Smoothing convex functions for non-differentiable optimization

Federico Pierucci

Joint work with
Zaid Harchaoui, Jérôme Malick

Laboratoire Jean Kuntzmann - Inria

Séminaire BiPoP
Pinsot
November 17th, 2015

Outline

Problem to solve

"Doubly" non-differentiable optimization problem:

$$\min_{W \in \mathbb{R}^{d \times k}} \underbrace{R(W)}_{\text{non-differentiable loss}} + \underbrace{\lambda \|W\|}_{\text{non-differentiable regularization}}$$

The regularization is need to make "robust" the learning task

Motivations

$$\min_{W \in \mathbb{R}^{d \times k}} \|\mathcal{B}W\|_1 + \lambda \|W\|_{\sigma,1}$$

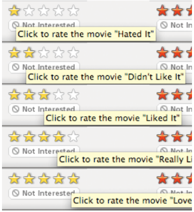
$$\min_{W \in \mathbb{R}^{d \times k}} \|\mathcal{B}W\|_\infty + \lambda \|W\|_{\sigma,1}$$

$\mathcal{B}$ Affine application that depend on data.

$\|W\|_{\sigma,1}$ 1) Nuclear norm, i.e. the sum of singular values of W
2) It is the convex hull of $\mathrm{rank}(W)$ when $\max_{ij}\{W_{ij}\} \leq 1$

Motivation 1

Collaborative filtering - Example: Netflix challenge



- **Data:** for user i and movie j
  $X_{ij} \in \{0, 0.5, \ldots, 4.5, 5\}$ ratings
  $\mathcal{I}$ set of indices of observations
- Characteristics of collaborative filtering:
  - **large scale**: size$(X) \sim 100\,000 \times 100\,000$
  - **sparse data**: size$(\mathcal{I}) < 0.1\%$
- The **aim** is to guess a future evaluation
  New $(i, j) \mapsto X_{ij} = ?$

$$\min_{W \in \mathbb{R}^{d \times k}} \quad \underbrace{\frac{1}{N} \sum_{(i,j) \in \mathcal{I}} |W_{ij} - X_{ij}|}_{R(W)} \quad + \quad \lambda \, \|W\|_{\sigma,1}$$

$X_{ij} \in \mathbb{R}$, with $(i, j) \in \mathcal{I}$: known rates (of movies)

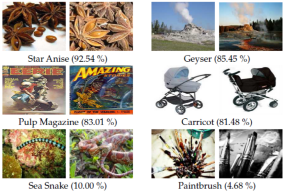
- $\|\cdot\|_{\sigma,1}$ regularization: enforces low rank solutions
- $|\cdot|$ loss: enforces robustness to outliers

Motivation 2

Multiclass classification - adaptation of SVM
(standard method in machine learning)

- **Data** $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}^k$: pairs of (picture, label)
  $W_j \in \mathbb{R}^d$: the j-th column of W
- The **aim** is to guess a future evaluation
  New picture $x \mapsto y = ?$

• Sample images from ImageNet with Top-1 accuracy for each class



$$\min_{W \in \mathbb{R}^{d \times k}} \quad \underbrace{\max\{0,\, 1 + \max_{r \text{ s.t. } r \neq y} \{\overbrace{W_r^T x - W_y^T x}^{\mathcal{A}W}\}\}}_{R(W) := H(\mathcal{A}W)} + \lambda \, \|W\|_{\sigma,1}$$

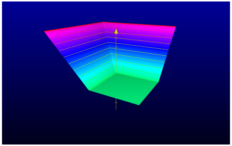
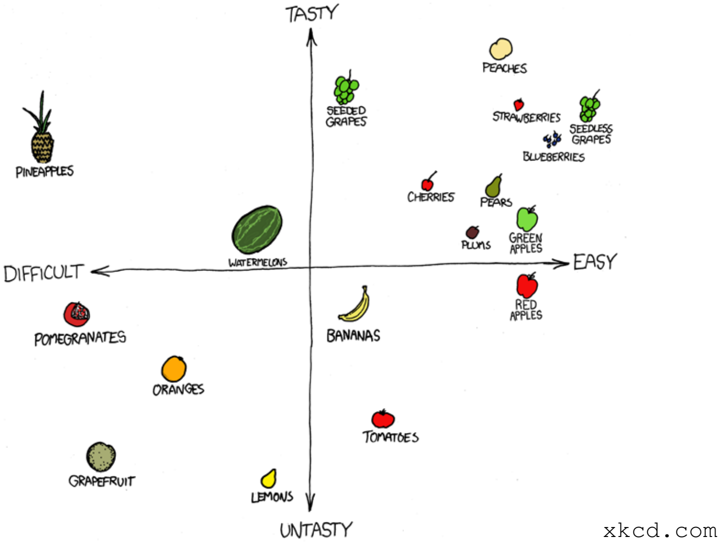


Figure: $H(\cdot)$

- R loss: minimizes the misclassification error
- $\|\cdot\|_{\sigma,1}$ regularization: enforces low rank models

Why nuclear-norm regularizer?

Classes are embedded in a low dimension subspace of the feature space.



Existing algorithms for nonsmooth optimization

$$\min_{W \in \mathbb{R}^{d \times k}} \underbrace{R(W)}_{\text{non-differentiable loss}} + \underbrace{\lambda \|W\|}_{\text{non-differentiable regularization}}$$

- General approach: Subgradient algorithms
- Special approaches:
  - reformultaions (e.g. QP, LP)
  - for special cases, Douglas-Rachford algorithm [Douglas, Rachford 1956]

Both algorithms are not scalable for double nonsmooth problems with $\|\cdot\|_{\sigma,1}$

Existing algorithms for nonsmooth optimization

$$\min_{W \in \mathbb{R}^{d\times k}} \underbrace{R(W)}_{\text{non-differentiable loss}} + \underbrace{\lambda \|W\|}_{\text{non-differentiable regularization}}$$

- General approach: Subgradient algorithms
- Special approaches:
  - reformultaions (e.g. QP, LP)
  - for special cases, Douglas-Rachford algorithm [Douglas, Rachford 1956]

Both algorithms are not scalable for double nonsmooth problems with $\|\cdot\|_{\sigma,1}$

What if the loss were smooth?

$$\min_{W \in \mathbb{R}^{d\times k}} \underbrace{\widetilde{R}(W)}_{\text{smooth loss}} + \underbrace{\lambda \|W\|}_{\text{nonsmooth regularization}}$$

Algorithms for smooth loss are "good" (by convergence)

- Proximal gradient algorithms. [Nemirovski, Yudin 1976] [Nesterov 2005] [Beck, Teboulle, 2009]
- Composite conditional gradient algorithm. Efficient iterations for $\|\cdot\|_{\sigma,1}$ [Harchaoui, Juditsky, Nemirovski, 2013]

Our approach

- The idea:
  combine existing algorithms with smoothing techniques
  “New algorithm = smoothing techniques + algorithm for smooth loss”
- This talk:
  Mainly about smoothing techniques
- In my thesis
  - Applications to machine learning problems
  - Real datasets: Imagenet, Movielens
  - “Optimal” smoothing

Outline

1. “Doubly” non-differentiable optimization problems
2. How to smooth a convex function?
3. Combining smoothing with algorithms
4. Conclusions and perspectives

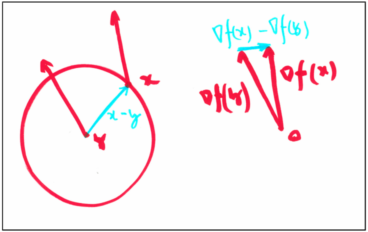
Definition (Smooth convex function)

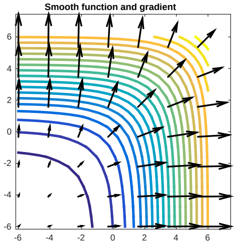
- The function f is differentiable on its domain
- The gradient ∇f is Lipschitz with modulus L, i.e

$$\text{for any } x, y \quad \|\nabla f(x) - \nabla f(y)\|_* \leq L \|x - y\|$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.

(Think about $\|\cdot\|$ = euclidean norm $= \|\cdot\|_*$)





Smoothing technique 1: convolution

We want to smooth g

$$g^c_\gamma(x) := \int_{\mathbb{R}^n} g(x - z)\mu_\gamma(z)dz$$

where μ_γ is a probability density function (concentration controlled by γ).

Smoothing technique 1: convolution

We want to smooth g

$$g_\gamma^c(x) := \int_{\mathbb{R}^n} g(x-z)\mu_\gamma(z)dz$$

where μ_γ is a probability density function (concentration controlled by γ).

Let μ_γ be the uniform distribution on a ball or normal distribution. Then a smooth surrogate g_γ has properties

- g_γ differentiable
- the gradient

$$\nabla g_\gamma^c(x) = \int_{\mathbb{R}^n} s(x-z)\mu_\gamma(z)dz, \quad \text{where } s(x-z) \in \partial g(x-z)$$

  is Lipschitz with modulus $L_\gamma = O(1/\gamma)$
- g_γ is uniform approximation of g, i.e. $\exists m$, $\exists M$ s.t.

$$g(x) - \gamma m \leq g_\gamma(x) \leq g(x) + \gamma M, \qquad \text{for all } x$$

  [Bertsekas 1978] [Duchi et al. 2012] [Pierucci et al. 2015]

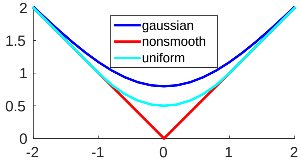
Numerical integration is difficult
Our objective is to obtain g_γ easy to evaluate numerically, possibly explicitly

Examples of explicit expressions in $\mathbb{R}$

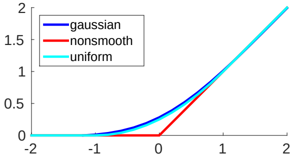
Uniform distribution: $\mu_\gamma(z) = \frac{1}{2\gamma} I_{[-1,1]}(\frac{z}{\gamma})$.

Gaussian distribution: $\mu_\gamma(z) = \frac{1}{\gamma\sqrt{2\pi}} \exp\left(-\frac{z^2}{2\gamma^2}\right)$, F: cumulative distribution

$g(\xi)$	μ	$g_r(\xi)$	$\nabla g_r(\xi)$
$\lvert\xi\rvert$	uniform	$\begin{cases} \frac{r}{2}(\frac{\xi}{r})^2 + \frac{1}{2} & \text{if } \lvert\xi\rvert \le r \\ \lvert\xi\rvert & \text{if } \lvert\xi\rvert > r \end{cases}$	$\begin{cases} \frac{\xi}{r} & \text{if } \lvert\xi\rvert \le r \\ \text{sign}(\xi) & \text{if } \lvert\xi\rvert > r \end{cases}$
$\lvert\xi\rvert$	gaussian	$-\xi F(-\frac{\xi}{r}) + \frac{\sqrt{2}}{\sqrt{\pi}} r e^{-\frac{\xi^2}{2r^2}} + \xi F(\frac{\xi}{r})$	$F(\frac{\xi}{r}) - F(-\frac{\xi}{r})$
$\max\{0,\xi\}$	uniform	$\begin{cases} 0 & \text{if } \xi \le -r \\ \frac{r}{4}(\frac{\xi}{r}+1)^2 & \text{if } -r < \xi < r \\ \xi & \text{if } r \ge \xi \end{cases}$	$\begin{cases} 0 & \text{if } \xi \le -r \\ \frac{\xi}{2r} + \frac{1}{2} & \text{if } -r < \xi < r \\ 1 & \text{if } r \ge \xi \end{cases}$
$\max\{0,\xi\}$	gaussian	$\frac{r}{\sqrt{2\pi}} e^{-\frac{1}{2r^2}\xi^2} + \xi F(\frac{\xi}{r})$	$F(\frac{\xi}{r})$



$g(\xi) = |\xi|$



$g(\xi) = \max\{0, \xi\}$

Examples of explicit expressions in $\mathbb{R}^n$

To **smooth** in $\mathbb{R}^n$ can be **complicate** (for easy numerical evaluation)
But for a decomposition

$$g(x) = \sum_{i=1}^{n} g^{(i)}(x_i), \qquad g^{(i)} \text{defined on } \mathbb{R}$$

we find a smooth $g_\gamma^{(i)}$ for each component and get

$$g_\gamma(x) = \sum_{i=1}^{n} g_\gamma^{(i)}(x_i)$$

Example: norm ℓ^1

- $g(x) = \|x\|_1 = \sum_{i=1}^{n} |x_i|$ to make smooth
- $\mu_\gamma(z) = \frac{1}{\gamma}\frac{1}{2^n} I_{B_\infty}\left(\frac{z}{\gamma}\right)$ uniform distribution on $B_\infty = \{\|\cdot\|_\infty \leq 1\}$
- $g_\gamma(x) = \sum_{i=1}^{k} \gamma H\left(\frac{x_i}{\gamma}\right)$, with $H(t) = \begin{cases} \frac{1}{2}t^2 + \frac{1}{2} & |t| \leq 1 \\ |t| & |t| > 1 \end{cases}$

Smoothing technique 2: infimal convolution

We want to smooth g

$$g_\gamma^{ic}(x) := \inf_{z \in \mathbb{R}^n} g(x - z) + \omega_\gamma(z)$$

where $\omega_\gamma(\cdot) = \gamma\omega\left(\frac{\cdot}{\gamma}\right)$ and ω is a smooth function.

Then a smooth surrogate g_γ has properties

- g_γ differentiable
- The gradient

$$\nabla g_\gamma(x) = \nabla \omega_\gamma(x - z_\mu^\star(x)), \qquad \text{with } z_\mu^\star(x) \text{ optimal in } g_\gamma^{ic}(x),$$

  is Lipschitz with modulus $L_\gamma = O(1/\gamma)$
- g_γ is uniform approximation of g, i.e. $\exists m$, $\exists M$ s.t.

$$g(x) - \gamma m \leq g_\gamma(x) \leq g(x) + \gamma M \qquad \text{for all } x$$

Examples of infimal convolution

We retrieve usual smoothing of the literature:

- Moreau-Yosida: $\omega_\gamma(z) = \frac{1}{2\gamma} \|z\|^2$ [Moreau 1965]

$$g_\gamma^{ic}(x) := \inf_{z \in \mathbb{R}^n} g(z) + \frac{1}{2\gamma} \|z - x\|_2^2$$

- Fenchel-type: $\omega_\gamma = \gamma d^*$, with d strongly convex [Nesterov 2007]

$$g_\gamma^{ic}(x) := \max_{z \in \mathcal{Z}} \langle x, \mathcal{A}z \rangle - \phi(z) - \gamma d(z)$$

  where $\mathcal{A}$ affine function, ϕ convex, and $\mathcal{Z} \subset \mathbb{R}^n$ compact convex set.

- Asymptotic: any smooth ω_γ s.t. $\lim_{\gamma \to 0^+} \omega_\gamma(x) = g(x)$ [Beck, Teboulle 2012]

$$g_\gamma^{ic}(x) := \omega_\gamma(x)$$

Our objective is to obtain g_γ easy to evaluate numerically, possibly explicitly

Examples with Fenchel-type smoothing

Nonsmooth $\sigma(\xi)$	Ball $\mathcal{Z}$	Proximity $\omega(z)$	Smooth surrogate $\sigma(\xi,\gamma)$
$\lvert\xi\rvert$	$[-1,1]$	$\frac{1}{2}\lvert\cdot\rvert^2$	$\begin{cases} \frac{1}{2\gamma}\xi^2 & \text{if } \lvert\xi\rvert \le \gamma \\ \lvert\xi\rvert - \frac{\gamma}{2} & \text{if } \lvert\xi\rvert > \gamma \end{cases}$
$\lvert\xi\rvert$	$[-1,1]$	$(1-\lvert z\rvert)\ln(1-\lvert z\rvert)+\lvert z\rvert$	$f(\xi,\gamma)=\gamma e^{-\left\lvert\frac{\xi}{\gamma}\right\rvert}+\lvert\xi\rvert-\gamma$
$\max_i\{\xi_i,0\}$	$\mathrm{co}(\Delta_n \cup \{\mathbf{0}\})$	$\frac{1}{2}\lVert\cdot\rVert^2$	$\left\langle \xi, \pi_{\mathcal{Z}}\left(\frac{\xi}{\gamma}\right)\right\rangle - \frac{\gamma}{2}\left\lVert \pi_{\mathcal{Z}}\left(\frac{\xi}{\gamma}\right)\right\rVert^2$
$\max_i\{\xi_i,0\}$	$\mathrm{co}(\Delta_n \cup \{\mathbf{0}\})$	$1+\sum_{i=1}^{n} z_i\log(z_i)-z_i$	$\begin{cases} \gamma\left(-1+\sum_{i=1}^{n}\exp(\xi_i/\gamma)\right) & \text{if } \frac{\xi}{\gamma}\in C \\ \gamma\log\left(\sum_{i=1}^{n}\exp(\xi_i/\gamma)\right) & \text{if } \frac{\xi}{\gamma}\in B \end{cases}$

$B = \left\{ s \in \mathbb{R}^n \,\middle|\, \sum_{i=1}^n \exp(s_i) > 1 \right\}$
$C = \left\{ s \in \mathbb{R}^n \,\middle|\, \sum_{i=1}^n \exp(s_i) \le 1 \right\}$
α: permutation that orders in decreasing order

Note:
Statistics and optimization lead to the same surrogate for $\max_i\{x_i, 0\}$

Outline

Algorithms

1. Doubly non-smooth problem to solve:

$$\underset{W \in \mathbb{R}^{d \times k}}{\text{minimize}} \qquad F(W) := R(W) + \lambda \, \|W\|_{\sigma,1}$$

2. Smoothed problem solved with a standard algorithm:

$$\underset{W \in \mathbb{R}^{d \times k}}{\text{minimize}} \quad R^{\gamma}(W) + \lambda \, \|W\|_{\sigma,1}$$

3. Convergence + Explicit formula for good γ [Pierucci et al. 2013]

Theorem (Convergence)

If the iterations W_t are generated with the composite conditional gradient algorithm to solve the smoothed problem, then

$$F(W_t) - \min_{x} F(W) \quad \leq \quad \underbrace{O(\gamma) + O\left(\frac{1}{\gamma t}\right)}_{\varepsilon}$$

i.e. for any ε, it exists $\gamma = O(\varepsilon)$ such that we get an ε-optimal solution for the nonsmooth problem

Overview

1) **Main objective** (Statistical learning): have accurate predictions for new data

$$f_W(x) = y.$$

2) **A modelization** for 1) is to solve

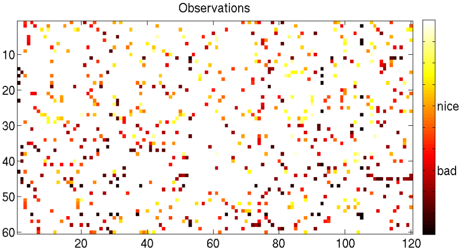
$$\min_W R(W) + \lambda \|W\|_{\sigma,1},$$

because to find low rank linear models is a robust technique for movie recommendation and image classifications.

3) **To optimize** the problem at 2) we are interested in smoothing techniques.

Our contribution is at the point 3), to find accurate solutions to 2), but we keep in mind that the ultimate objective is 1).

Numerical illustration



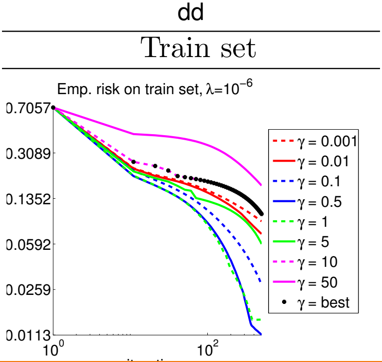
- X with ratings of movies 943(*users*) $\times$ 1682(*movies*)
- $\mathcal{I}$ = indices of known entries (1 %)
- Yellow = ”nice” movie
- Dark red = ”bad” movie

$$\min_{W \in \mathbb{R}^{d \times k}} \quad \underbrace{\frac{1}{N} \sum_{(i,j) \in \mathcal{I}} |W_{ij} - X_{ij}|}_{R_{\mathcal{I}}(W)} \quad + \quad \lambda \, \|W\|_{\sigma,1}$$

Numerical illustration - optimization

- A grid of different values for $\gamma \in \{0.0001, 0.01, 0.1, 0.5, 1, 5, 10, 50\}$
- Each dataset is split into: **train**, **validation**, and **test** sets
- On train we run algorithm for each value of γ.
- At each iteration we obtain parameters W_t^γ and plot $R_{\mathcal{I}_{train}}(W_t^\gamma)$
- Stop criterion = fixed number of iterations.
  Simple, but enough to show the effect of smoothing

Plot of
empirical risk
vs iterations

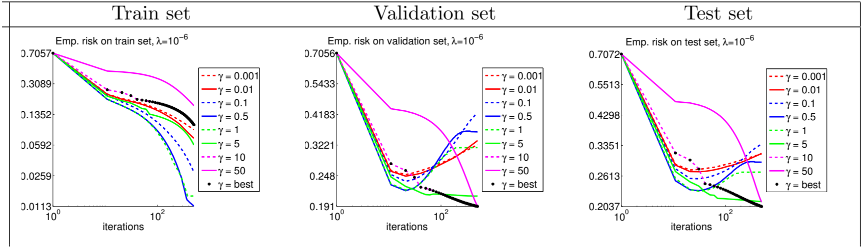


Numerical illustration - learning

1) X^{tr} **Train**

2) X^{val} **Validation**: to chose the best γ, i.e. that makes most accurate predictions. We plot $R_{\mathcal{I}_{validation}}(W_t^\gamma)$

3) X^{ts} **Test:** To check finally the results we plot $R_{\mathcal{I}_{test}}(W_t^\gamma)$



Plots of empirical risk $R_{\mathcal{I}}$ vs iterations

Outline

Conclusions

This research opens

- Choice of $\gamma \Leftarrow$ heavy computations
- Need of a simple automatic way for calibrating γ
- We came up to an “optimal” (in the sense of complexity analysis of the algorithm) and iteration-dependent

$$\gamma_t = O\left(\frac{1}{\sqrt{t}}\right)$$

In this talk

- A way to combine standard algorithms and smooth surrogates
- Two techniques of smoothing
  - Infimal convolution
  - Convolution

Thank you for your attention

- Pierucci, Harchaoui, Malick 2015 - *Smoothing convex functions for nonsmooth optimization* (in preparation)
- Pierucci, Harchaoui, Malick 2015 - *Conditional gradient algorithms for doubly non-smooth learning* (in preparation)
- Pierucci, Harchaoui, Malick 2013 - *A smoothing approach for composite conditional gradient with nonsmooth loss* (CAP conférence Apprentissage)