Nonsmooth calculus and optimization in machine learning: first-order sampling and implicit differentiation

PhD defense

Tam Le

advised by Jérôme Bolte (TSE) and Edouard Pauwels (TSE),
joint work with Antonio Silveti-Falls (CentraleSupelec)

Nonsmooth optimization in machine learning

Many machine learning problems cast into an optimization problem.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w})$$

First-order methods, e.g. gradient method are really popular

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k)$$

Nonsmooth optimization in machine learning

Many machine learning problems cast into an optimization problem.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w})$$

First-order methods, e.g. gradient method are really popular

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k)$$

- Automatic differentiation libraries (Pytorch, Tensorflow, JAX)

Nonsmooth optimization in machine learning

Many machine learning problems cast into an optimization problem.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w})$$

First-order methods, e.g. gradient method are really popular

$$w_{k+1} = w_k - \alpha_k(\nabla F(w_k) + \epsilon_k)$$

- Automatic differentiation libraries (Pytorch, Tensorflow, JAX)
- Can be adapted to handle massive training sets (Stochastic algorithms)

Nonsmooth optimization in machine learning

Many machine learning problems cast into an optimization problem.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w})$$

First-order methods, e.g. gradient method are really popular

$$w_{k+1} = w_k - \alpha_k(\nabla F(w_k) + \epsilon_k)$$

- Automatic differentiation libraries (Pytorch, Tensorflow, JAX)
- Can be adapted to handle massive training sets (Stochastic algorithms)
- Computational power (GPU)

Nonsmooth optimization in machine learning

Many machine learning problems cast into an optimization problem.

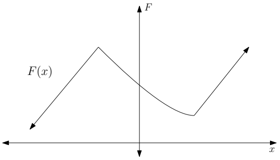
$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w})$$

First-order methods, e.g. gradient method are really popular

$$w_{k+1} = w_k - \alpha_k(\nabla F(w_k) + \epsilon_k)$$

- Automatic differentiation libraries (Pytorch, Tensorflow, JAX)
- Can be adapted to handle massive training sets (Stochastic algorithms)
- Computational power (GPU)

Observation: In many practical situations, F is **nonconvex and nonsmooth**.



Example 1: neural networks

Supervised learning.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad \mathbb{E}_{(x,y)\sim P}[d(h(\mathbf{w},x),y)]$$

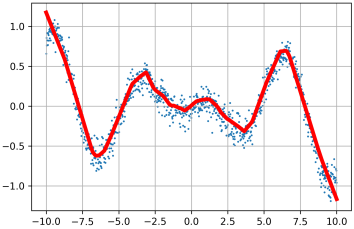
prediction task $h(w,x) \approx y$

Example 1: neural networks

Supervised learning.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad \mathbb{E}_{(x,y)\sim P}[d(h(\mathbf{w}, x), y)]$$

prediction task $h(w, x) \approx y$

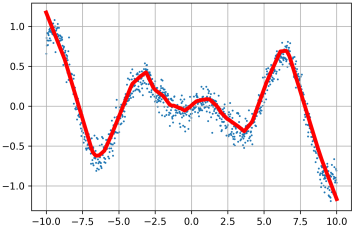


Example 1: neural networks

Supervised learning.

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad \mathbb{E}_{(x,y)\sim P}[d(h(\mathbf{w},x),y)]$$

prediction task $h(w,x) \approx y$



Neural networks. In deep learning, predictions are built upon multiple compositions.

$$h(w,x) = \sigma_L(A_L\sigma_{L-1}(A_{L-1}\ldots\sigma_2(A_2\sigma_1(A_1x+b_1)+b_2)+b_{L-1}\ldots)+b_L)$$

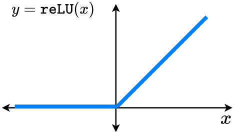
$w = (A_1, A_2 \ldots A_L, b_1, \ldots, b_L)$.

many applications: image classification, speech recognition, language prediction...

Example 1: neural networks

The σ_i are nonlinear and often **nonsmooth** functions:
ReLU function

$$\texttt{reLU}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$



nonsmooth at 0.

MaxPooling

$$\text{MaxPooling}\left(\begin{bmatrix} 3 & 1 & \mathbf{13} & 8 \\ 7 & \mathbf{20} & 6 & 2 \\ \mathbf{7} & 3 & \mathbf{5} & 4 \\ 6 & 2 & 1 & 5 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{20} & \mathbf{13} \\ \mathbf{7} & \mathbf{5} \end{bmatrix}.$$

max function is nonsmooth when components are equal.

Example 2: Bi-level optimization

$$\underset{w\in\mathbb{R}^p}{\text{Minimize}} \; F(w,z) \quad \text{s.t. } z \in \underset{\theta\in\mathcal{C}}{\operatorname{argmin}}\, g(w,\theta).$$

Studied beforehand in economics game theory (Stackelberg, 1952), optimization (Bracken and McGill (1973), Dempe, Ye...).

Renewed interest in machine learning:

Example 2: Bi-level optimization

$$\underset{w \in \mathbb{R}^p}{\text{Minimize}} \;\; F(w, z) \quad \text{s.t. } z \in \underset{\theta \in \mathcal{C}}{\operatorname{argmin}}\, g(w, \theta).$$

Studied beforehand in economics game theory (Stackelberg, 1952), optimization (Bracken and McGill (1973), Dempe, Ye...).

Renewed interest in machine learning:

- Optimization layers, Amos & Kolter 2017.

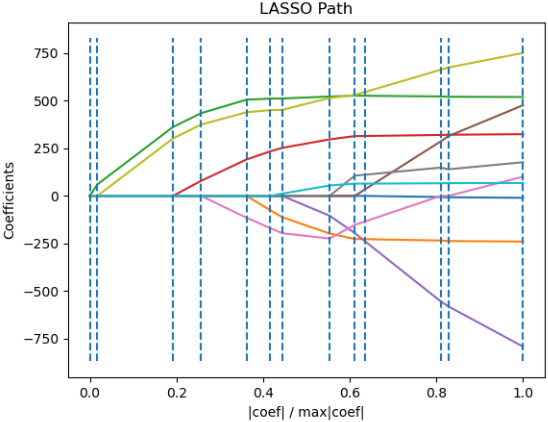
$$\xrightarrow{x} \boxed{\mathbf{z} \in \operatorname{argmin}_{\theta \in \mathcal{C}}\, g(w, x, \theta)} \xrightarrow{z}$$

- Hyperparameter optimization: Bengio 2000; Do et al. 2007; Bertrand et al. 2020
  Example of the lasso:

$$\underset{\lambda}{\text{Minimize}} \;\; \text{Criterion}(\beta(\lambda))$$
$$\text{s.t. } \beta(\lambda) \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}}\, \|X\beta - Y\|^2 + \lambda \|\beta\|_1 .$$

Solution paths are often nonsmooth

LASSO path $\beta(\cdot)$ is piecewise linear.



Outline

Observation: a gap between the classical theory in nonsmooth opt. vs practice in ML

- First-order methods, "gradient methods" are used on nonsmooth functions in practice, thanks to automatic differentiation libraries.
- The classical theory for nonsmooth functions (Clarke subgradient) doesn't explain this practice.

Outline

Observation: a gap between the classical theory in nonsmooth opt. vs practice in ML

- First-order methods, "gradient methods" are used on nonsmooth functions in practice, thanks to automatic differentiation libraries.
- The classical theory for nonsmooth functions (Clarke subgradient) doesn't explain this practice.

Solution: Nonsmooth calculus with Conservative derivatives

We focus on generalized derivatives called **Conservative derivatives**, which justifies automatic differentiation.

We propose **two extensions**

- Differentiation under nonsmooth expectation $\rightarrow$ stochastic methods.
- Nonsmooth Implicit differentiation $\rightarrow$ gradient methods for bi-level problems.

Outline

Observation: a gap between the classical theory in nonsmooth opt. vs practice in ML

- First-order methods, "gradient methods" are used on nonsmooth functions in practice, thanks to automatic differentiation libraries.
- The classical theory for nonsmooth functions (Clarke subgradient) doesn't explain this practice.

Solution: Nonsmooth calculus with Conservative derivatives

We focus on generalized derivatives called **Conservative derivatives**, which justifies automatic differentiation.

We propose **two extensions**

- Differentiation under nonsmooth expectation $\to$ stochastic methods.
- Nonsmooth Implicit differentiation $\to$ gradient methods for bi-level problems.

Output: Analysis of nonsmooth first-order algorithms.

Using **ODE approaches**, we show the convergence of nonsmooth stochastic optimization algorithms **as implemented in practice**.

Outline

1. Nonsmooth optimization: classical theory and practice in ML

2. Nonsmooth calculus with Conservative derivatives

3. Analysis of nonsmooth first-order algorithms

1. Nonsmooth optimization: classical theory and practice in ML

2. Nonsmooth calculus with Conservative derivatives

3. Analysis of nonsmooth first-order algorithms

A glance at the differentiable setting

Gradient method

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k).$$

- $-\nabla F$ generates **descent** trajectories, F decreases along the gradient curves

$$\dot{w} = -\nabla F(w).$$

  $\rightarrow$ Gradient method as ODE discretization.

- ∇F can be computed by **calculus** rules: $\nabla(f+g) = \nabla f + \nabla g$, $\mathrm{Jac}(u \circ v) = (\mathrm{Jac}\, u \circ v)\, \mathrm{Jac}\, v$ ...

What if F is nonsmooth?

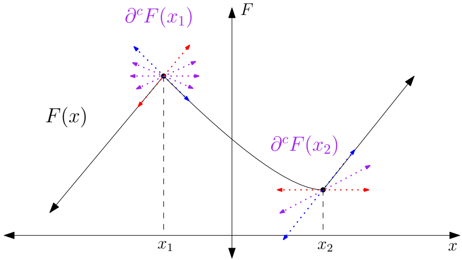
The Clarke subgradient: gradient for nonsmooth functions

Let $F : \mathbb{R}^n \to \mathbb{R}$ locally Lipschitz, differentiable on diff_F of full Lebesgue measure.

Clarke subgradient ∂^c

$$\partial^c F(x) = \mathrm{conv}\left\{ \lim_{k \to +\infty} \nabla F(x_k) : x_k \in \mathrm{diff}_F, x_k \underset{k \to +\infty}{\to} x \right\}$$

(extends to Jacobians)



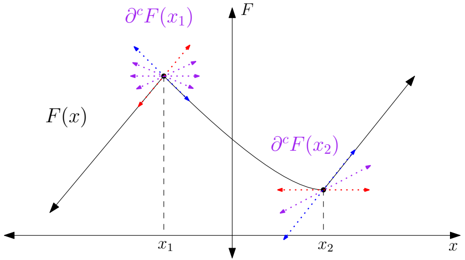
The Clarke subgradient: gradient for nonsmooth functions

Let $F : \mathbb{R}^n \to \mathbb{R}$ locally Lipschitz, differentiable on diff_F of full Lebesgue measure.

Clarke subgradient ∂^c

$$\partial^c F(x) = \mathrm{conv}\left\{ \lim_{k\to+\infty} \nabla F(x_k) : x_k \in \mathrm{diff}_F, x_k \underset{k\to+\infty}{\to} x \right\}$$

(extends to Jacobians)



$\partial^c F$ is graph-closed, locally bounded, convex-valued.
$\to$ existence of the continuous dynamic $\dot{x} \in -\partial^c F(x)$.
$\to$ subgradient method $w_{k+1} \in w_k - \alpha_k \partial^c F(w_k)$ as ODE discretization.

Path-differentiable functions

Do we have descent along $\dot{\gamma} \in -\partial^c F(\gamma)$?

Not in general. There exist Lipschitz functions F such that $\partial^c F = B(0,1)$ everywhere.

Path-differentiable functions

Do we have descent along $\dot{\gamma} \in -\partial^c F(\gamma)$?

Not in general. There exist Lipschitz functions F such that $\partial^c F = B(0, 1)$ everywhere.

Path differentiability, Valadier 1989

Let F locally Lipschitz. F is called path-differentiable if for all absolutely continuous curve $\gamma : [0, 1] \to \mathbb{R}^p$, for almost all $t \in [0, 1]$

$$\frac{\mathrm{d}}{\mathrm{d}t}(F \circ \gamma)(t) = \langle v, \dot{\gamma}(t) \rangle, \quad \forall v \in \partial^c F(\gamma(t)).$$

$\to$ enforces descent along the curves $\dot{\gamma} \in -\partial^c F(\gamma)$:

Path-differentiable functions

How “generic” is the class of path-differentiable functions?

Path-differentiable functions

How "generic" is the class of path-differentiable functions?

$\rightarrow$ **Path differentiable functions are ubiquitous in practice.**

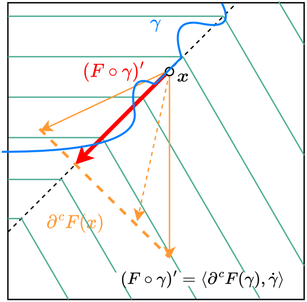
- Convex functions (Brézis, 1973)
- Semialgebraic ($\sim$ piecewise polynomial), **definable** functions, (Davis et al., 2019).
- **Definable functions**: functions written as compositions involving elementary functions $(\mathrm{if}, \mathrm{else}, +, \cdot, \times, \exp, \log)$

Path-differentiable functions

How "generic" is the class of path-differentiable functions?

$\rightarrow$ **Path differentiable functions are ubiquitous in practice.**

- Convex functions (Brézis, 1973)
- Semialgebraic ($\sim$ piecewise polynomial), **definable** functions, (Davis et al., 2019).
- **Definable functions**: functions written as compositions involving elementary functions $(\mathrm{if}, \mathrm{else}, +, \cdot, \times, \exp, \log)$

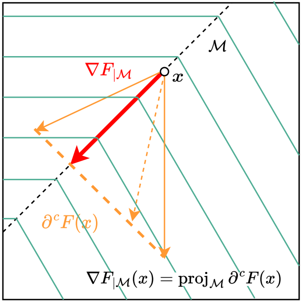


Path-differentiable functions

How "generic" is the class of path-differentiable functions?

$\rightarrow$ **Path differentiable functions are ubiquitous in practice.**

- Convex functions (Brézis, 1973)
- Semialgebraic ($\sim$ piecewise polynomial), **definable** functions, (Davis et al., 2019).
- **Definable functions**: functions written as compositions involving elementary functions $(\mathrm{if}, \mathrm{else}, +, \cdot, \times, \exp, \log)$



Projection formula, Bolte et al. 2007

Concerns

- The Clarke subgradient provides descent for path-differentiable functions.

Concerns

- The Clarke subgradient provides descent for path-differentiable functions.

  $\rightarrow$ Can we easily compute elements of the Clarke subgradient? Does **automatic differentiation** output **Clarke subgradients**?

Concerns

- The Clarke subgradient provides descent for path-differentiable functions.

  $\rightarrow$ Can we easily compute elements of the Clarke subgradient? Does **automatic differentiation** output **Clarke subgradients**?

- Definable functions: functions implemented in practice are path-differentiable.

Concerns

- The Clarke subgradient provides descent for path-differentiable functions.

  $\rightarrow$ Can we easily compute elements of the Clarke subgradient? Does **automatic differentiation** output **Clarke subgradients**?

- Definable functions: functions implemented in practice are path-differentiable.

  $\rightarrow$ What can we say about **expectation** minimization $F(w) = \mathbb{E}_{\xi \sim P}[f(w, \xi)]$?

What is automatic differentiation?

Auto-differentiation libraries (PyTorch, Tensorflow) differentiate programs

$$\mathrm{relu}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases} \quad \underset{\text{autodiff}}{\longrightarrow} \quad \mathrm{relu}'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases}$$

What is automatic differentiation?

Auto-differentiation libraries (PyTorch, Tensorflow) differentiate programs

$$\mathrm{relu}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases} \quad \xrightarrow[\text{autodiff}]{} \quad \mathrm{relu}'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases}$$

How about compositions, e.g. neural nets?

What is automatic differentiation?

Auto-differentiation libraries (PyTorch, Tensorflow) differentiate programs

$$\mathrm{relu}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases} \quad \xrightarrow[\text{autodiff}]{} \quad \mathrm{relu}'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases}$$

How about compositions, e.g. neural nets?

Automatic differentiation applies the chain rule on Clarke derivatives.

$$f(w) = g_r \circ g_{r-1} \circ \ldots \circ g_1(w)$$

$$\begin{aligned} \mathrm{autodiff}_w\, f(w) \in \partial^c g_r(g_{r-1} \circ \ldots \circ g_1(w))^T \\ \times \mathrm{Jac}^c\, g_{r-1}(g_{r-2} \circ \ldots \circ g_1(w)) \times \ldots \times \mathrm{Jac}^c_w\, g_1(w). \end{aligned}$$

What is automatic differentiation?

Auto-differentiation libraries (PyTorch, Tensorflow) differentiate programs

$$\mathrm{relu}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases} \quad \xrightarrow[\text{autodiff}]{} \quad \mathrm{relu}'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{else.} \end{cases}$$

How about compositions, e.g. neural nets?

Automatic differentiation applies the chain rule on Clarke derivatives.

$$f(w) = g_r \circ g_{r-1} \circ \ldots \circ g_1(w)$$

$$\begin{aligned} \mathrm{autodiff}_w\, f(w) \in\ & \partial^c g_r(g_{r-1} \circ \ldots \circ g_1(w))^T \\ & \times \mathrm{Jac}^c\, g_{r-1}(g_{r-2} \circ \ldots \circ g_1(w)) \times \ldots \times \mathrm{Jac}^c_w\, g_1(w). \end{aligned}$$

But do calculus rules apply to Clarke derivatives?

No.

- (Sum rule) $\partial^c(f+g) \subsetneq \partial^c f + \partial^c g$.
- (Composition rule) $\mathrm{Jac}^c(F \circ G) \subsetneq \mathrm{conv}\, \mathrm{Jac}^c F(G)\, \mathrm{Jac}^c G$,

Formal differentiation in machine learning.

Yet, autodiff is used extensively in machine learning:

Formal differentiation in machine learning.

Yet, autodiff is used extensively in machine learning:

Example 1. Stochastic methods. To minimize $F = \mathbb{E}_{\xi \sim P}[f(\cdot, \xi)]$ we may sample $\nabla_w f(\cdot, \xi)$, $\xi \sim P$ to have a noisy estimate of

$$\mathbb{E}_{\xi \sim P}[\nabla_w f(\cdot, \xi)] = \nabla F \qquad \text{(Differentiation under integral)}$$

$\rightarrow$ Practice: f is nonsmooth, $\mathrm{autodiff}_w\, f(w, \xi)$ is sampled instead of $\nabla_w f(w, \xi)$.

Formal differentiation in machine learning.

Yet, autodiff is used extensively in machine learning:

Example 1. Stochastic methods. To minimize $F = \mathbb{E}_{\xi\sim P}[f(\cdot,\xi)]$ we may sample $\nabla_w f(\cdot,\xi)$, $\xi \sim P$ to have a noisy estimate of

$$\mathbb{E}_{\xi\sim P}[\nabla_w f(\cdot,\xi)] = \nabla F \qquad \text{(Differentiation under integral)}$$

$\rightarrow$ Practice: f is nonsmooth, $\mathrm{autodiff}_w\, f(w,\xi)$ is sampled instead of $\nabla_w f(w,\xi)$.

Example 2. Implicit differentiation. $H(x,y) = 0$, H continuously differentiable. How to differentiate y w.r.t. x?

$$\frac{\partial y}{\partial x} = -\left[\frac{\partial H}{\partial y}\right]^{-1} \frac{\partial H}{\partial x} \qquad \text{(Implicit differentiation)}$$

$\rightarrow$ Practice: H nonsmooth (e.g. optimality conditions). ∂H is replaced by $\mathrm{autodiff}\, H$.

Conservative gradients, Bolte & Pauwels (2021)

Let $F : \mathbb{R}^n \to \mathbb{R}$ locally Lipschitz, and a set-valued map $D : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$,
D is a **conservative gradient** for F if

- It generates descent trajectories

For all absolutely continuous curve
$\gamma : [0, 1] \to \mathbb{R}^n$,

$$\frac{\mathrm{d}}{\mathrm{d}t}(F \circ \gamma)(t) = \langle v, \dot{\gamma}(t) \rangle, \quad \forall v \in D(\gamma(t)),$$

for almost all $t \in [0, 1]$.

(extends to Jacobians)

F decreases along $\dot{\gamma} \in -D(\gamma)$

Conservative gradients, Bolte & Pauwels (2021)

Let $F : \mathbb{R}^n \to \mathbb{R}$ locally Lipschitz, and a set-valued map $D : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$,
D is a **conservative gradient** for F if

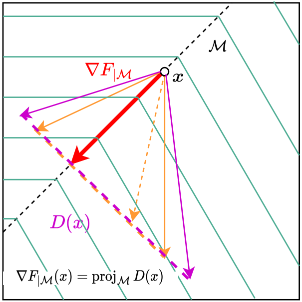
- It generates descent trajectories

For all absolutely continuous curve $\gamma : [0,1] \to \mathbb{R}^n$,

$$\frac{\mathrm{d}}{\mathrm{d}t}(F \circ \gamma)(t) = \langle v, \dot{\gamma}(t) \rangle, \quad \forall v \in D(\gamma(t)),$$

for almost all $t \in [0,1]$.

(extends to Jacobians)



F decreases along $\dot{\gamma} \in -D(\gamma)$

Conservative gradients, Bolte & Pauwels (2021)

Let $F : \mathbb{R}^n \to \mathbb{R}$ locally Lipschitz, and a set-valued map $D : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$,
D is a **conservative gradient** for F if

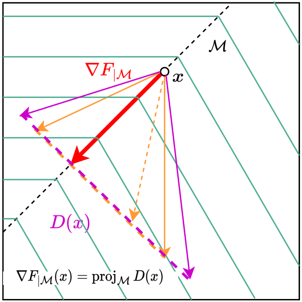
- It generates descent trajectories

For all absolutely continuous curve $\gamma : [0,1] \to \mathbb{R}^n$,

$$\frac{\mathrm{d}}{\mathrm{d}t}(F \circ \gamma)(t) = \langle v, \dot{\gamma}(t) \rangle, \quad \forall v \in D(\gamma(t)),$$

for almost all $t \in [0,1]$.

(extends to Jacobians)



F decreases along $\dot{\gamma} \in -D(\gamma)$

- Existence of the flow $\dot{\gamma} \in -D(\gamma)$:

D is graph-closed, nonempty (convex) valued, locally bounded.

Conservative calculus

- $\partial^c F$ is the minimal conservative gradient: if $\exists D_F$ conservative, then F is path differentiable.

Conservative calculus

- $\partial^c F$ is the minimal conservative gradient: if $\exists D_F$ conservative, then F is path differentiable.

- **Sum rule** Let D_f, D_g be conservative gradients for f and g, $D_f + D_g$ is conservative gradient for $f + g$.
- **Chain rule** Let J_u, J_v be conservative Jacobians for u and v. $J_u(v)J_v$ is conservative Jacobian for $u \circ v$. $\rightarrow$ automatic differentiation outputs conservative Jacobians.

Conservative calculus

- $\partial^c F$ is the minimal conservative gradient: if $\exists D_F$ conservative, then F is path differentiable.

- **Sum rule** Let D_f, D_g be conservative gradients for f and g, $D_f + D_g$ is conservative gradient for $f + g$.
- **Chain rule** Let J_u, J_v be conservative Jacobians for u and v. $J_u(v)J_v$ is conservative Jacobian for $u \circ v$. $\rightarrow$ automatic differentiation outputs conservative Jacobians.

Two extensions to conservative calculus.

1. **Implicit differentiation.** Given

$$F(x, y) = 0$$

where F is nonsmooth, how to "differentiate" y with respect to x?
$\rightarrow$ differentiating solutions path.

Conservative calculus

- $\partial^c F$ is the minimal conservative gradient: if $\exists D_F$ conservative, then F is path differentiable.

- **Sum rule** Let D_f, D_g be conservative gradients for f and g, $D_f + D_g$ is conservative gradient for $f + g$.
- **Chain rule** Let J_u, J_v be conservative Jacobians for u and v. $J_u(v)J_v$ is conservative Jacobian for $u \circ v$. $\rightarrow$ automatic differentiation outputs conservative Jacobians.

Two extensions to conservative calculus.

1. **Implicit differentiation.** Given

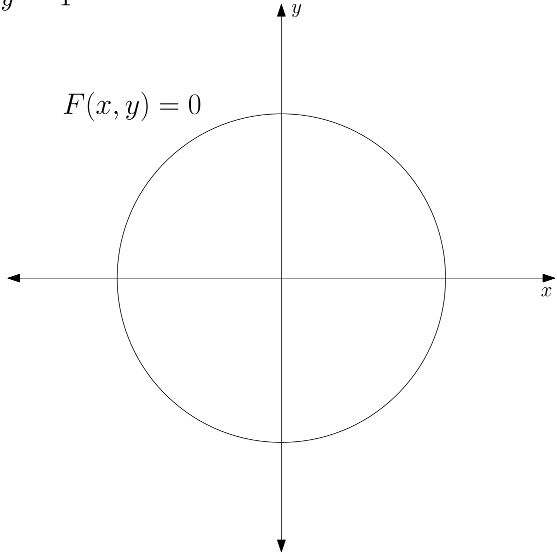
$$F(x, y) = 0$$

where F is nonsmooth, how to “differentiate” y with respect to x?
$\rightarrow$ differentiating solutions path.

2. **Integral rule.** “differentiating” under integral/expectation $F = \mathbb{E}_{\xi \sim P}[f(\cdot, \xi)]$
$\rightarrow$ nonsmooth stochastic methods.

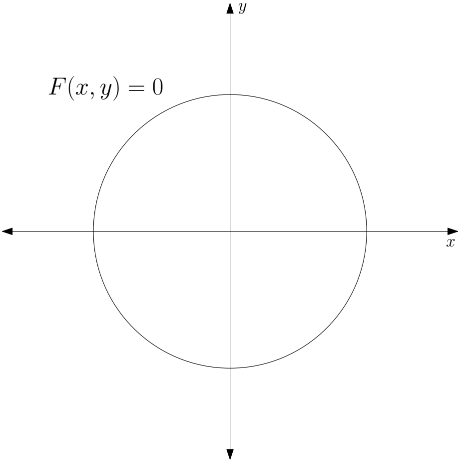
1. The Implicit Function Theorem (Cauchy)

$F(x, y) = x^2 + y^2 - 1$



1. The Implicit Function Theorem (Cauchy)

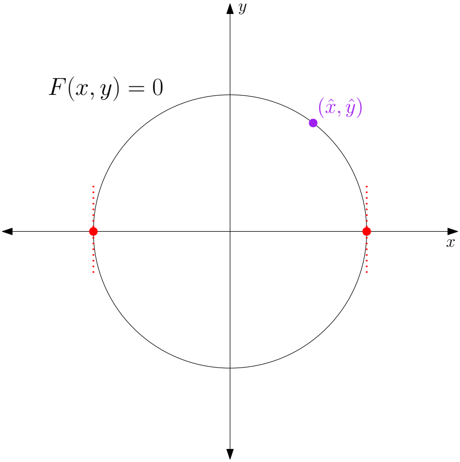
Existence and regularity.



1. The Implicit Function Theorem (Cauchy)

Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ continuously differentiable and $(\hat{x}, \hat{y})$ such that

$$F(\hat{x}, \hat{y}) = 0$$

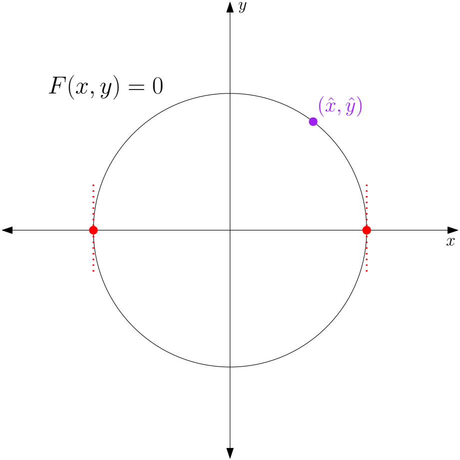


1. The Implicit Function Theorem (Cauchy)

Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ continuously differentiable and $(\hat{x}, \hat{y})$ such that

$$F(\hat{x}, \hat{y}) = 0$$

and $\frac{\partial F}{\partial y}(\hat{x}, \hat{y})$ is invertible



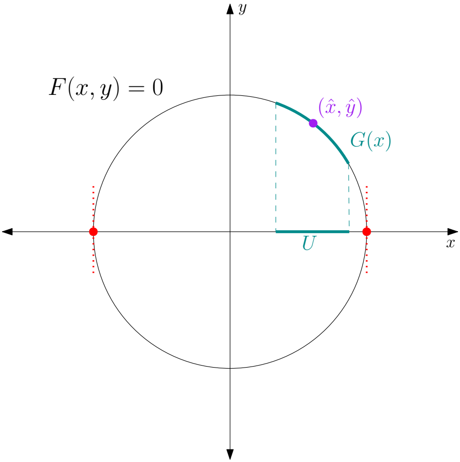
1. The Implicit Function Theorem (Cauchy)

Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ continuously differentiable and $(\hat{x}, \hat{y})$ such that

$$F(\hat{x}, \hat{y}) = 0$$

and $\frac{\partial F}{\partial y}(\hat{x}, \hat{y})$ is invertible
Then there exists a neighborhood U containing $\hat{x}$, and a unique continuously differentiable function $G : U \to \mathbb{R}^m$, verifying for all $x \in U$

$$F(x, G(x)) = 0.$$



1. The (smooth) Implicit Differentiation formula

$$F(\hat{x}, G(\hat{x})) = 0.$$

1. The (smooth) Implicit Differentiation formula

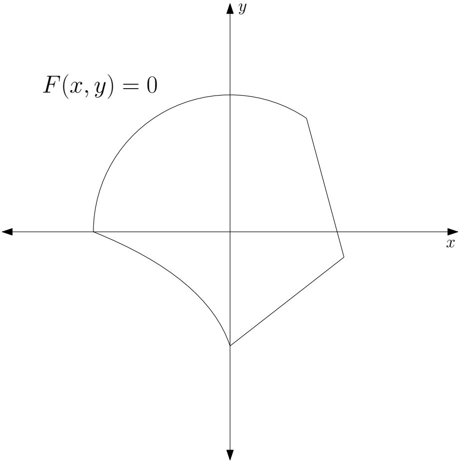
$$F(\hat{x}, G(\hat{x})) = 0.$$

Differentiating the equality on U leads to the **implicit differentiation** formula

$$\frac{\partial G}{\partial x}(\hat{x}) = -\left[\frac{\partial F}{\partial y}(\hat{x}, \hat{y})\right]^{-1} \frac{\partial F}{\partial x}(\hat{x}, \hat{y}).$$

1. Nonsmooth Implicit Function Theorem (Clarke, 1976)

Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be locally Lipschitz

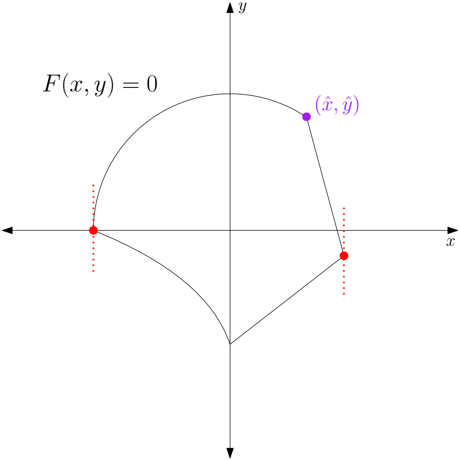


1. Nonsmooth Implicit Function Theorem (Clarke, 1976)

Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be locally Lipschitz and $(\hat{x}, \hat{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ such that

$$F(\hat{x}, \hat{y}) = 0.$$

If, $\forall [A\ B] \in \mathrm{Jac}^c F(\hat{x}, \hat{y})$ with $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times m}$, B is invertible,



1. Nonsmooth Implicit Function Theorem (Clarke, 1976)

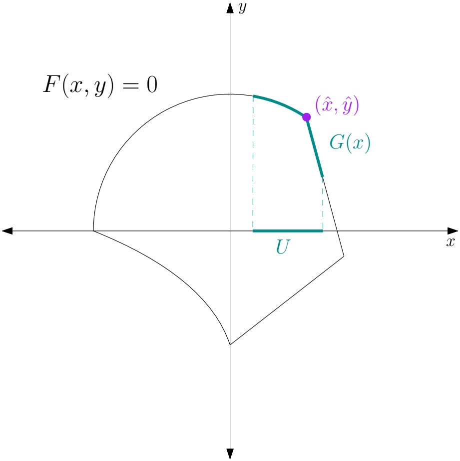
Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be locally Lipschitz and $(\hat{x}, \hat{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ such that

$$F(\hat{x}, \hat{y}) = 0.$$

If, $\forall [A\ B] \in \operatorname{Jac}^c F(\hat{x}, \hat{y})$ with $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times m}$, B is invertible, then $\exists U \subset \mathbb{R}^n$ a neighborhood of $\hat{x}$ and a locally Lipschitz function $G(x)$ so that

$$F(x, G(x)) = 0 \qquad \forall x \in U,$$

and $y = G(x)$ is the unique such solution in a neighborhood of $\hat{y}$.



1. Nonsmooth Implicit Function Theorem (Clarke, 1976)

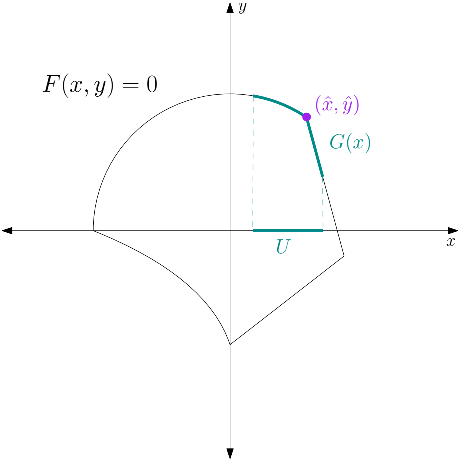
Existence and regularity.
Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be locally Lipschitz and $(\hat{x}, \hat{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ such that

$$F(\hat{x}, \hat{y}) = 0.$$

If, $\forall [A\ B] \in \mathrm{Jac}^c F(\hat{x}, \hat{y})$ with $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{m \times m}$, B is invertible, then $\exists U \subset \mathbb{R}^n$ a neighborhood of $\hat{x}$ and a locally Lipschitz function $G(x)$ so that

$$F(x, G(x)) = 0 \qquad \forall x \in U,$$

and $y = G(x)$ is the unique such solution in a neighborhood of $\hat{y}$.



1. Nonsmooth implicit differentiation

Nonsmooth implicit differentiation, Bolte, L., Pauwels, Silveti-Falls (2021)

Assume inplace that F is path-differentiable (semi-alg.). Then G is path-differentiable and

$$J_G(x) = \{-B^{-1}A \mid [A\ B] \in \mathrm{Jac}^c\, F(x, G(x))\}$$

is a **conservative Jacobian** for G.

Illustrative example: lasso hyperparameter optimization

Lasso hyperparameter tuning

$$\begin{aligned} &\underset{\lambda}{\text{Minimize}}\ C(\beta(\lambda)) \\ &\text{s.t. } \beta(\lambda) \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|X\beta - Y\|^2 + e^{\lambda}\|\beta\|_1. \end{aligned}$$

Let's apply nonsmooth implicit differentiation to $\beta(\lambda)$:

1. Implicit relation between β and λ = optimality condition:

$$F(\beta, \lambda) = \beta - \mathrm{prox}_{e^{\lambda}\|\cdot\|_1}(\beta - e^{\lambda} X^T (X\beta - Y)) = 0$$

Illustrative example: lasso hyperparameter optimization

Lasso hyperparameter tuning

$$\underset{\lambda}{\text{Minimize}}\ C(\beta(\lambda))$$
$$\text{s.t. } \beta(\lambda) \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \|X\beta - Y\|^2 + e^{\lambda}\|\beta\|_1 .$$

Let's apply nonsmooth implicit differentiation to $\beta(\lambda)$:

1. Implicit relation between β and λ = optimality condition:

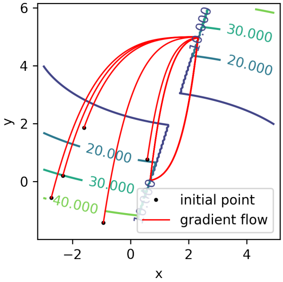
$$F(\beta, \lambda) = \beta - \operatorname{prox}_{e^{\lambda}\|\cdot\|_1}(\beta - e^{\lambda} X^T (X\beta - Y)) = 0$$

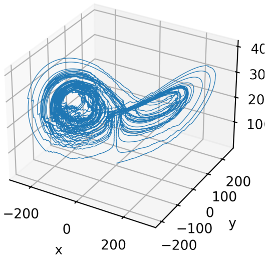
2. Uniqueness/invertibility assumption: (Osborne et al. 2000; Mairal, Yu 2012)

Define the *equicorrelation set* $\mathcal{E} := \{j \in \{1, \ldots, p\} : |X_j^{\mathsf{T}}(y - X\hat{\beta}(\lambda))| = e^{\lambda}\}$ and assume $X_{\mathcal{E}}^T X_{\mathcal{E}}$ has full rank

Then: nonsmooth implicit differentiation applies to F.

Invertibility condition is essential to satisfy: otherwise it can give really pathological training dynamics!





Other applications

- Differentiating through cone programs
- Implicit layers, Deep equilibrium networks

$$\mathbf{z} = \sigma(W\mathbf{z} + b + Ux)$$

2. Integral rule

Stochastic minimization

Consider

$$F(w) := \mathbb{E}_{\xi \sim P}[f(\cdot, \xi)]$$

Under mild conditions, one can differentiate under $\mathbb{E}$:

$$\nabla F = \mathbb{E}_{\xi \sim P}[\nabla_w f(\cdot, \xi)]$$

First-order sampling: Sample $\xi \sim P$, $\nabla_w f(w, \xi) \approx \nabla F(w)$
$\rightarrow$ Stochastic gradient method: $w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, \xi_k)$

2. Integral rule

Stochastic minimization

Consider

$$F(w) := \mathbb{E}_{\xi \sim P}[f(\cdot, \xi)]$$

Under mild conditions, one can differentiate under $\mathbb{E}$:

$$\nabla F = \mathbb{E}_{\xi \sim P}[\nabla_w f(\cdot, \xi)]$$

First-order sampling: Sample $\xi \sim P$, $\nabla_w f(w, \xi) \approx \nabla F(w)$
$\to$ Stochastic gradient method: $w_{k+1} = w_k - \alpha_k \nabla_w f(w_k, \xi_k)$

In practice, $f(\cdot, \xi)$ is nonsmooth, and

$$\partial^c F \subsetneq \mathbb{E}_{\xi \sim P}[\partial^c_w f(\cdot, \xi)]$$

But we have access to a **conservative gradient** of $f(\cdot, \xi)$, $D(\cdot, \xi)$, e.g., **autodiff**.

Question: What is the expectation $\mathbb{E}_{\xi \sim P}[D(\cdot, \xi)]$?

2. Nonsmooth Integral rule

Theorem (Bolte, L., Pauwels 2022)

If $D(\cdot, \xi)$ is conservative gradient for $f(\cdot, \xi)$, then $\mathbb{E}_{\xi \sim P}[D(\cdot, \xi)]$ is a conservative gradient for F.

2. Nonsmooth Integral rule

Theorem (Bolte, L., Pauwels 2022)

If $D(\cdot,\xi)$ is conservative gradient for $f(\cdot,\xi)$, then $\mathbb{E}_{\xi\sim P}[D(\cdot,\xi)]$ is a conservative gradient for F.

Assumptions:
1. (Measurability assumptions) ...
2. (Boundedness assumption) For all compact subset $C \subset \mathbb{R}^p$, there exists an integrable function $\kappa : S \to \mathbb{R}_+$ such that for all

$$(x,s) \in C \times S,\ \|D(x,s)\| \leq \kappa(s)$$

where for $(x,s) \in \mathbb{R}^p \times S$, $\|D(x,s)\| := \sup_{y\in D(x,s)} \|y\|$.

Main outcomes:

- **Justifies first-order sampling** in practical implementations: e.g. autodiff or implicit differentiation.
- F **is path-differentiable**, under simple assumptions.

1. Nonsmooth optimization: classical theory and practice in ML

2. Nonsmooth calculus with Conservative derivatives

3. Analysis of nonsmooth first-order algorithms

Nonsmooth Stochastic Gradient method in practical implementations

We consider the problem

$$\underset{\mathbf{w}\in\mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w}) := \mathbb{E}_{\xi\sim P}[f(\mathbf{w},\xi)],$$

We study a nonsmooth stochastic gradient method

$$w_{k+1} \in w_k - \alpha_k D(w_k, \xi_k). \tag{1}$$

For $\xi \in \mathbb{R}^m$, $D(\cdot,\xi)$ is a conservative gradient for $f(\cdot,\xi)$ $\rightarrow$ encompasses practical calculus: autodiff, implicit differentiation in the nonsmooth setting

Nonsmooth Stochastic Gradient method in practical implementations

We consider the problem

$$\underset{\mathbf{w} \in \mathbb{R}^p}{\text{Minimize}} \quad F(\mathbf{w}) := \mathbb{E}_{\xi \sim P}[f(\mathbf{w}, \xi)],$$

We study a nonsmooth stochastic gradient method

$$w_{k+1} \in w_k - \alpha_k D(w_k, \xi_k). \tag{1}$$

For $\xi \in \mathbb{R}^m$, $D(\cdot, \xi)$ is a conservative gradient for $f(\cdot, \xi) \to$ encompasses practical calculus: autodiff, implicit differentiation in the nonsmooth setting

Integral rule: (1) writes

$$w_{k+1} \in w_k - \alpha_k (D_F(w_k) + \epsilon_k),$$

where $D_F = \mathbb{E}_{\xi \sim P}[D(\cdot, \xi)]$ is conservative gradient for F, ϵ_k has zero conditional mean w.r.t. w_k.

The ODE approach

Studying algorithms as ODE discretizations:

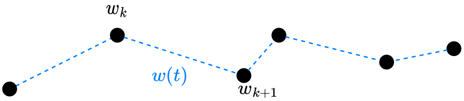
$$\frac{w_{k+1} - w_k}{\alpha_k} = -D_F(w_k) + \epsilon_k \qquad \rightsquigarrow \qquad \dot{\gamma} \in -D_F(\gamma) \tag{2}$$

The ODE approach

Studying algorithms as ODE discretizations:

$$\frac{w_{k+1} - w_k}{\alpha_k} = -D_F(w_k) + \epsilon_k \qquad \rightsquigarrow \qquad \dot{\gamma} \in -D_F(\gamma) \tag{2}$$

Interpolated process w:



Asymptotic pseudo trajectory Benaim (1999), Benaim-Hofbauer-Sorin (2005)

$w : \mathbb{R}_+ \to \mathbb{R}^p$ is an asymptotic pseudo trajectory (APT) if for all $T > 0$,

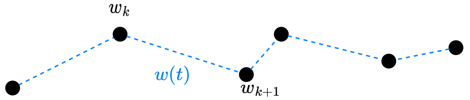
$$\lim_{t \to \infty} \inf_{\gamma \text{ solution}} \sup_{s \in [0,T]} \|w(t+s) - \gamma(s)\| = 0.$$

The ODE approach

Studying algorithms as ODE discretizations:

$$\frac{w_{k+1} - w_k}{\alpha_k} = -D_F(w_k) + \epsilon_k \qquad \rightsquigarrow \qquad \dot{\gamma} \in -D_F(\gamma) \tag{2}$$

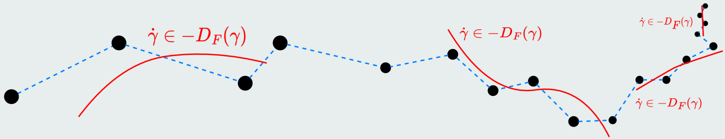
Interpolated process w:



Asymptotic pseudo trajectory Benaim (1999), Benaim-Hofbauer-Sorin (2005)

$w : \mathbb{R}_+ \to \mathbb{R}^p$ is an asymptotic pseudo trajectory (APT) if for all $T > 0$,

$$\lim_{t\to\infty} \inf_{\gamma \text{ solution}} \sup_{s\in[0,T]} \|w(t+s) - \gamma(s)\| = 0.$$



Convergence results

F decreases along $\dot{\gamma} \in -D_F(\gamma)$ (**conservative gradient**) $+$ w is APT
$=$ asymptotic descent.

Assumptions

- $(w_k)_{k \in \mathbb{N}}$ bounded a.s.
- $\alpha_k > 0$, $\sum \alpha_k = \infty$, $\sum \alpha_k^2 < \infty$
- $\|D(w,s)\| \le \kappa(s)\psi(w)$, κ square integrable, ψ locally bounded.

Convergence results

- Ermoliev, Norkin (1998), Benaim, Hofbauer, Sorin (2005): accumulation points w^* s.t. $\liminf_{k\to\infty} F(w_k) = F(w^*)$ satisfies $0 \in D_F(w^*)$
- Bianchi, Rios-Zertuche (2021): **essential** accumulation points w^* satisfy $0 \in D_F(w^*)$.

Convergence results

F decreases along $\dot{\gamma} \in -D_F(\gamma)$ (**conservative gradient**) + w is APT
= asymptotic descent.

Assumptions

- $(w_k)_{k\in\mathbb{N}}$ bounded a.s.
- $\alpha_k > 0$, $\sum \alpha_k = \infty$, $\sum \alpha_k^2 < \infty$
- $\|D(w,s)\| \leq \kappa(s)\psi(w)$, κ square integrable, ψ locally bounded.

Convergence results

- Ermoliev, Norkin (1998), Benaim, Hofbauer, Sorin (2005): accumulation points w^* s.t. $\liminf_{k\to\infty} F(w_k) = F(w^*)$ satisfies $0 \in D_F(w^*)$
- Bianchi, Rios-Zertuche (2021): **essential** accumulation points w^* satisfy $0 \in D_F(w^*)$.

Essential accumulation point w^* satisfies for all open $U \ni w^*$,

$$\limsup_{k\to\infty} \frac{\sum_{i=0}^k \alpha_i \mathbf{1}_{w_i \in U}}{\sum_{i=0}^k \alpha_i} > 0 \quad \text{a.s.}$$

Proportion of time spent around w^*

Sard condition

Furthermore, if

Sard condition

The set of critical values, $\{F(w) \ : \ 0 \in D_F(w)\}$ has empty interior.

Quite restrictive: holds if F and D_F are semialgebraic (definable):

- P finite support
- $P \ll$ Lebesgue with semialgebraic density.[1]

[1] Integration of constructible functions, Cluckers & Miller (2009)

Sard condition

Furthermore, if

Sard condition

The set of critical values, $\{F(w) \ : \ 0 \in D_F(w)\}$ has empty interior.

Quite restrictive: holds if F and D_F are semialgebraic (definable):

- P finite support
- $P \ll$ Lebesgue with semialgebraic density.[1]

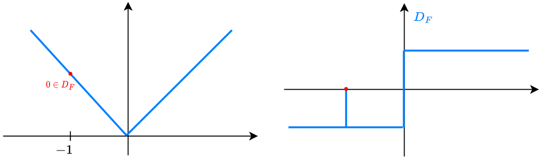
Then

- $F(w_k)$ converges as $k \to \infty$
- Every accumulation point w^* of $(w_k)_{k\in\mathbb{N}}$ satisfies $0 \in D_F(w^*)$.

[1] Integration of constructible functions, Cluckers & Miller (2009)

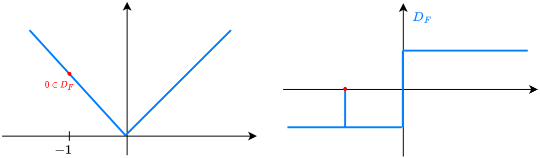
Clarke criticality for "most" sequences

Convergence to $\{0 \in D_F\}$ is unsatisfactory. We may have artificial critical points:



Clarke criticality for "most" sequences

Convergence to ${\color{red}\{0 \in D_F\}}$ is unsatisfactory. We may have artificial critical points:



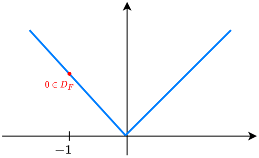
Theorem, Bolte & Pauwels (2020)

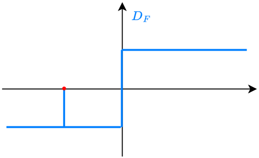
$D_F = \nabla F$ Lebesgue almost everywhere.

$\rightarrow$ **Suspicion (true): convergence to** ${\color{blue}\{0 \in \partial^c F\}}$ **often happens.**

Clarke criticality for "most" sequences

Convergence to $\{0 \in D_F\}$ is unsatisfactory. We may have artificial critical points:





Theorem, Bolte & Pauwels (2020)

$D_F = \nabla F$ Lebesgue almost everywhere.

$\rightarrow$ **Suspicion (true): convergence to** $\{0 \in \partial^c F\}$ **often happens.**

There exists $\Gamma \subset \mathbb{R}$, $W \subset \mathbb{R}^p$ "big" such that if $\{\alpha_k\}_{k \in \mathbb{N}} \subset \Gamma$, $w_0 \in W$,

$$w_{k+1} = w_k - \alpha_k(\nabla F(w_k) + \epsilon_k)$$

a.s., hence we have convergence to $\{0 \in \partial^c F\}$

Clarke criticality for "most" sequences

How big?
- **Bianchi, Hachem, Schechtman (2020)**: $s \sim P$, $f(\cdot, s)$ C^2 a.e. (e.g. semigalgebraic, definable)

Γ^c and W^c have zero Lebesgue measure,

Clarke criticality for "most" sequences

How big?
- **Bianchi, Hachem, Schechtman (2020)**: $s \sim P$, $f(\cdot, s)$ C^2 a.e. (e.g. semigalgebraic, definable)

Γ^c and W^c have zero Lebesgue measure,

Can we exploit the definability of f ?

- *Bolte and Pauwels (2020)*: f definable, P finitely discrete.

Γ^c is finite, W^c is a countable union of low dimensional manifolds.

- *Bolte, L., Pauwels (2022)*: $F(w) = \mathbb{E}_{\xi \sim P}[f(w, \xi)]$, $P \ll \mathrm{Lebesgue}$, f jointly definable.

Γ^c is finite, W^c is a countable union of low dimensional manifolds.

Versatility of the ODE approach: analysis of stochastic heavy ball

Nonsmooth stochastic heavy ball

$$
\begin{aligned}
w_{k+1} =& w_k - \alpha_k y_k \\
y_{k+1} \in& \beta_k D(w_{k+1}, \xi_{k+1}) + (1 - \beta_k) y_k.
\end{aligned}
$$

for all $k \in \mathbb{N}$, $\alpha_k > 0$ and $\beta_k \in (0, 1)$.

Related works:

- Smooth setting: Gadat et al. (2018)
- Nonsmooth: Ruszczyński (2020), Bianchi & Rios-Zertuche (2021)

Versatility of the ODE approach: analysis of stochastic heavy ball

Limiting dynamical system:

$$\dot{w} \in -ry$$
$$\dot{y} \in D_F(w) - y.$$

Lyapunov function:

$$E(w, y) = F(w) + \frac{r}{2}\|y\|^2$$

Stationary set:

$$\{0 \in D_F\} \times \{0\}$$

Artificial points avoidance:

If for a.e. s, $f(\cdot, s)$ is definable, then there exists $W \subset \mathbb{R}^p \times \mathbb{R}^p$ of full measure such that if $(w_0, w_1) \in W$, "accumulation points"[a] belong to $\{0 \in \partial^c F\} \times \{0\}$

[a] minimizing, essential, or all under Sard condition

Conclusion and perspectives

- Clarke subdifferential doesn't come with a calculus, and can't explain machine learning practice. **Conservative derivatives** provide a justification to many implementations

Conclusion and perspectives

- Clarke subdifferential doesn't come with a calculus, and can't explain machine learning practice. **Conservative derivatives** provide a justification to many implementations

- nonsmooth automatic differentiation,
- **Implicit differentiation** $\rightarrow$ bi-level programming, optimization layers, implicit layers
- **Differentiation under integral** $\rightarrow$ nonsmooth stochastic algorithms
- many other applications: value function, differentiation of ODE flows, monotone inclusion, iterative algorithms...

Conclusion and perspectives

- Clarke subdifferential doesn't come with a calculus, and can't explain machine learning practice. **Conservative derivatives** provide a justification to many implementations

- nonsmooth automatic differentiation,
- **Implicit differentiation** $\rightarrow$ bi-level programming, optimization layers, implicit layers
- **Differentiation under integral** $\rightarrow$ nonsmooth stochastic algorithms
- many other applications: value function, differentiation of ODE flows, monotone inclusion, iterative algorithms...

- Chain rule along curves $\rightarrow$ **ODE approach** $\rightarrow$ convergence results.

Conclusion and perspectives

- Clarke subdifferential doesn't come with a calculus, and can't explain machine learning practice. **Conservative derivatives** provide a justification to many implementations

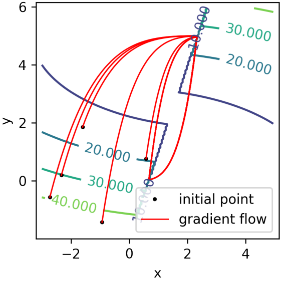
- nonsmooth automatic differentiation,
- **Implicit differentiation** $\rightarrow$ bi-level programming, optimization layers, implicit layers
- **Differentiation under integral** $\rightarrow$ nonsmooth stochastic algorithms
- many other applications: value function, differentiation of ODE flows, monotone inclusion, iterative algorithms...

- Chain rule along curves $\rightarrow$ **ODE approach** $\rightarrow$ convergence results.

- Convergence theory incomplete: Sard condition in stochastic optimization, complexity...
- Algorithmic extensions: constraints, biased oracle; beyond vanishing stepsizes: adaptive algorithms; not i.i.d. samples, ...
- Can we develop "nonsmooth-friendly" algorithms?

Thank you!

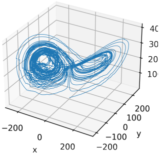
Pathological examples: cycles



$$\min_{x,y,s} \quad \ell(x,y,s) := (x-s_1)^2 + 4(y-s_2)^2$$

$$\text{s.t.} \quad s \in s(x,y) := \arg\max \{(a+b)(-3x+y+2) : \; a \in [0,3], b \in [0,5]\}.$$

Pathological examples: lorenz attractor



$$\max_{u \in \mathbb{R}^3} \quad u^\mathsf{T} z \qquad \text{s.t.} \qquad z \in \arg\min_{s \in \mathbb{R}^3} \|s - F(u)\|^4$$

F is Lorenz attractor vector field.

Optimality condition of the subproblem: $\|s - F(u)\|^3 (s - F(u)) = 0$.
Very qualitative explanation: The "gradient" of $u^T z$ is

$$z(u) + \widehat{\mathrm{Jac}}\, z(u) u$$

$\widehat{\mathrm{Jac}}$ is provided using the pseudo-inverse in the implicit differentiation formula, equal to zero. $z(u) = F(u)$. Finally, gradient ascent is approximately the Lorenz attractor.

Failure of Clarke calculus

Counterexample to a potential Clarke integral rule.
Let $f\colon (w,s) \mapsto s|w|$ and $P = \frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$ or P is the uniform density on $[-1,1]$. Then if $F(w) := \int_{[-1,1]} f(w,s)\,\mathrm{d}P(s)$, one has $\partial^c F(w) = 0$ for all $w \in \mathbb{R}$, but $\mathbb{E}_{\xi\sim P}[\partial^c_w f(0,\xi)] = [-1,1]$.

Counterexample to a potential "Clarke implicit differentiation"
Let $\Psi = \Phi^{-1}$

$$\Phi(x,y) = (|x| + y, 2x + |y|)$$

Implicit differentiation on $(u,v) \to u - \Psi(v)$ gives

$$[\mathrm{Jac}^c\,\Psi(0)]^{-1} \not\subset [\mathrm{Jac}^c\,\Phi(0)]$$

Norkin semismooth gradient

Let $F : \mathbb{R}^p \to \mathbb{R}$ locally Lipschitz, D_F is a semismooth generalized gradient if

$$F(y) = f(x) + \langle v, y - x \rangle + o(\|x - y\|) \text{ as } y \to x, \text{ for all } v \in D_F(y).$$

Asymptotic descent lemma (Ermoliev, Norkin 1998)

Let (w_k) generated by SGD, assume [...]. Let w^* be an accumulation point such that $0 \notin D_F(w^*)$, $(w_{i_k})_{k\in\mathbb{N}}$ a subsequence converging to w^*.

Then for any $\epsilon > 0$, there exists a subsequence $(w_{l_k})_{k\in\mathbb{N}}$ such that $\|w_k - w^*\| \leq \epsilon$ for all $k \in [i_k, l_k)$, and

$$\limsup_k F(w_{l_k}) < F(w^*)$$

Definition: definable in an o-minimal structure

Definition (o-minimal structure)

Let $\mathcal{O} = (\mathcal{O}_p)_{p \in \mathbb{N}}$ be a collection of sets such that, for all $p \in \mathbb{N}$, $\mathcal{O}_p$ is a set of subsets of $\mathbb{R}^p$. $\mathcal{O}$ is an o-minimal structure on $(\mathbb{R}, +, \cdot)$ if it satisfies the following axioms, for all $p \in \mathbb{N}$:

1. $\mathcal{O}_p$ is stable by finite intersection, union, and complementation, and contains $\mathbb{R}^p$.
2. If $A \in \mathcal{O}_p$ then $A \times \mathbb{R}$ and $\mathbb{R} \times A$ belong to $\mathcal{O}_{p+1}$.
3. If $A \in \mathcal{O}_{p+1}$ then $\pi(A) \in \mathcal{O}_p$, where π projects on the p first coordinates,.
4. $\mathcal{O}_p$ contains all sets of the form $\{x \in \mathbb{R}^p : P(x) = 0\}$, where P is a polynomial.
5. The elements of $\mathcal{O}_1$ are exactly the finite unions of intervals.