Non-convexity Analysis: Online Optimization and Online Learning

Javad Lavaei

University of California, Berkeley

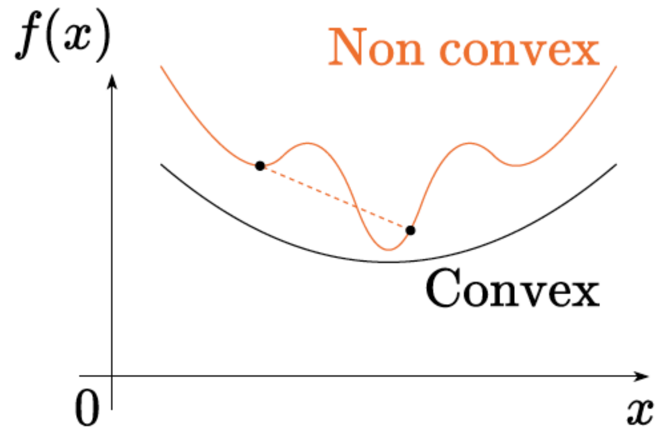
Outline

- Online optimization
  - Introduce sequential decision-making
  - Connection to projected gradient ODEs
  - Study dynamic regret
- Online learning (joint work with Eduardo Sontag)
  - Study learning of nonlinear systems under adversaries
  - Bounded and Lipschitz cases

Online Optimization

- Classic optimization: $\min_{x} f(x)$

- The area of convex optimization is rich, and the area of non-convex optimization is at the core of AI.



- What if we have a sequence of functions to minimize?

$$\min_{x} f_1(x) \Rightarrow \min_{x} f_2(x) \Rightarrow \cdots \Rightarrow \min_{x} f_T(x)$$

- In online optimization, we often assume that we do not know the current function until we select a solution, and so there is a delay.

Online Optimization

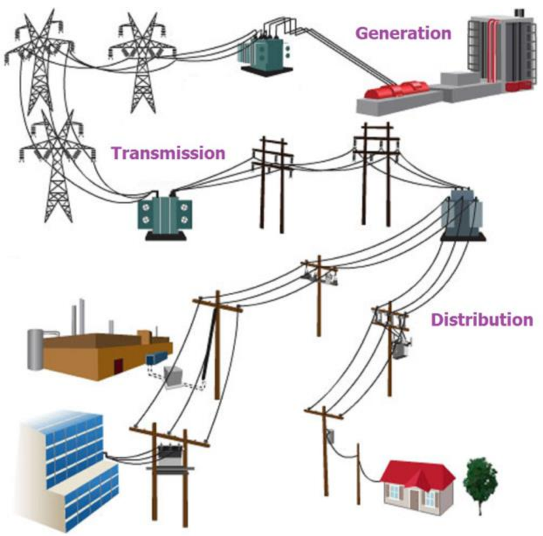
- If we treat this as *T* separate problems, it becomes classic optimization.
- However, in sequential decision-making, the functions are related to each other:
  - ✓ Gradual change in function: $\|f_t(x) - f_{t-1}\| \leq r$
  - ✓ Gradual change in solution: $\|x_t^* - x_{t-1}^*\| \leq r$

Global minima

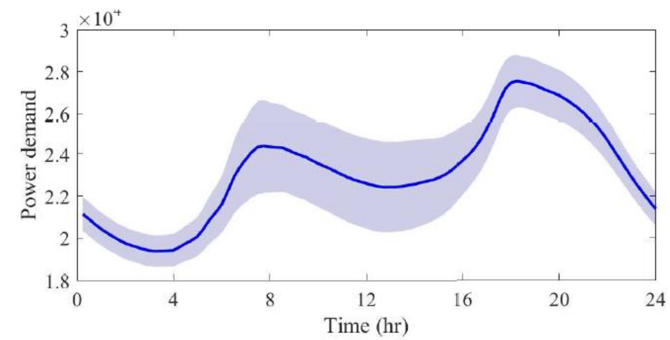
- In classic optimization, computational complexity is well studied through the notions of convex optimization, number of spurious solutions, etc.
- How to measure the computational complexity of a sequence of problems rather than a single one?

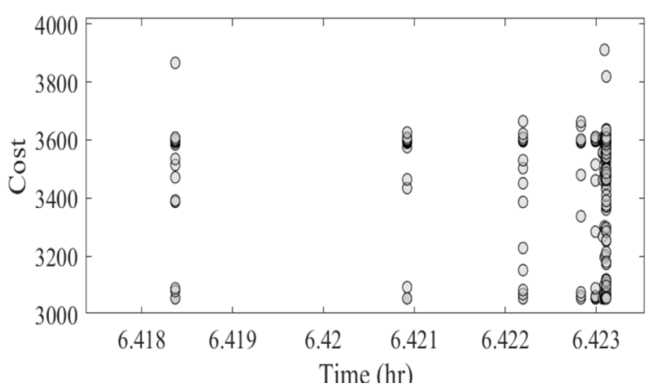
Online Optimization

❑ Motivating example: Time-varying optimal power flow

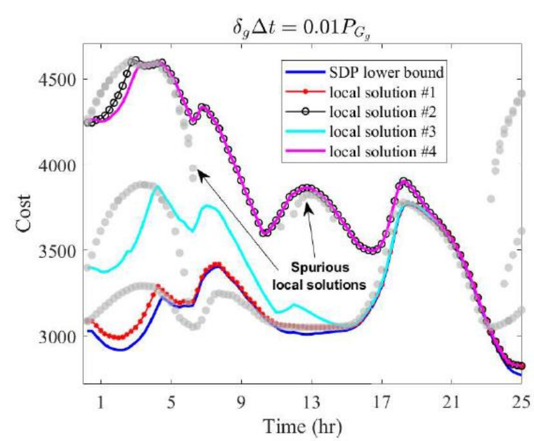


Average daily data for California in Jan. 2019





Many local minima at different times

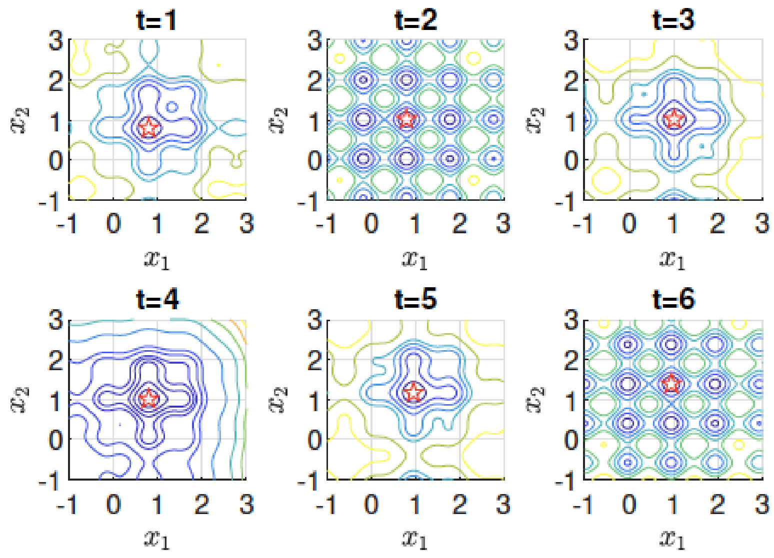


Online trajectories converge

Start at a local min, but tomorrow the same time, the solution is global.

Online Optimization

- Examples of contours of a sequence of bivariate functions, showing regions of attraction of different local minima:



- Hard problem at time 2 and easy problem at time 5.

- Question: If there is a single easy problem in the sequence, will it break down computational complexity in the future of the sequence?

Online Optimization

- Assumed we have already solved: $\min_x f_{t-1}(x)$
- To solve $\min_x f_t(x)$, we do the following:
  - Use a local search algorithm initialized at last optimal solution.
  - Generate a number of random initial points and run local search.
  - Select best solution obtained from random initialization and historical initialization.
- More general case: Sequential decision-making subject to constraints

$$\min_x f_t(x) \quad \text{subject to} \quad x \in \mathcal{S}$$

- Assume objective function is non-convex while feasible set is convex.
- Our approach: ODE models of numerical algorithms

Online Optimization

- Polyak-Łojasiewicz inequality for unconstrained problems:

$$\underbrace{\frac{1}{2}||\nabla f_t(\mathbf{x})||^2 \geq \mu(f_t(\mathbf{x}) - f_t^*)}_{\text{PL inequality}} \quad \forall \mathbf{x} \in \mathbb{R}^n$$

- This is more general than convex optimization but still makes the problem tractable.

- Proximal Polyak-Łojasiewicz inequality for constrained problems:

$$\frac{1}{2}\mathcal{D}_t(\mathbf{x}, \beta) \geq \mu(f_t(\mathbf{x}) - f_t^*)$$

where

$$\mathcal{D}_t(\mathbf{x}, \beta) = -2\beta \min_y \left[\langle \nabla f_t(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{\beta\|\mathbf{y} - \mathbf{x}\|^2}{2} + \mathbb{I}_{\mathbb{S}}(\mathbf{y}) - \mathbb{I}_{\mathbb{S}}(\mathbf{x})\right]$$

- Control theory can be used to study the above inequality.

Online Optimization

- Note that

$$\mathcal{D}_t(\mathbf{x},\beta) = -2\beta\left[\langle \nabla f(\mathbf{x}), \Pi_{\mathbb{S}}(\mathbf{x} - \frac{1}{\beta}\nabla f_t(\mathbf{x})) - \mathbf{x}\rangle + \frac{\beta}{2}\|\Pi_{\mathbb{S}}(\mathbf{x} - \frac{1}{\beta}\nabla f_t(\mathbf{x})) - \mathbf{x}\|^2\right]$$

projected gradient flow system

$$\dot{\mathbf{x}}_t(\ell) = \Pi_{\mathbb{T}_{\mathbb{S}}(\mathbf{x}_t)}[-\nabla f_t(\mathbf{x}_t(\ell))]$$

Projection on tangent cone

- Gradient ODEs are well studied but projected gradient ODEs are less studied.

- Proximal-PL region:

$$\mathcal{P}_t(\mu,\beta) := \left\{\mathbf{x} \in \mathbb{S} \;\middle|\; \frac{1}{2}\mathcal{D}_t(\mathbf{x},\beta) \geq \mu(f_t(\mathbf{x}) - f_t^*)\right\}$$

Online Optimization

- Consider a subset of the discrete RoA contained in the proximal-PL region:

$$\mathcal{RP}_t^D(\mu,\beta,s) := \{\mathbf{x} \mid \mathbf{x}_t^{k+1} = \Pi_{\mathbb{S}}(\mathbf{x}_t^k - s\nabla f_t(\mathbf{x}_t^k)),\ \mathbf{x}_0 = \mathbf{x},\ \lim_{k\to\infty} \mathbf{x}_t^k = \mathbf{x}_t^* \ \textit{and}\ \{\mathbf{x}_t^k\}_{k=0}^{\infty} \subset \mathcal{P}_t(\mu,\beta)\}$$

- Consider a subset of the continuous RoA contained in the proximal-PL region:

$$\mathcal{RP}_t^C(\mu,\beta) := \{\mathbf{x} \mid \dot{\mathbf{x}}_t = \Pi_{\mathbb{T}_{\mathbb{S}}(\mathbf{x}_t)}(-\nabla f_t(\mathbf{x}_t)), \mathbf{x}_t(0) = \mathbf{x},\ \lim_{\ell\to\infty} \mathbf{x}_t(\ell) = \mathbf{x}_t^* \ \textit{and}\ \mathbf{x}_t(\ell) \in \mathcal{P}_t(\mu,\beta)\ \forall \ell \geq 0\}$$

- We care about intersection of both ROA-type regions and call it a target region.
- By studying the projected gradient ODE, we proved quadratic growth over target region:

$$\sqrt{f_t(\mathbf{x}) - f_t^*} \geq \sqrt{\frac{\mu}{2}}\|\mathbf{x} - \mathbf{x}_t^*\|$$

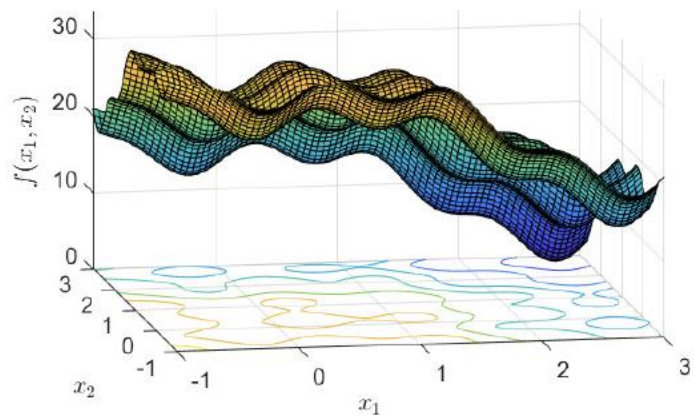
- Also, if projected gradient is initialized in the target region, we obtain a linear convergence for every iteration *N*:

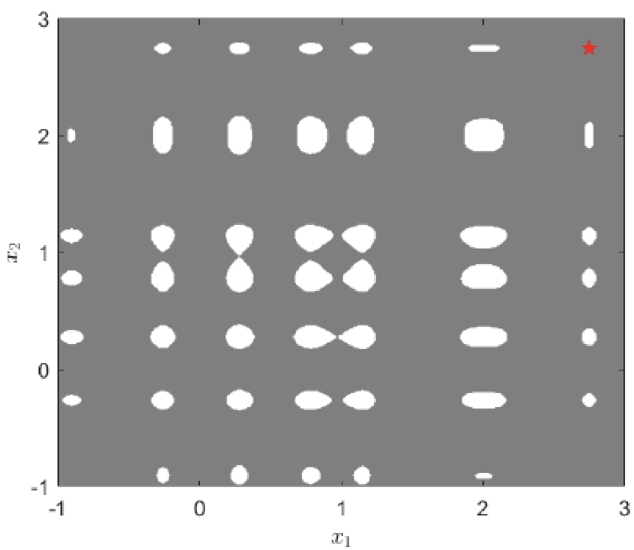
$$f_t(\mathbf{x}_t^N) - f_t^* \leq (1-\mu s)^N\,[f_t(\mathbf{x}_t^0) - f_t^*],$$

Online Optimization

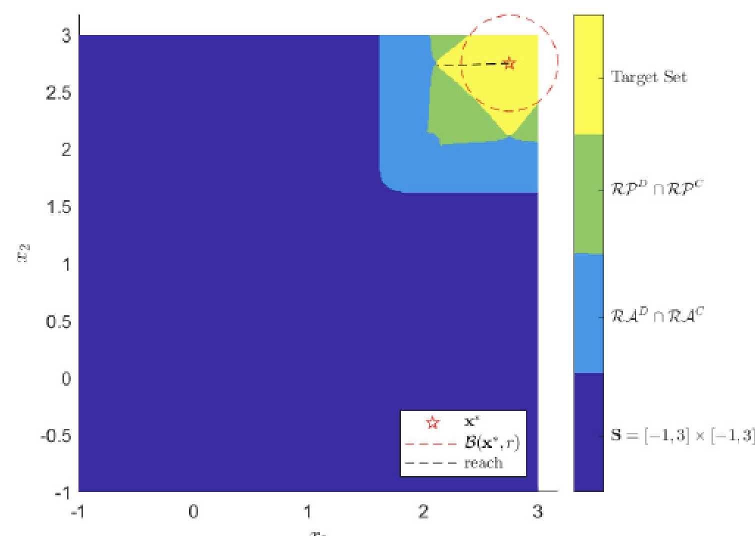
- Illustration of different types of regions of attractions:

$$
\begin{aligned}
\min\ & f(x_1, x_2) = x_1^4 - 4x_1^3 + x_1^2 + 2x_1 + \frac{3}{2}\sin(2\pi x_1) \\
& \qquad + x_2^4 - 4x_2^3 + x_2^2 + 2x_2 + \frac{3}{2}\sin(2\pi x_2) + 28.87 \\
\text{s.t.}\ & -1 \le x_1 \le 3, \quad -1 \le x_2 \le 3
\end{aligned}
$$





Grey region: Proximal-PL region



Yellow region is where control theory is useful for analysis

Online Optimization

- In sequential decision-making, *N* cannot be chosen a large number since we need to move on whenever a new problem arrives:

$$\min_x f_{t-1}(x) \longrightarrow \min_x f_t(x)$$

Can't fully optimize due to deadlines

Algorithm 1 Online Projected Gradient Descent with Desirable Initialization

Require: $\mathbf{x}_1 \in \mathcal{T}_1(\mu, \beta, s)$, $0 < s < \min\{\frac{1}{\mu}, \frac{1}{\beta}\}$

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Play $\mathbf{x}_t$
3: Set $\mathbf{z}_0 = \mathbf{x}_t$
4: **for** $i = 1, ..., S_t$ **do**
5: Observe $\nabla f_t(\mathbf{z}_{i-1})$
6: Perform projected gradient descent update: $\mathbf{z}_i = \Pi_{\mathbb{S}}\left[\mathbf{z}_{i-1} - s\nabla f_t(\mathbf{z}_{i-1})\right]$
7: **end for**
8: Set $\mathbf{x}_{t+1} = \mathbf{z}_{S_t}$
9: **end for**

Theorem: If the initial point is the target set of problem at some time *t=k*, the future points generated by Algorithm 1 will all remain in the target sets of problems at *t=k+1,k+*2,…,*T*.

Online Optimization

- Define the notion of dynamic regret:

$$\mathbf{Reg}_T^d(\mathbf{x}_1, \ldots, \mathbf{x}_T) := \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t^*$$

Solution returned by Algorithm 1

- We want to prove that the regret for a large *T* depends on the first time we observe an easy problem in the sequence.

- If the problem at time *t=1* is easy, then

$$\boldsymbol{Reg}_T^d(\mathbf{x}_1,\ldots,\mathbf{x}_T) \leq \frac{M_1}{1-\gamma} \sum_{t=2}^{T} \|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\| + \frac{\rho_1(\mu, \beta, s)}{(1-\gamma)/M_1}$$

- This is the first result in the literature in the non-convex case.

- Can we improve it by generating random initial points?

Online Optimization

Algorithm 2 Online Projected Gradient Descent with Random Exploration

Require: $\mathbf{x}_1 \in \mathbb{S}$, $\mathcal{M}_1 = \emptyset$, $m = 0$, $0 < s < \min\{\frac{1}{\mu}, \frac{1}{\beta}\}$
for $t = 1, 2, \ldots, T$ **do**
  • Play $\mathbf{x}_t$
  • Create $\mathcal{W}_t = \{\mathbf{w}_t^1, \ldots, \mathbf{w}_t^q\}$ by uniformly sampling q random points from $\mathbb{S}$
  • Set $\mathcal{Y}_t = \mathcal{W}_t \bigcup \mathcal{M}_t \bigcup \{\mathbf{x}_t\} := \{\mathbf{y}_t^1, \ldots, \mathbf{y}_t^{q+m+1}\}$
  for $k = 1, 2, \ldots, q+m+1$ **do**
    • Initialize $\mathbf{z}_0^k = \mathbf{y}_t^k$, $\mathbf{z}_*^k = \mathbf{y}_t^k$, $\underline{c}_k = \infty$, $\overline{b}_k = -\infty$
    • Set $i = 1$
    while $\underline{c}_k - \overline{b}_k > \epsilon$ or $i \le S_t$ **do**
      • Observe $\nabla f_t(\mathbf{z}_{i-1}^k)$
      • Compute $\mathbf{z}_i^k = \Pi_{\mathbb{S}}\left[\mathbf{z}_{i-1}^k - s\nabla f_t(\mathbf{z}_{i-1}^k)\right]$
      • Observe $c_i^k = f_t(\mathbf{z}_i^k)$
      if $c_i^k < \underline{c}^k$ **then**
        • $\mathbf{z}_*^k = \mathbf{z}_i^k$, $\underline{c}_k = c_i^k$
      end if
      • $b_i^k = \left(f_t(\mathbf{z}_i^k) - (1-\mu s)^i f_t(\mathbf{z}_0^k)\right) / (1-(1-\mu s)^i)$
      • Update $\overline{b}_k = \max\{\overline{b}_k, b_i^k\}$
      • Update $i = i+1$
    end while
    • Return $I_t^k = i$
  end for
  • Let $K = \operatorname{argmin}_k \underline{c}_k$, and set $\mathbf{x}_{t+1} = \mathbf{z}_*^K$
  • Store in memory all other points in $\{\mathbf{z}_*^k\}_{k=1}^{q+m+1}$ which could be in the proximal PL-region at time t: $\mathcal{M}_{t+1} = \{\mathbf{z}_*^k : \underline{c}_k \le \underline{c}_K + \epsilon, k \in \{1, \ldots, q+m+1\} \setminus K\}$
  $m = |\mathcal{M}_{t+1}|$
end for

Main idea: To solve problem at time *t*, use the solution at time *t-1* and a number of randomly generated points since problem *t* could be far away from problem *t-1*.

❑ We can relate regrets at two different times:

$$\mathbb{P}\Big[\boldsymbol{Reg}_T^d(\mathbf{x}_1, \ldots, \mathbf{x}_T) \le \boldsymbol{Reg}_{\bar{T}-1}^d(\mathbf{x}_1, \ldots, \mathbf{x}_{\bar{T}-1}) + \frac{M_1 \rho_{\bar{T}}(\mu, \beta, s)}{(1-\gamma)} + \frac{M_1}{1-\gamma}\sum_{t=\bar{T}+1}^{T} \|\mathbf{x}_t^* - \mathbf{x}_{t-1}^*\|\Big] \ge 1 - \prod_{t=1}^{T}\left(1 - \frac{Vol(\mathcal{T}_t(\mu, \beta, s))}{Vol(\mathbb{S})}\right)^q,$$

Probability due to random exploration

This measures how big the target set is compared to the feasible set

Online Optimization

$$\mathbb{P}\Bigg[\boldsymbol{Reg}_T^d(\mathbf{x}_1,\ldots,\mathbf{x}_T) \le \boldsymbol{Reg}_{\bar{T}-1}^d(\mathbf{x}_1,\ldots,\mathbf{x}_{\bar{T}-1}) + \frac{M_1\rho_T(\mu,\beta,s)}{(1-\gamma)} + \frac{M_1}{1-\gamma}\sum_{t=\bar{T}+1}^{T}\|\mathbf{x}_t^*-\mathbf{x}_{t-1}^*\|\Bigg] \ge 1-\prod_{t=1}^{T}\left(1-\frac{Vol(\mathcal{T}_t(\mu,\beta,s))}{Vol(\mathbb{S})}\right)^q,$$

- ❑ If this is 1 at a single time, the probability becomes 1 and so the regret stops growing.
- ❑ The value of this being close to 1 or 0 measures the complexity of a single problem

- ❑ The above formula shows how the complexity of each problem in the sequence affects the complexity of the entire sequence.
- ❑ A single problem in the sequence breaks down the complexity.
- ❑ Our proofs depend on analysis of continuous-time ODEs.

Outline

- ❑ Online optimization
  - ➢ Introduce sequential decision-making
  - ➢ Connection to projected gradient ODEs
  - ➢ Study dynamic regret
- ❑ Online learning (joint work with Eduardo Sontag)
  - ➢ Study learning of nonlinear systems under adversaries
  - ➢ Bounded and Lipschitz cases

Online Learning

$$x(k+1) = f(x(k), u(k), w(k))$$

$$y(k) = h(x(k), u(k), v(k))$$

Adversarial attacks

Measurements

Dynamics

Leaning x(t) from y(t) is studied in ML extensively

x(t) is manipulated and existing tools in ML do not apply directly

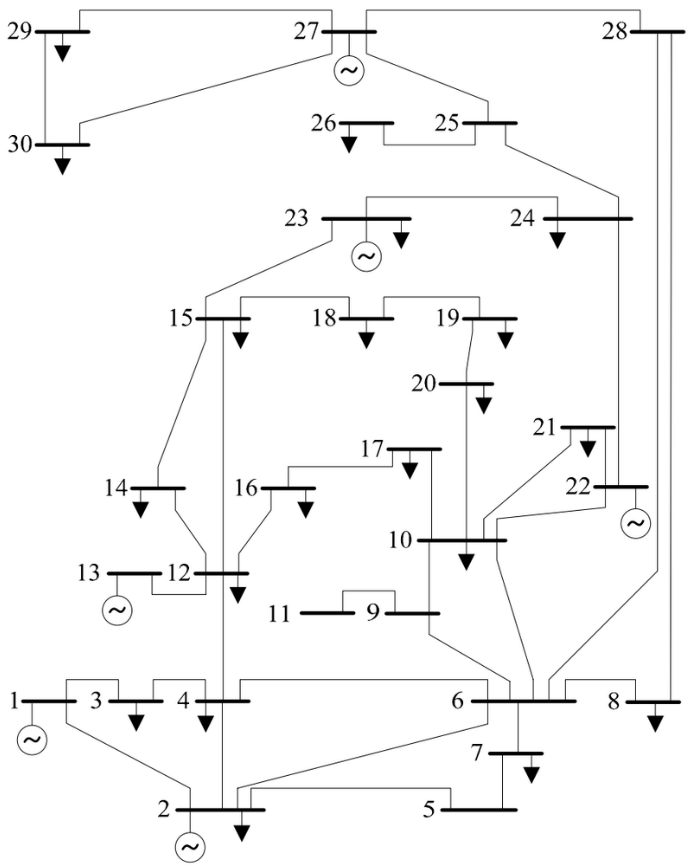
?

Learning
(learn *f*, *h* and *w*)

Control
(optimal *u*)

Online Learning

- Motivating example:
  - Consumers report their demands to suppliers
  - If the data is hacked, producers over-supply electricity
  - This is an input-type attack and makes the dynamical system experience a cascading failure.
  - Due to complexity of smart grids, we do not have models of all EVs, DERs, social behavior, etc. and so we should learn the model and possible attacks together.



Online Learning

- We write the unknown dynamics as a sum of nonlinear basis functions:

$$x_{t+1} = \bar{A} f(x_t) + \bar{d}_t, \quad \forall t = 0, \ldots, T-1,$$

Need to learn this

Vector of chosen basis functions

Unknown disturbances

- We use a non-smooth estimator:

$$\min_{A \in \mathbb{R}^{n \times m}} \sum_{t=0}^{T-1} \|x_{t+1} - A f(x_t)\|_2$$

- We assume that the attacker can attack the input with probability *p* at each time, and that the attack value is arbitrary (as long as it is stealth)

Online Learning

- Bounded case: $\|f(x)\|_\infty \leq B, \quad \forall x \in \mathbb{R}^n$

- Theorem: For every $\delta < 1$, with probability at least $1 - \delta$ the estimator recovers the correct dynamics if

$$T \geq \Theta\left[\frac{m^2\kappa^4}{p(1-p)^2}\left[mn\log\left(\frac{m\kappa}{p(1-p)}\right) + \log\left(\frac{1}{\delta}\right)\right]\right],$$

- This is the first result in the literature saying that exact recovery is possible even when *p* goes to 1 meaning that the system is constantly under attack.

- Can the bound be improved significantly? No, there is a counter example with a high probability if

$$T < \max\left\{\frac{m}{2p(1-p)}, \frac{2}{-\log[p(1-p)]}\log\left(\frac{2}{\delta}\right)\right\},$$

Online Learning

- Lipschitz case:

$$f(0_n) = 0_m \quad \textit{and} \quad \|f(x) - f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \in \mathbb{R}^n,$$

- Assume a stability-type condition: $\|\bar{A}\|_2 < \frac{1}{L}$.

- Theorem: For every $\delta < 1$, with probability at least $1 - \delta$ the estimator recovers the correct dynamics if

$$T \geq \Theta\left[\max\left\{\frac{\kappa^{10}}{(1-\rho L)^3(1-p)^2}, \frac{\kappa^4}{p(1-p)}\right\} \times \left[mn \log\left(\frac{1}{(1-\rho L)\kappa p(1-p)}\right) + \log\left(\frac{1}{\delta}\right)\right]\right]$$

- If stability is violated, there are counterexamples showing the failure of the estimator.

Online Learning

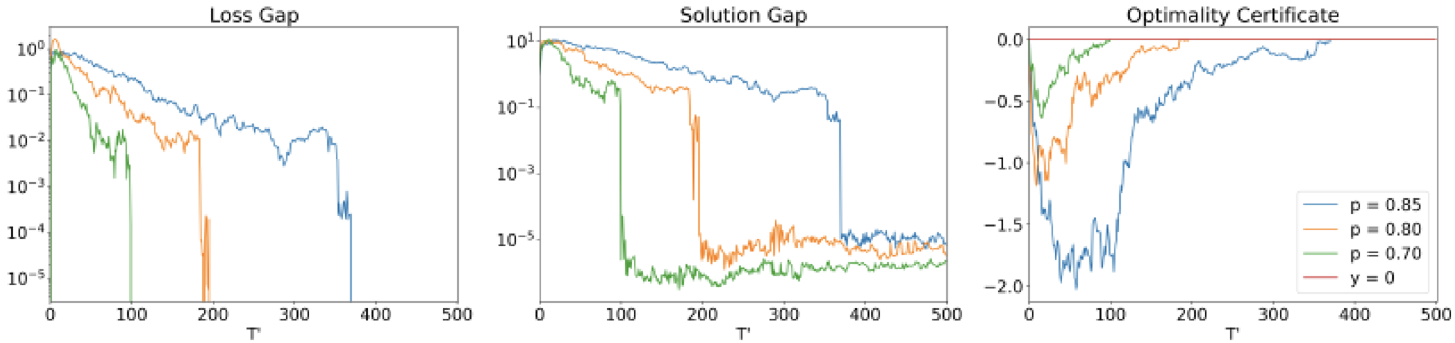


Figure 1: Loss gap, solution gap and optimality certificate of the Lipschitz basis function case with attack probability $p = 0.7, 0.8$ and 0.85.

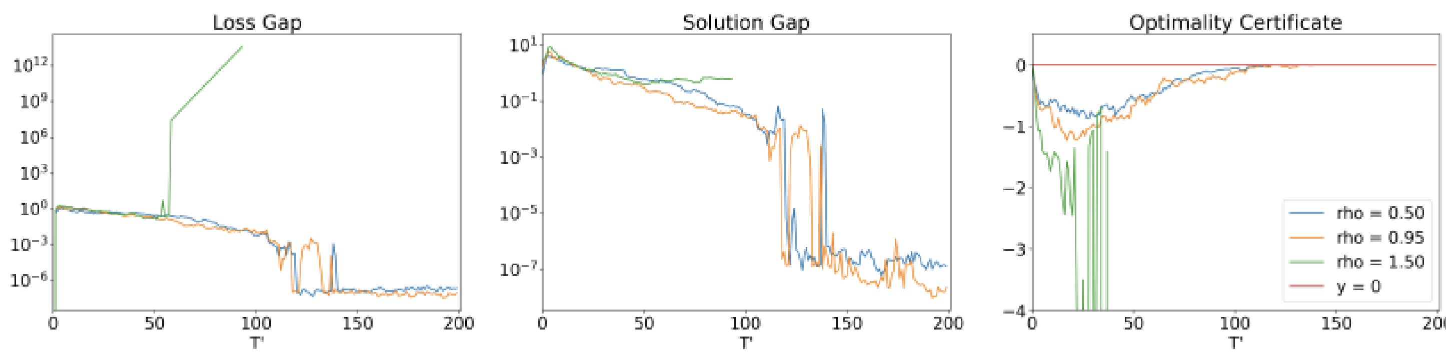


Figure 3: Loss gap, solution gap and optimality certificate of the Lipschitz basis function case with spectral norm $\rho = 0.5, 0.95$ and 1.5.

Conclusions

- Online optimization
  - Connection to projected gradient ODEs
  - Study dynamic regret
  - On easy problem in the sequence breaks down the complexity of the sequence
- Online learning (joint work with Eduardo Sontag)
  - Study learning of nonlinear systems under adversaries
  - Bounded and Lipschitz cases: First results in the literature