Q}} \rho(\boldsymbol{QM}) = \max_{\boldsymbol{Q}\in B\Delta} \rho(\boldsymbol{QM}) = \mu'_\Delta(\boldsymbol{M}) \leqslant \inf_{\boldsymbol{D}\in\mathcal{D}} \bar{\sigma}(\boldsymbol{DMD}^{-1})$$

where

$$\begin{aligned}
\mathcal{Q} &:= \{\boldsymbol{\Delta}\in\Delta:\ \boldsymbol{\Delta}^H\boldsymbol{\Delta}=\boldsymbol{I}\} \\
\mathrm{B}\Delta &:= \{\boldsymbol{\Delta}\in\Delta:\ \bar{\sigma}(\boldsymbol{\Delta})\leqslant 1\} \\
\mathcal{D} &:= \{\mathrm{diag}(\boldsymbol{D}_1,\dots,\boldsymbol{D}_s,d_1\boldsymbol{I}_{m_1},\dots,d_1\boldsymbol{I}_{m_f}): \\
&\qquad\qquad \boldsymbol{D}_1\in\mathbb{S}^{r_1},\dots,\boldsymbol{D}_s\in\mathbb{S}^{r_s},\ d_1,\dots,d_f\in\mathbb{R}_+\}
\end{aligned}$$

Estimation of structured singular values

Packard, Fan and Doyle (1988) developed a power iterations method to approximate $\max_{\boldsymbol{Q}\in\mathcal{Q}}\rho(\boldsymbol{Q}\boldsymbol{M})$ (a lower bound of $\mu'_\Delta(\boldsymbol{M})$):

$$
\begin{aligned}
\tilde{\mu}(l+1) &\leftarrow \|\boldsymbol{M}\boldsymbol{b}(l)\| \\
\boldsymbol{a}(l+1) &\leftarrow \frac{1}{\tilde{\mu}(l+1)}\boldsymbol{M}\boldsymbol{b}(l) \\
\boldsymbol{z}_{r_j}(l+1) &\leftarrow \frac{\boldsymbol{w}_{r_j}^H(l)\boldsymbol{a}_{r_j}(l+1)}{|\boldsymbol{w}_{r_j}^H(l)\boldsymbol{a}_{r_j}(l+1)|}\boldsymbol{w}_{r_j}(l), \quad j=1,\ldots,s \\
\boldsymbol{z}_{m_k}(l+1) &\leftarrow \frac{\|\boldsymbol{w}_{m_k}(l)\|}{\|\boldsymbol{a}_{m_k}(l+1)\|}\boldsymbol{a}_{m_k}(l+1), \quad k=1,\ldots,f \\
\bar{\mu}(l+1) &\leftarrow \|\boldsymbol{M}^H\boldsymbol{z}(l+1)\| \\
\boldsymbol{w}(l+1) &\leftarrow \frac{1}{\bar{\mu}(l+1)}\boldsymbol{M}^H\boldsymbol{z}(l+1) \\
\boldsymbol{b}_{r_j}(l+1) &= \frac{\boldsymbol{a}_{r_j}^H(l+1)\boldsymbol{w}_{r_j}(l+1)}{|\boldsymbol{a}_{r_j}^H(l+1)\boldsymbol{w}_{r_j}(l+1)|}\boldsymbol{a}_{r_j}(l+1), \quad j=1,\ldots,s \\
\boldsymbol{b}_{m_k}(l+1) &= \frac{\|\boldsymbol{a}_{m_k}(l+1)\|}{\|\boldsymbol{w}_{m_k}(l+1)\|}\boldsymbol{w}_{m_k}(l+1), \quad k=1,\ldots,f.
\end{aligned}
$$

These are iterated over $l=0,1,\ldots$ ($\boldsymbol{b}(0)$,$\boldsymbol{w}(0)$ are unit vectors)

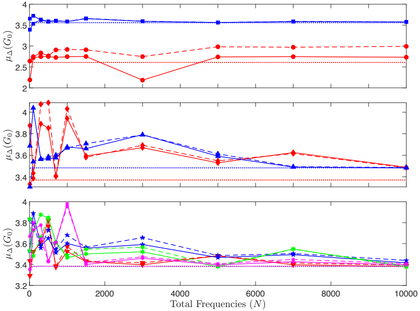
Details:

M. A. Guerrero, B. Lakshminarayanan, and C.R.R. “Data-driven estimation of structured singular values”. *IEEE L-CSS*, 2025.

Simulations

- MIMO system with $\boldsymbol{\Delta}, \boldsymbol{G}_0 \in \mathbb{C}^{3\times 3}$
- We benchmark against the Matlab command `mussv`
- 8 different uncertainty structures:

Case	Scalar blocks	Full blocks
1	[3]	-
2	-	[3]
3	[2]	[1]
4	[1, 1]	[2]
5	[1, 1]	[1]
6	[1]	[1]
7	[1, 1, 1]	-
8	-	[1, 1, 1]



Digression: Counter-factual explainability

- Machine learning (ML) and data-driven solutions are being increasingly utilized in diverse application domains
- Especially for applications in high-stakes domains (law, medicine, autonomous driving, *etc.*), providing explanations for ML predictions is necessary
- Some techniques developed for explaining ML predictions can be profitably adapted to address problems in cyber-security

Digression: Counter-factual explainability (cont.)

- Imagine you apply for a bank loan
- The bank uses an ML algorithm to analyze your case …
- … and rejects your application

- In the EU, one has the *right to an explanation*
- What does an "explanation" mean?

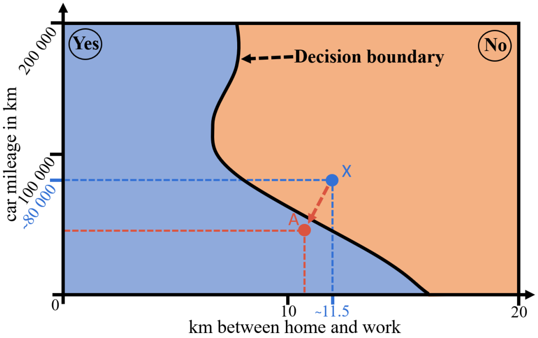
- One possibility: **counterfactual explainability** (CFE)

$\boldsymbol{X}$: input vector
$y = f(\boldsymbol{X})$: output of classifier (yes / no)

what is the smallest perturbation to $\boldsymbol{X}$ *needed to change the classifier output?*

This is a generalization of the vulnerability identification problem!

Digression: Counter-factual explainability (cont.)



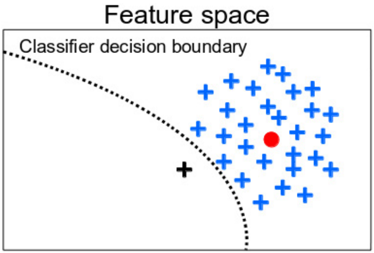
CFE optimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}} \quad & d(\boldsymbol{X},\boldsymbol{A}) \\ \text{s.t.} \quad & f(\boldsymbol{A}) \neq f(\boldsymbol{X}) \end{aligned}$$

Digression: Counter-factual explainability (cont.)

Problem: f is often a black box!

One approach: *Growing spheres algorithm*



Problem: Potentially many evaluations of f are needed!

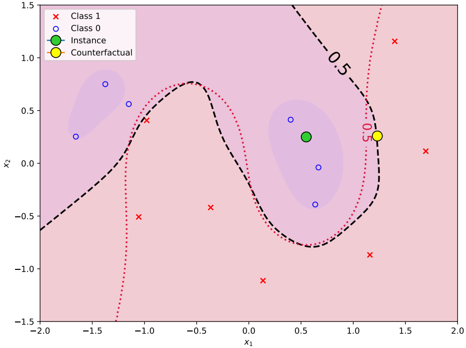
An alternative: *Bayesian Optimization*

- Model $f(\boldsymbol{X}) = \mathrm{sgn}(h(\boldsymbol{X}))$, where h is a Gaussian process
- Write CFE problem as $c(\boldsymbol{A}) = d(\boldsymbol{X}, \boldsymbol{A}) + \lambda\,[h(\boldsymbol{A})]^2 \quad (\lambda \gg 1)$
- Apply Bayesian optimization to adaptively sample points $\boldsymbol{A}$

Difficulty: For CFE, h is not directly measured!

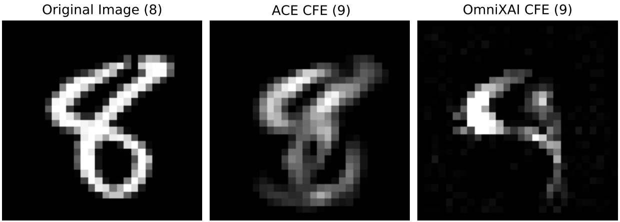
Digression: Counter-factual explainability (cont.)

Make Moons dataset (low-dimensional visualization):



Digression: Counter-factual explainability (cont.)

MNIST dataset (high-dimensional visualization):



Details:

M. A. Guerrero and C.R.R. “ACE: Adapting sampling for Counterfactual Explanations ”. arXiv: 2509.26322.

Motivation

- Most theoretical results in identification are asymptotic: *e.g.*,

$$\sqrt{N}(\hat{\boldsymbol{\theta}}_N - \boldsymbol{\theta}_0) \xrightarrow[N\to\infty]{d} \mathcal{N}(\mathbf{0}, \boldsymbol{P}).$$

- Although asymptotically tight, these bounds may not be good for small sample sizes.
- In recent years, drawing from research in computer science, statistics and machine learning, several types of finite sample bounds for system identification procedures have been derived.
- These bounds yield valid, high probability confidence regions for identification estimates, and provide guidance on how several user choices affect their accuracy.
- Here we will provide an example related to subspace identification...

Subspace Identification

with J. He (KTH), H. Hjalmarsson (KTH),
S. J. Qin (City U. Hong Kong) and I. Ziemann (U. Pennsylvania)

Subspace identification

- State-space model:
  - innovations form: $\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{K}\boldsymbol{e}_k \\ \boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{e}_k \end{cases}$
- Assumptions:
  - Stable: $\rho(\boldsymbol{A}) < 1$ and $\rho(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}) < 1$
  - Minimal: $(\boldsymbol{A}, [\boldsymbol{B}, \boldsymbol{K}])$ is controllable and $(\boldsymbol{A}, \boldsymbol{C})$ is observable
  - Innovations: $\boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_e^2 \boldsymbol{I})$
  - Inputs: $\boldsymbol{u}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_u^2 \boldsymbol{I})$, and independent of $\{\boldsymbol{e}_k\}$

Subspace identification (cont.)

Data collection:

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{y}_{k+1} \\ \vdots \\ \boldsymbol{y}_{k+f} \end{bmatrix}}_{\boldsymbol{y}_f(k)} = \underbrace{\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CA} \\ \vdots \\ \boldsymbol{CA}^f \end{bmatrix}}_{\Gamma_f} \boldsymbol{x}_k + \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CB} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{f-1}\boldsymbol{B} & \boldsymbol{CA}^{f-2}\boldsymbol{B} & \cdots & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{G}_f} \underbrace{\begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+f} \end{bmatrix}}_{\boldsymbol{u}_f(k)} + \boldsymbol{H}_f \underbrace{\begin{bmatrix} \boldsymbol{e}_k \\ \boldsymbol{e}_{k+1} \\ \vdots \\ \boldsymbol{e}_{k+f} \end{bmatrix}}_{\boldsymbol{e}_f(k)}$$

Extended state-space model: $\boldsymbol{Y}_f = \Gamma_f \boldsymbol{X}_k + \boldsymbol{G}_f \boldsymbol{U}_f + \boldsymbol{H}_f \boldsymbol{E}_f$, where

$$\boldsymbol{Y}_f = \begin{bmatrix} \boldsymbol{y}_f(1) & \boldsymbol{y}_f(2) & \cdots & \boldsymbol{y}_f(N) \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}_k & \boldsymbol{y}_{k+1} & \cdots & \boldsymbol{y}_{k+N-1} \\ \boldsymbol{y}_{k+1} & \boldsymbol{y}_{k+2} & \cdots & \boldsymbol{y}_{k+N} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{y}_{k+f-1} & \boldsymbol{y}_{k+f} & \cdots & \boldsymbol{y}_{k+f+N-2} \end{bmatrix}$$

$$\boldsymbol{X}_k = \begin{bmatrix} \boldsymbol{x}_k & \boldsymbol{x}_{k+1} & \cdots & \boldsymbol{x}_{k+N-1} \end{bmatrix}$$

- Recover the current state $\boldsymbol{X}_k$ from past input and output:

$$\boldsymbol{X}_k = \boldsymbol{L}_p \boldsymbol{Z}_p + \boldsymbol{A}_K^p \boldsymbol{X}_{k-p}, \text{where}$$

$$\boldsymbol{L}_p = \begin{bmatrix} \boldsymbol{A}_K^{p-1}\boldsymbol{K} & \cdots & \boldsymbol{A}_K\boldsymbol{K} & \boldsymbol{K} & \boldsymbol{A}_K^{p-1}\boldsymbol{B} & \cdots & \boldsymbol{A}_K\boldsymbol{B} & \boldsymbol{B} \end{bmatrix}$$

$$\boldsymbol{A}_K = \boldsymbol{A} - \boldsymbol{K}\boldsymbol{C},\ \boldsymbol{Z}_p = \begin{bmatrix} \boldsymbol{Y}_p^T & \boldsymbol{U}_p^T \end{bmatrix}^T$$

- Replacing $\boldsymbol{X}_k$ yields $\boldsymbol{Y}_f = \Gamma_f \boldsymbol{L}_p \boldsymbol{Z}_p + \boldsymbol{G}_f \boldsymbol{U}_f + \boldsymbol{H}_f \boldsymbol{E}_f + \Gamma_f \boldsymbol{A}_K^p \boldsymbol{X}_{k-p}$

Subspace identification (cont.)

Roadmap of subspace identification:

$Y_f = \Gamma_f L_p Z_p + G_f U_p + H_f E_p + G_f A_K^p X_{k-p}$

Step 1: Estimate Hankel matrix $\Gamma_f L_p$ → **Error 1: $\widehat{\Gamma_f L_p} - \Gamma_f L_p$**

Step 2: Estimate Obs.&Ctrl. matrices Γ_f, L_p → **Error 2: $\widehat{\Gamma_f} - \Gamma_f T, \widehat{L_p} - T^{-1} L_p$**

Step 3: Estimate system matrices A, B, C, K → **Error 3: $\widehat{A} - T^{-1}AT$, $\widehat{C} - CT, \widehat{B} - T^{-1}B, \widehat{K} - T^{-1}K$**

Problem: Given data $\{\boldsymbol{y}_k, \boldsymbol{u}_k\}_{k=1}^N$, estimate matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{K})$ and establish their error bounds explicitly (with probability at least $1-\delta$),

$$\max\{\|\hat{\boldsymbol{A}} - \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\|, \|\hat{\boldsymbol{C}} - \boldsymbol{C}\boldsymbol{T}\|, \|\hat{\boldsymbol{B}} - \boldsymbol{T}^{-1}\boldsymbol{B}\|, \|\hat{\boldsymbol{K}} - \boldsymbol{T}^{-1}\boldsymbol{K}\|\}$$

$$\leqslant \mathcal{E} \propto (\text{Noise level}) \times \sqrt{\frac{\text{problem dimension} + \log(1/\delta)}{\text{sample size}}}$$

Subspace identification (cont.)

Tricky part: $\widehat{\Gamma_f L_p}$ is non-causal for most subspace methods, *i.e.*, does not have a self-normalized martingale structure

PARSIM (Qin et.al., 2005) overcomes this issue as follows:

- Utilize the structure of G_f:

$$\underbrace{\begin{bmatrix} \boldsymbol{Y}_{f1} \\ \boldsymbol{Y}_{f2} \\ \vdots \\ \boldsymbol{Y}_{ff} \end{bmatrix}}_{\boldsymbol{Y}_f} = \underbrace{\begin{bmatrix} \boldsymbol{\Gamma}_{f1} \\ \boldsymbol{\Gamma}_{f2} \\ \vdots \\ \boldsymbol{\Gamma}_{ff} \end{bmatrix}}_{\boldsymbol{\Gamma}_f} \boldsymbol{L}_p \boldsymbol{Z}_p + \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{CB} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{CA}^{f-1}\boldsymbol{B} & \boldsymbol{CA}^{f-2}\boldsymbol{B} & \cdots & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{G}_f} \underbrace{\begin{bmatrix} \boldsymbol{U}_{f1} \\ \boldsymbol{U}_{f2} \\ \vdots \\ \boldsymbol{U}_{ff} \end{bmatrix}}_{\boldsymbol{U}_i} + \cdots$$

$$\boldsymbol{Y}_{fi} = \boldsymbol{\Gamma}_{fi} \boldsymbol{L}_p \boldsymbol{Z}_p + \boldsymbol{G}_{fi} \boldsymbol{U}_i + \boldsymbol{H}_{fi} \boldsymbol{E}_i + \boldsymbol{\Gamma}_{fi} \boldsymbol{A}_K^p \boldsymbol{x}_k, \quad i = 1, 2, \ldots, f$$

- Estimate several ARX models in parallel:

$$\hat{\boldsymbol{\theta}}_i := [\widehat{\boldsymbol{\Gamma}_{fi} \boldsymbol{L}_p} \quad \hat{\boldsymbol{G}}_{fi}] = \boldsymbol{Y}_{fi} \begin{bmatrix} \boldsymbol{Z}_p \\ \boldsymbol{U}_i \end{bmatrix}^{\dagger} \quad \text{(a self-normalized martingale!)}$$

- Stack f estimates together: $\widehat{\boldsymbol{\Gamma}_f \boldsymbol{L}_p} = \begin{bmatrix} \widehat{\boldsymbol{\Gamma}_{f1} \boldsymbol{L}_p} \\ \vdots \\ \widehat{\boldsymbol{\Gamma}_{ff} \boldsymbol{L}_p} \end{bmatrix}$

Subspace identification (cont.)

Details:

- J. He, I. Ziemann, C.R.R. and H. Hjalmarsson. "Finite sample analysis for a class of subspace identification methods. *CDC*, 2024.
- J. He, I. Ziemann, C.R.R., S. J. Qin and H. Hjalmarsson. "Finite sample analysis of open-loop subspace identification methods". arXiv: 2501.16639.

Conclusions

- Many connections between identification and machine learning!
  - Counter-factual explainability
  - Bandits
  - Finite sample analysis, . . .
- Other areas of overlap (not discussed):
  - Causal inference vs. closed-loop identification
  - Simulation-driven estimators, . . .

Thank you for your attention.

`people.kth.se/~crro`