From Statistical Bounds to Optimization Complexity in Sparse Recovery Problems

Vincent Roulet

University of Washington

February 26th, 2021

Plan

Recovery Problems

Optimization Complexity

Condition Number

Recovery Problems

Recovery from direct measurements
Recover an unknown signal $\beta^* \in \mathbb{R}^d$ with d features from n observations

$$y_i = x_i^\top \beta^* \quad \text{for } i = 1, \ldots, n$$

Example

$\longrightarrow$

Recovery Procedures

Dense β^*
Without further assumptions, solve

$$\min_{\beta \in \mathbb{R}^d} \quad \|X\beta - y\|_2^2 \tag{LS}$$

with $y = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$, $X = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times d}$

Statistical Viewpoint

- Requires $\sigma_{\min}(X) > 0$ to recover β^*, so at least $n \geq d$ observations

Optimization Viewpoint

- Needs at most

  $$\sqrt{\kappa} \log \varepsilon^{-1} \qquad \text{where } \kappa = \sigma_{\max}(X)^2 / \sigma_{\min}(X)^2,$$

  iterations to get an ε accuracy using e.g. a conjugate gradient method

Recovery Procedures

Sparse β^*
Consider the additional assumption that β^* is k-**sparse**, i.e.,

$$\boxed{\|\beta^*\|_0 := |\{i : \beta_i^* \neq 0\}| = k \ll d}$$

Ideal Procedure
Solve

$$\begin{aligned} \min_{\beta \in \mathbb{R}^d} \quad & \|\beta\|_0 \\ \text{s.t.} \quad & y = X\beta \end{aligned}$$

Statistical viewpoint

- Needs approx. $k \log d$ random observations (Cohen et al., 2009)

Optimization viewpoint

- Cannot be solved in reasonable time

Recovery Procedures

Sparse β^* Consider the additional assumption that β^* is k-**sparse**, i.e.,

$$\boxed{\|\beta^*\|_0 := |\{i : \beta_i^* \neq 0\}| = k \ll d}$$

Dantzig selector (Candes and Tao, 2007)
Approximate $\|\beta\|_0$ by $\|\beta\|_1$

$$\begin{aligned} \min_{\beta \in \mathbb{R}^d} \quad & \|\beta\|_1 \\ \text{s.t.} \quad & y = X\beta \end{aligned} \tag{D}$$

Statistical viewpoint

- Needs approx. $k \log d$ random observations e.g. (Cohen et al., 2009)

Optimization viewpoint

- Can be solved in polynomial time

Optimization and Statistical Complexities

	Dense β^*	k-sparse β^*
Statistical Complexity (number of random observations needed to recover β^*)	d	$k \log d$
Optimization complexity (number of iterations to get an ε accuracy)	$\sqrt{\kappa} \log \varepsilon^{-1}$	$\mathbf{1/\varepsilon}$
Condition number κ	$\sigma_{\max}(X)^2/\sigma_{\min}(X)^2$	**?**

Questions

1. Can we get a better convergence in the sparse case?
2. What is the condition number in the sparse case?

Plan

Recovery Problems

Optimization Complexity

Condition Number

Optimization Method

Optimization algorithm (NESTA) (Becker et al., 2011)
A classical optimization algorithm for the Dantzig selector problem is to

1. Get an ε-accurate smooth approximation of $\|\cdot\|_1$, denoted h_ε
2. Apply an accelerated projected[1] gradient descent

$$\begin{aligned} \min_{\beta \in \mathbb{R}^d} \quad & h_\varepsilon(\beta) \\ \text{s.t.} \quad & X\beta = y \end{aligned}$$

Problems

- Convergence rate of NESTA does not depend on recovery conditions ...
- In practice, this algorithm is restarted to obtain faster convergence, but no theoretical guarantees exist ...

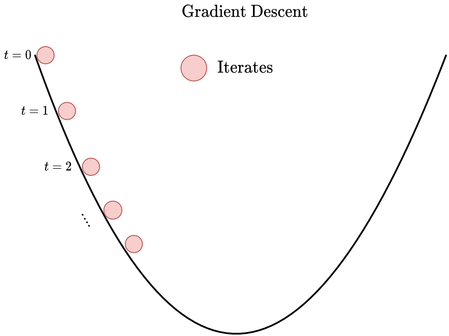
[1] The projection is assumed to be easily available

Restarts

Gradient Descent

Discretization of

$$\dot{x}(t) = -\nabla f(x(t))$$



Restarts

Gradient Descent
Discretization of

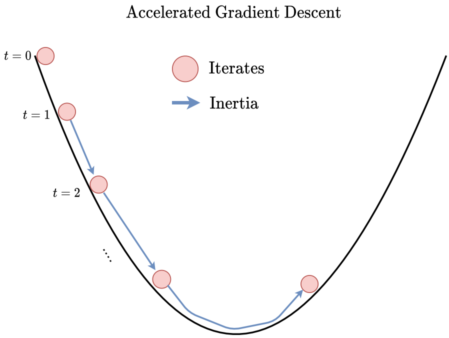
$$\dot{x}(t) = -\nabla f(x(t))$$

Accelerated gradient descent
Discretization of

$$m\ddot{x}(t) + \alpha\dot{x}(t) = -\nabla f(x(t))$$

- m mass of a ball
- α friction coefficient
- $-\nabla f(x(t))$ driving force

$\rightarrow$ The ball accumulates inertia



Restarts

Gradient Descent

Discretization of

$$\dot{x}(t) = -\nabla f(x(t))$$

Accelerated gradient descent

Discretization of

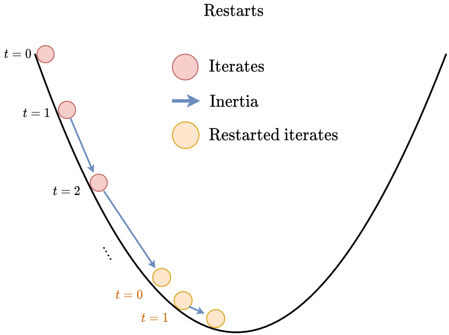
$$m\ddot{x}(t) + \alpha\dot{x}(t) = -\nabla f(x(t))$$

- m mass of a ball
- α friction coefficient
- $-\nabla f(x(t))$ driving force

$\rightarrow$ The ball accumulates inertia

Restarts

- Stop the ball at some time (cancel the inertia of the ball)
- Restart the movement from last position



Scheduled Restarts

Formalization
The NESTA algorithm can be summarized as a procedure

$$\mathcal{A} : \beta_0, \varepsilon, t \to \hat{\beta}$$

where

- β_0 is the initial point
- ε is the target accuracy (controls the approximation of $\|\cdot\|_1$)
- t is the number of iterations
- $\hat{\beta}$ is the output

Scheduled restarts
Restart the algorithm from last iterate after some number of iterations, i.e., build a sequence

$$x_i = \mathcal{A}(x_{i-1}, \varepsilon_i, t_i)$$

with

- $\varepsilon_i = \varepsilon_{i-1}/2$ (smaller target accuracy at each restart)
- t_i chosen in advance

Error Bound

Why do restarts accelerate convergence for sparse recovery problems?
Convexity is not enough to explain the phenomenon

Definition (Error bound)

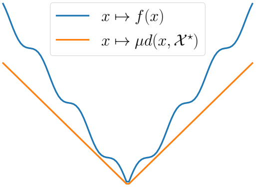
A function f is said to satisfy an error bound of order 1 with param. μ if

$$f(x) - \min_x f(x) \geq \mu \,\mathrm{dist}(x, \mathcal{X}^\star) \qquad \text{(EB)}$$

where $\mathrm{dist}(x, \mathcal{X}^\star)$ is the Euclidean distance from x to $\mathcal{X}^\star = \arg\min_x f(x)$.

Idea:
The objective f is a good surrogate for the distance to the set of minimizers $\mathcal{X}^\star$

See e.g. (Bolte et al., 2017)



Non-convex function that satisfies (EB)

Linear Convergence with Restarts

Without restarts (Nesterov, 2005)
After N iterations, NESTA outputs $\hat{\beta}$ s.t.

$$\|\hat{\beta}\|_1 - \|\beta^\star\|_1 \leq \frac{2d\|\beta_0 - \beta^\star\|_2^2}{\varepsilon N^2} + \frac{\varepsilon}{2}$$

where $\beta^\star$ a minimizer of the Dantzig selector problem

Proposition (R. et al., 2020a)

Assume that the sparse recovery problem satisfies an error bound,

$$\|\beta\|_1 - \|\beta^\star\|_1 \geq \mu \, \mathrm{dist}(\beta, \mathcal{B}^\star) \quad \textit{for any } \beta \in \mathbb{R}^d \textit{ s.t. } X\beta = y$$

where $\beta^\star \in \mathcal{B}^\star$ and $\mathcal{B}^\star$ is the set of minimizers of the problem.

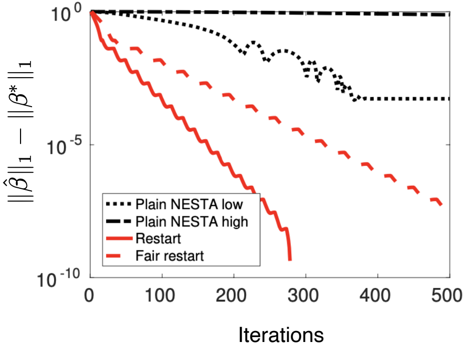
After N total number of iterations, optimal scheduled restarts output $\hat{\beta}$ s.t.

$$\|\hat{\beta}\|_1 - \|\beta^\star\|_1 \leq \mathcal{O}(\exp(-\mu N))$$

Take-aways:

- Using restarts we get an exponential convergence rate
- If μ is unknown adaptive strategies are optimal up to a logarithmic factor

Plain NESTA vs NESTA with Restarts



- Best restarted NESTA (solid red line)
- Practical restart schemes (dashed red line)
- Plain NESTA with low accuracy $\varepsilon = 10^{-1}$ (dotted black line)
- Plain NESTA with higher accuracy $\varepsilon = 10^{-3}$ (dash-dotted black line)

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \qquad \textit{(Non-Smooth)}$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2 \qquad \text{(Smooth)}$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x-y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$

$$f(x) - \min_x f(x) \geq \mu \, \mathrm{dist}(x, \mathcal{X}^\star) \qquad \textit{(Sharp Error Bound)}$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ *s.t.*

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$

$$f(x) - \min_x f(x) \geq \mu \, \mathrm{dist}(x, \mathcal{X}^\star)^2 \qquad \textit{(Quadratic Error Bound)}$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ *s.t.*

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$

$$f(x) - \min_x f(x) \geq \mu \,\mathrm{dist}(x, \mathcal{X}^\star)^r \qquad {\scriptstyle (s \leq r)} \quad \textit{(Hölderian Error Bound)}$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$

$$f(x) - \min_x f(x) \geq \mu \, \mathrm{dist}(x, \mathcal{X}^\star)^r \qquad {\scriptstyle (s \leq r)} \quad \textit{(Hölderian Error Bound)}$$

Consider the optimal algorithm $\mathcal{A}$ for convex, Hölder smooth functions with rate of convergence after N iterations,

$$f(\hat{x}) - \min_x f(x) \leq \mathcal{O}(N^{-\rho})$$

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$

$$f(x) - \min_x f(x) \geq \mu \,\mathrm{dist}(x, \mathcal{X}^\star)^r \qquad {\scriptstyle (s \leq r)} \quad \textit{(Hölderian Error Bound)}$$

Consider the optimal algorithm $\mathcal{A}$ for convex, Hölder smooth functions with rate of convergence after N iterations,

$$f(\hat{x}) - \min_x f(x) \leq \mathcal{O}(N^{-\rho})$$

then optimal/adaptive restarts of $\mathcal{A}$ output $\hat{x}$ s.t.

$$f(\hat{x}) - \min_x f(x) \leq \left\{ \begin{array}{cc} \mathcal{O}(\exp(-N)) & \text{if} \quad s = r \\ \mathcal{O}(N^{-\rho/(1-s/r)}) & \text{if} \quad s < r \end{array} \right\} \leq \mathcal{O}(N^{-\rho})$$

where N is the total number of iterations of the algorithm.

Optimal Convergence Rates with Restarts

More generally consider the problem

$$\min_x \quad f(x)$$

Proposition (R. and d'Aspremont, 2020)

Consider f convex and $L, \mu > 0$ s.t.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2^{s-1} \qquad {\scriptstyle (1 \leq s \leq 2)} \quad \textit{(Hölder smooth)}$$
$$f(x) - \min_x f(x) \geq \mu \operatorname{dist}(x, \mathcal{X}^\star)^r \qquad {\scriptstyle (s \leq r)} \quad \textit{(Hölderian Error Bound)}$$

Consider the optimal algorithm $\mathcal{A}$ for convex, Hölder smooth functions with rate of convergence after N iterations,

$$f(\hat{x}) - \min_x f(x) \leq \mathcal{O}(N^{-\rho})$$

then optimal/adaptive restarts of $\mathcal{A}$ output $\hat{x}$ s.t.

$$f(\hat{x}) - \min_x f(x) \leq \left\{ \begin{array}{ccc} \mathcal{O}(\exp(-N)) & \text{if} & s = r \\ \mathcal{O}(N^{-\rho/(1-s/r)}) & \text{if} & s < r \end{array} \right\} \leq \mathcal{O}(N^{-\rho})$$

where N is the total number of iterations of the algorithm.

Take-away

- Restarts can exploit the error bound property of the objective to get **exponential** or **faster** convergence rates than without restarts

Plan

Recovery Problems

Optimization Complexity

Condition Number

How to Uncover an Error Bound

Condition for exact recovery

For a given β^* s.t. $y = X\beta^*$, the Dantzig selector problem

$$\min_{\beta \in \mathbb{R}^d} \quad \|\beta\|_1 \qquad \text{(D)}$$
$$\text{s.t.} \quad y = X\beta$$

recovers the original signal if there exists **no** $\beta \neq \beta^*$ s.t.

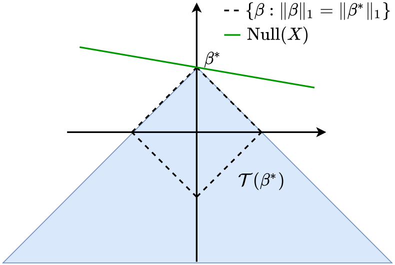
$$y = X\beta \qquad \text{and} \qquad \|\beta\|_1 \leq \|\beta^*\|_1$$

In terms of descent direction

There is no $z = (\beta - \beta^*) \neq 0$ s.t.

$$Xz = 0 \quad \text{and} \quad z \in \mathcal{T}(\beta^*)$$

where $\mathcal{T}(\beta^*)$ is the cone of descent directions for $\|\cdot\|_1$ at β^*



$\mathcal{T}(\beta^*) := \text{cone}\{z : \|\beta^* + z\|_1 \leq \|\beta^*\|_1\}$

Condition for Exact Recovery as Conic Infeasibility Problem

Formulation as an infeasibility problem

Assessing exact recovery is then equivalent to assess the infeasibility of

$$\begin{aligned} \text{find} \quad & z \\ \text{s.t.} \quad & Xz = 0 \\ & z \in \mathcal{T} \setminus \{0\} \end{aligned} \qquad (P_{X,\mathcal{T}})$$

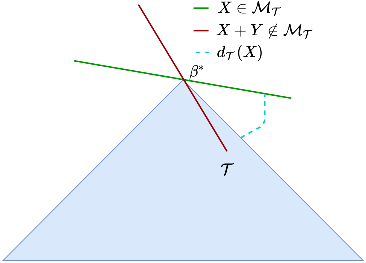
where $\mathcal{T} = \mathcal{T}(\beta^*)$

Distance to infeasibility

Let $\mathcal{M}_{\mathcal{T}} = \{X : P_{X,\mathcal{T}} \text{ is infeasible}\}$

$$d_{\mathcal{T}}(X) = \inf_{Y}\{\|Y\|_2 \text{ s.t. } X+Y \notin \mathcal{M}_{\mathcal{T}}\}$$

the distance to infeasibility of $P_{X,\mathcal{T}}$



Condition Number and Error Bounds

Definition (Condition Number)

Define the condition number of solving $P_{X,\mathcal{T}}$ as

$$C_{P_{X,\mathcal{T}}} := \frac{\|X\|_2}{d_{\mathcal{T}}(X)}$$

Proposition (R. et al., 2020a)

If $C_{P_{X,\mathcal{T}}} < +\infty$, then the Dantzig selector problem satisfies the error bound

$$\|\beta\|_1 - \|\beta^*\|_1 \geq (2C_{P_{X,\mathcal{T}}} - 1)^{-1}\|\beta - \beta^*\|_1$$
$$\text{for all } \beta \in \mathbb{R}^d \text{ s.t. } \quad X\beta = X\beta^*,$$

which ensures that

- β^* *is the unique minimizer*
- *Number of total iterations of restarts to get ε accuracy is bounded by*

$$\mathcal{O}(C_{P_{X,\mathcal{T}}} \log \varepsilon^{-1})$$

Link with Usual Exact Recovery Conditions

Proposition (Freund and Vera, 1999)

The distance to infeasibility for $P_{X,\mathcal{T}}$ can be expressed as

$$d_{\mathcal{T}}(X) = \min_{\substack{\beta\in\mathcal{T}\\ \|\beta\|_2=1}} \|X\beta\|_2 := \sigma_{\min,\mathcal{T}}(X)$$

*i.e., it is **the minimal conically restricted singular value** of X.*

Minimal Conically Restricted Singular Values in recovery Problems

- (Bickel et al., 2009) if $\sigma_{\min,\mathcal{T}}(X) > 0$,
  then exact recovery for the Dantzig selector is ensured
- (Bickel et al., 2009) $\sigma_{\min,\mathcal{T}}(X)$ controls the oracle performance of the Lasso
- (Chandrasekaran et al., 2012) $\sigma_{\min,\mathcal{T}}(X)$ controls the performance of the recovery problem for noisy observations

Illustration for Random Observations

Proposition (R. et al., 2020a)
For $(x_1, \ldots, x_n) \stackrel{i.i.d.}{\sim} \mathcal{N}(0, \mathrm{I}_d)$, *in high probability, if* β^* *is k-sparse with*

$$k \lesssim \frac{n}{\log d}$$

then

$$\|\beta\|_1 - \|\beta^*\|_1 \geq (1 - \sqrt{k \log(d)/n}) \|\beta - \beta^*\|_1$$
$$\text{for all } \beta \in \mathbb{R}^d \text{ s.t. } \quad X\beta = X\beta^*,$$

- β^* *is the unique minimizer*
- *Number of total iterations of restarts to get* ε *accuracy is bounded by*

$$\mathcal{O}\left(\frac{\log \varepsilon^{-1}}{1 - \sqrt{k \log(d)/n}}\right)$$

Take-away

- More observations, fewer iterations

Optimization and Statistical Complexities

	Dense β^*	k-sparse β^*
Statistical Complexity (number of random observations needed to recover β^*)	d	$k \log d$
Optimization complexity (number of iterations to get an ε accuracy)	$\sqrt{\kappa} \log \varepsilon^{-1}$	$\kappa \log \varepsilon^{-1}$
Condition number κ	$\sigma_{\max}(X)^2 / \sigma_{\min}(X)^2$	$\sigma_{\max}(X) / \sigma_{\min,\mathcal{T}}(X)$

Take-away

1. Optimal convergence rates are obtained by exploiting **error bounds** using **restarts**
2. Error bounds can be derived from previous statistical analysis

Non-Linear Dynamical Problems from an Optimization Viewpoint

Vincent Roulet

University of Washington

February 26th, 2021

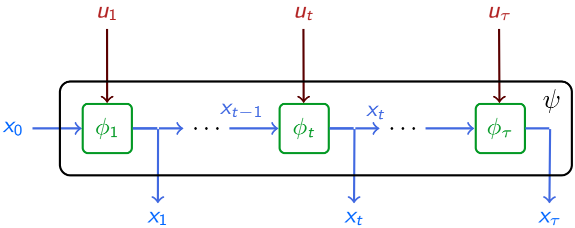
Non-Linear Dynamical Problems

Non-Linear Dynamics
We consider systems described by the following computations

$$x_0 = x \quad x_t = \phi_t(x_{t-1}, u_t) \quad \text{for } t = 1, \ldots, \tau$$

summarized as

$$\psi : (x, u_1, \ldots, u_\tau) \to (x_1, \ldots, x_\tau)$$

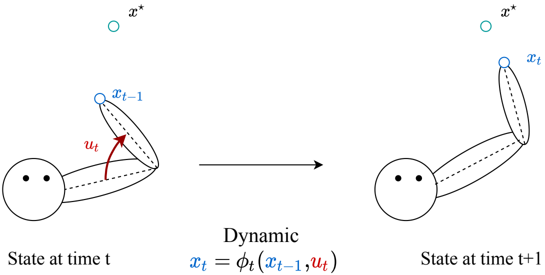


Non-Linear Control Problems

Control Example

$$x_0 = x \quad x_t = \phi_t(x_{t-1}, u_t) \quad \text{for } t = 1, \ldots, \tau$$

- x_t state of the system
- u_t control of the system (e.g. through a force)
- ϕ_t dynamics of the system known by Newton's law (often non-linear)
- τ length of the movement



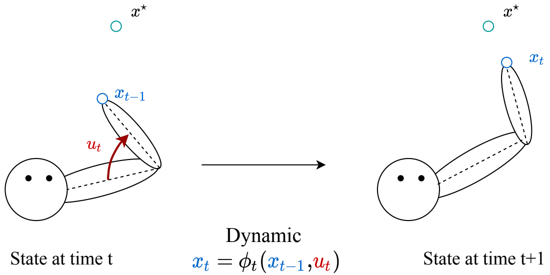
Non-Linear Control Problems

Control Objective

$$\min_{u_1,\ldots,u_\tau} \quad \|x_\tau - x^\star\|_2^2 + \sum_{t=1}^{\tau} \lambda \|u_t\|_2^2$$

$$\text{s.t} \quad x_0 = x \quad x_t = \phi_t(x_{t-1}, u_t) \quad \text{for } t = 1, \ldots, \tau$$

- x_t state of the system
- u_t control of the system (e.g. through a force)
- ϕ_t dynamics of the system known by Newton's law (often non-linear)
- τ length of the movement

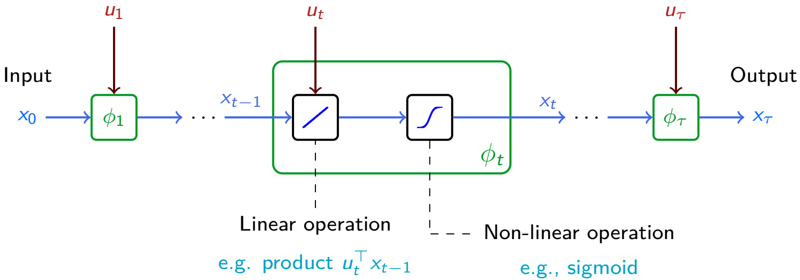


Non-Linear Prediction Models

Deep Network Structure

$$x_0 = x \quad x_t = \phi_t(x_{t-1}, u_t) \quad \text{for } t = 1, \dots, \tau$$

- x_0 input of the network
- u_t weights of the network at layer t
- ϕ_t t[th] layer of the netwrok
- τ depth of the network



Non-Linear Prediction Models

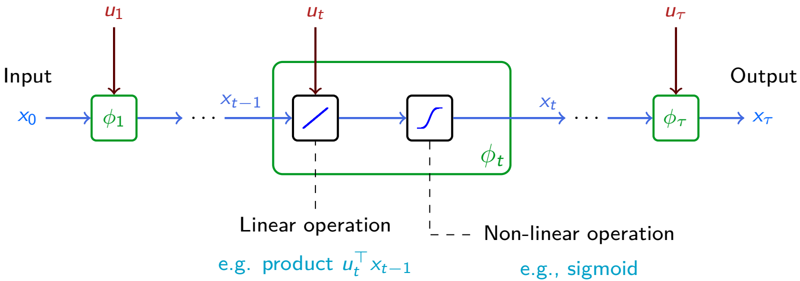
Deep Learning

n pair of inputs outputs examples $(x^{(i)}, y^{(i)})$, loss ℓ, regularization g_t

$$\min_{u_1,\ldots,u_\tau} \quad \frac{1}{n}\sum_{i=1}^{n} \ell(x_\tau^{(i)}, y^{(i)}) + \sum_{t=1}^{\tau} g_t(u_t)$$

$$\text{s.t.} \quad x_0^{(i)} = x^{(i)} \quad x_t^{(i)} = \phi_t(x_{t-1}^{(i)}, u_t) \quad \text{for } t = 1, \ldots, \tau$$

- x_0 input of the network
- u_t weights of the network at layer t
- ϕ_t t$^{\text{th}}$ layer of the netwrok
- τ depth of the network



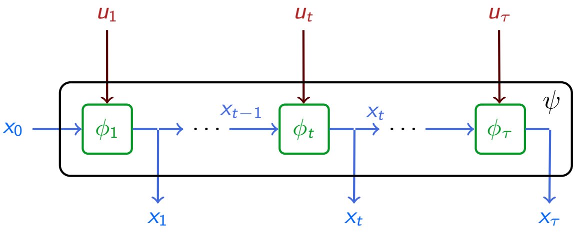
Non-Linear Dynamical Problems

Generic problem

Given τ computations ϕ_t define $\psi(x_0, u) = (x_1; \ldots; x_\tau)$ as below, generic problems read

$$\min_u h(\psi(x_0, u)) + g(u)$$

- $h(x) = \sum_{t=1}^{\tau} h_t(x_t)$ with $x = (x_1; \ldots; x_\tau)$
- $g(u) = \sum_{t=1}^{\tau} g_t(u_t)$ with $u = (u_1; \ldots; u_\tau)$
- $\psi(x_0, u)$ is **a chain of computations**



Non-Linear Dynamical Problems

Motivation

- In practice, classical non-linear control algo. are extremely efficient
- How can we explain this phenomenon from an optimization viewpoint?

Questions from an optimization viewpoint

How does the structure of the chain of computations impact

- the computational complexity of classical optimization methods?

$\rightarrow$ can we use e.g. Newton/Gauss-Newton methods that may be faster?

- the smoothness properties of the problem?

$\rightarrow$ how many time steps τ are reasonable to get fast convergence?

- the convergence of classical optimization methods?

$\rightarrow$ can we prove global convergence under suitable assumptions?

Cost of one Step of Classical Optimization Methods

Analysis
Each step can be defined as a subproblem

1. Decompose the sub-problem into the chain of computations
2. Get an efficient implementation of the subproblem

Lemma (R. et al., 2019 [2])

Gradient, Gauss-Newton or Newton steps amount can be solved by dynamic programming at a linear cost w.r.t. to the length τ.

Take-aways:

- Naive Gauss-Newton and Newton implementations would require $O(\tau^3)$
- For e.g. deep learning, Gauss-Newton steps can also be computed by automatic-differentiation, see (R. et al., 2019)
- Compared to classical methods (ILQR, ILEQG (Li and Todorov, 2004; Whittle, 1990)) our analysis reveals that they are missing a regularization term, see (R. et al., 2019, 2020b)

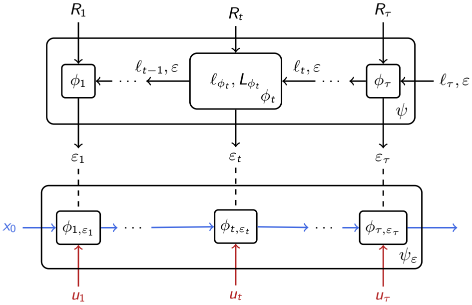
[2] See also (Dunn and Bertsekas, 1989) , (Sideris and Bobrow, 2005)

Smoothness Properties

Automatic smoothness estimates (R. and Harchaoui, 2019)
Given the smoothness properties of the computations ϕ_t defining ψ, we developed an automatic procedure to provide estimates of (i) a bound, (ii) the Lip. cont., (iii) the smoothness of ψ on any bounded sets
Example for ϕ_t, ℓ_ϕ Lip. continous, L_ϕ smooth,

$$\ell_\psi = \frac{\ell_\phi - \ell_\phi^{\tau+1}}{1-\ell_\phi} \qquad L_\psi = \frac{L_\phi\left(1-(1+2\tau)(1-\ell_\phi)\ell_\phi^\tau - \ell_\phi^{2\tau+1}\right)}{(1-\ell_\phi)^3}$$

Automatic smoothing (R. and Harchaoui, 2021)
Given a chain of non-smooth but smoothable computations ϕ_t defining ψ, we developed an automatic procedure to build a ε- smooth approximation of ψ



Moreau Gradients

Instead of computing $\nabla\psi$, consider computing

$$\nabla \operatorname{env}(\lambda^\top \psi)(x) = \arg\min_y \lambda^\top \psi(x+y) + \frac{1}{2}\|y\|_2^2$$

Intuition:

- If ψ is linear, we retrieve a gradient
- Generally we get an **implicit gradient**

Why?

- The error of approximation by Moreau gradients is controlled by **an optimization method**
- Can circumvent the vanishing/exploding smoothness issues

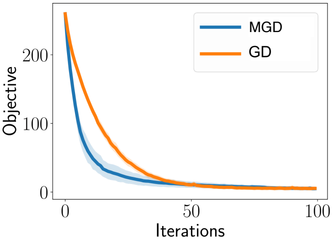
How?

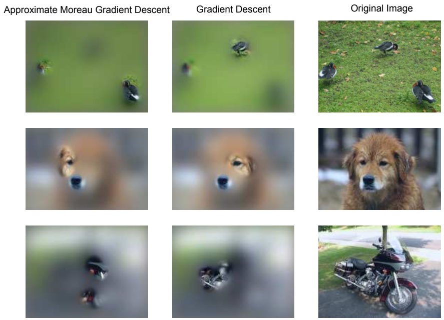
We proposed to approx. this oracle by back-propagating Moreau gradients

- $\nabla_x \phi_t(x_{t-1}, u_t)\lambda_t$ becomes $\nabla \operatorname{env}(\lambda^\top \phi_t(\cdot, u_t))(x_{t-1})$
- $\nabla_u \phi_t(x_{t-1}, u_t)\lambda_t$ becomes $\nabla \operatorname{env}(\lambda^\top \phi_t(x_{t-1}, \cdot))(u_t)$

Differentiable Programming à La Moreau: Application

Inverting a Deep Network (Fong et al., 2019)
Given an image, and a trained deep network,
find the part of the image responsible for its label





Conclusion and Future Directions

Non-linear dynamical problems from an optimization viewpoint
How does the structure of the chain of computations impact

- the computational complexity of classical optimization methods ✓
- the smoothness properties of the problem? ✓
- the convergence of classical optimization methods? **?**

Error bounds for non-linear dynamical problems

- For simple systems where we have control on every direction of the acceleration ✓
- More generally, for non-linear control problems in continuous time, feasibility of a movement has been studied in continuous time as the **controllability** of the system

→ could be translated into properties of the discretized problem

- This could open the path for non-convex statistical models with convergence guarantees of e.g. a gradient descent

Thanks!

Becker, S., Bobin, J., and Candès, E. J. (2011). Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39.

Bickel, P. J., Ritov, Y., Tsybakov, A. B., et al. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of statistics*, 37(4):1705–1732.

Bolte, J., Nguyen, T. P., Peypouquet, J., and Suter, B. W. (2017). From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507.

Candes, E. and Tao, T. (2007). The dantzig selector: Statistical estimation when p is much larger than n. *Annals of statistics*, 35(6):2313–2351.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849.

Cohen, A., Dahmen, W., and DeVore, R. (2009). Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231.

Dunn, J. C. and Bertsekas, D. P. (1989). Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems. *Journal of Optimization Theory and Applications*, 63(1):23–38.

Fong, R., Patrick, M., and Vedaldi, A. (2019). Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2950–2958.

Freund, R. M. and Vera, J. R. (1999). Some characterizations and properties of the "distance to ill-posedness" and the condition measure of a conic linear system. *Mathematical Programming*, 86(2):225–260.

Li, W. and Todorov, E. (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. In *1st International Conference on Informatics in Control, Automation and Robotics*, volume 1, pages 222–229.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152.

R., V., Boumal, N., and d'Aspremont, A. (2020a). Computational complexity versus statistical performance on sparse recovery problems. *Information and Inference: A Journal of the IMA*, 9(1):1–32.

R., V. and d'Aspremont, A. (2020). Sharpness, restart, and acceleration. *SIAM Journal on Optimization*, 30(1):262–289.

R., V., Fazel, M., Srinivasa, S., and Harchaoui, Z. (2020b). On the convergence of the iterative linear exponential quadratic gaussian algorithm to stationary points. In *2020 American Control Conference*, pages 132–137. IEEE.

R., V. and Harchaoui, Z. (2019). An elementary approach to convergence guarantees of optimization algorithms for deep networks. In *Proceedings of the 57th Annual Allerton Conference on Communication, Control, and Computing*, pages 84–91. IEEE.

R., V. and Harchaoui, Z. (2021). On the smoothing of deep networks. In *Proceedings of the 55th Annual Conference on Information Sciences and Systems*. To appear.

R., V., Srinivasa, S., Drusvyatskiy, D., and Harchaoui, Z. (2019). Iterative linearized control: Stable algorithms and complexity guarantees. In *Proceedings of the 36th International Conference on Machine Learning*.

Sideris, A. and Bobrow, J. E. (2005). An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems. In *Proceedings of the American Control Conference*, pages 2275–2280.

Whittle, P. (1990). *Risk-sensitive optimal control*. Wiley-Interscience series in systems and optimization. Wiley.