Iterative regularization for general inverse problems

Guillaume Garrigos

with L. Rosasco and S. Villa

CNRS, École Normale Supérieure

Séminaire CVN - Centrale Supélec - 23 Jan 2018

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$Ax = \bar{y} \tag{P}$$

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$Ax = \bar{y} \tag{P}$$

- Typically $\bar{y} = A\bar{x}$
- Signal/image processing: $\bar{x}$ the original signal deteriorated by A
- Linear regression: (a_i, y_i) the data, $A = (a_1; ...; a_i; ..)$
- Non-linear/Kernel regression/SVM: same but send the a_i's in a feature space

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$Ax = \bar{y} \tag{P}$$

(P) might be ill-posed!

- Typically $\bar{y} = A\bar{x}$
- Signal/image processing: $\bar{x}$ the original signal deteriorated by A
- Linear regression: (a_i, y_i) the data, $A = (a_1; ...; a_i; ..)$
- Non-linear/Kernel regression/SVM: same but send the a_i's in a feature space

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$Ax = \bar{y} \tag{P}$$

(P) might be ill-posed!

- (P) might have no solutions $\to$ introduce a discrepancy

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$x^\dagger = \arg\min D(Ax; \bar{y}) \qquad \text{(P)}$$

(P) might be ill-posed!

- (P) might have no solutions $\to$ introduce a discrepancy $D(Ax; \bar{y}) = \|Ax - \bar{y}\|, \|Ax - \bar{y}\|_1$, or $D_{KL}(Ax; \bar{y})$ ...

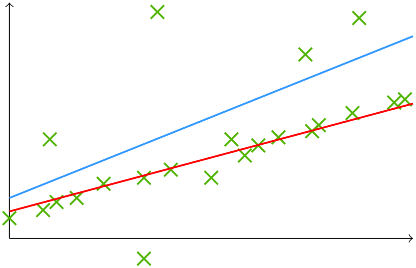
An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$x^\dagger = \arg\min D(Ax; \bar{y}) \qquad \text{(P)}$$

(P) might be ill-posed!

- (P) might have no solutions → introduce a discrepancy $D(Ax; \bar{y}) = \|Ax - \bar{y}\|, \|Ax - \bar{y}\|_1$, or $D_{KL}(Ax; \bar{y})$ ...



An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$x^\dagger = \arg\min D(Ax; \bar{y}) \tag{P}$$

(P) might be ill-posed!

- (P) might have no solutions $\to$ introduce a discrepancy $D(Ax; \bar{y}) = \|Ax - \bar{y}\|, \|Ax - \bar{y}\|_1$, or $D_{KL}(Ax; \bar{y})$ ...
- the solution $x^\dagger$ might be not unique $\to$ introduce a prior

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$x^\dagger = \underset{\arg\min D(Ax;\bar{y})}{\arg\min} \; R(x) \tag{P}$$

(P) might be ill-posed!

- (P) might have no solutions $\to$ introduce a discrepancy $D(Ax;\bar{y}) = \|Ax - \bar{y}\|, \|Ax - \bar{y}\|_1$, or $D_{KL}(Ax;\bar{y})$ ...
- the solution $x^\dagger$ might be not unique $\to$ introduce a prior $R(x)$ is a convex functional ($\|x\|^2$, $\|Wx\|_1$, $\|\nabla x\|$,...)

An ill-posed inverse problem

Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$x^\dagger = \underset{\arg\min D(Ax;\bar{y})}{\arg\min} \; R(x) \tag{P}$$

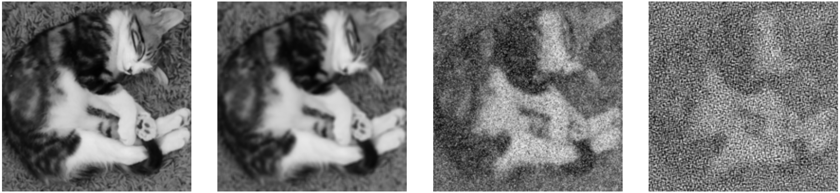
(P) might be ill-posed!

- (P) might have no solutions $\to$ introduce a discrepancy $D(Ax;\bar{y}) = \|Ax - \bar{y}\|$, $\|Ax - \bar{y}\|_1$, or $D_{KL}(Ax;\bar{y})$ ...
- the solution $x^\dagger$ might be not unique $\to$ introduce a prior $R(x)$ is a convex functional ($\|x\|^2$, $\|Wx\|_1$, $\|\nabla x\|$,...)
- (P) is our model.

What about the stability to noise? $\hat{y} = \bar{y} + \varepsilon$

A noisy example

$$x^\dagger = \underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x) \qquad (P)$$



$\bar{x}$ $\qquad$ $\bar{y} = A\bar{x}$ $\qquad$ $\hat{y}$ $\qquad$ $\hat{x}^\dagger$

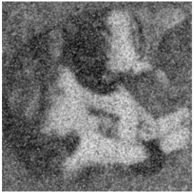
What about the stability to noise? $\hat{y} = \bar{y} + \varepsilon$

A noisy example

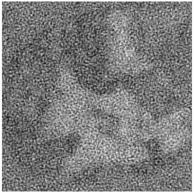
$$x^\dagger = \underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x) \tag{P}$$

$\bar{x}$

$\bar{y} = A\bar{x}$



$\hat{y}$



$\hat{x}^\dagger$

We need to impose well-posedness!

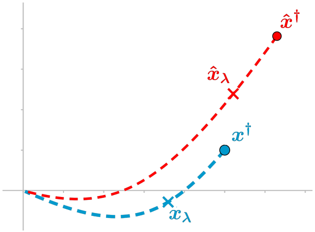
Regularization

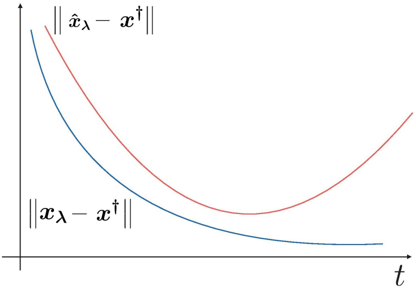
Regularization is a parametrization of a low-dimensional subset of the space of solutions, balancing between fitting the data/model.

Regularization is a parametrization of a low-dimensional subset of the space of solutions, balancing between fitting the data/model.

We want a map $(y, \lambda) \in Y \times \mathcal{P} \mapsto \{x_\lambda(y)\}_{\lambda \in \mathcal{P}} \subset X$ such that

1. $\lim_{\lambda \in \mathcal{P}} x_\lambda(\bar{y}) = x^\dagger$
2. $\|\hat{y} - \bar{y}\| \leq \delta \Rightarrow \exists \lambda_\delta \in \mathcal{P}, \quad \|x_{\lambda_\delta}(\hat{y}) - x^\dagger\| = O\left(\delta^\alpha\right)$





Regularization is a parametrization of a low-dimensional subset of the space of solutions, balancing between fitting the data/model.

We want a map $(y, \lambda) \in Y \times \mathcal{P} \mapsto \{x_\lambda(y)\}_{\lambda \in \mathcal{P}} \subset X$ such that

1. $\lim\limits_{\lambda \in \mathcal{P}} x_\lambda(\bar{y}) = x^\dagger$
2. $\|\hat{y} - \bar{y}\| \leq \delta \Rightarrow \exists \lambda_\delta \in \mathcal{P}, \quad \|x_{\lambda_\delta}(\hat{y}) - x^\dagger\| = O\left(\delta^\alpha\right)$

A good regularization method is a method for which α is big.

$$x^{\dagger} = \operatorname*{arg\,min}_{\operatorname*{arg\,min} D(Ax;\bar{y})} R(x) \tag{P}$$

Which regularization method for our model problem?

Regularization via Perturbation (Tikhonov)

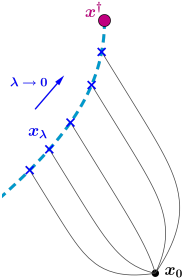
Penalization method

$$x_\lambda(y) := \operatorname*{arg\,min}_{x \in X} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Penalization method

$$x_\lambda(y) := \underset{x \in X}{\arg\min} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

In practice



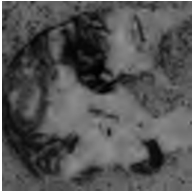
$$(P) \begin{array}{c} \nearrow \\ \to \\ \searrow \end{array} \begin{array}{ccc} (P_{\lambda_1}) & \xrightarrow{\text{optim}} & x_{\lambda_1} \\ (P_{\lambda_2}) & \xrightarrow{\text{optim}} & x_{\lambda_2} \\ (P_{\lambda_3}) & \xrightarrow{\text{optim}} & x_{\lambda_3} \end{array} \begin{array}{c} \searrow \\ \to \\ \nearrow \end{array} \text{reg. path} \xrightarrow{\text{param. selec.}} x_{\lambda_\delta}$$

Regularization via Penalization (Tikhonov)

Penalization method

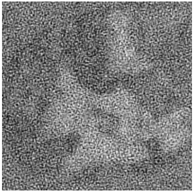
$$x_\lambda(y) := \underset{x \in X}{\arg\min} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Example



$\lambda = 1$

$\lambda = 0.3$



$\lambda = 0.01$

Regularization via Penalization (Tikhonov)

Penalization

$$x_\lambda(y) := \arg\min_{x \in X} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Tikhonov regularization is a regularization method (linear case)

Assume $R(x) = \|x\|^2$, $D(Ax; y) = \|Ax - y\|^2$ and $x^\dagger \in \mathrm{Range}(A^*)$.
Let $\|\hat{y} - \bar{y}\| \leq \delta$ and $\hat{x}_\lambda$ be generated by the data $\hat{y}$.

$$\text{If } \lambda_\delta = O(\delta), \text{ then } \left\| \hat{x}_{\lambda_\delta} - x^\dagger \right\| \lesssim \delta^{\frac{1}{2}}$$

Regularization via Penalization (Tikhonov)

Penalization

$$x_\lambda(y) := \operatorname*{arg\,min}_{x \in X} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Tikhonov regularization is a regularization method (linear case)

Assume $R(x) = \|x\|^2$, $D(Ax; y) = \|Ax - y\|^2$ and $x^\dagger \in \mathrm{Range}(A^*)$.
Let $\|\hat{y} - \bar{y}\| \leq \delta$ and $\hat{x}_\lambda$ be generated by the data $\hat{y}$.

$$\text{If } \lambda_\delta = O(\delta), \text{ then } \left\|\hat{x}_{\lambda_\delta} - x^\dagger\right\| \lesssim \delta^{\frac{1}{2}}$$

- the exponent $1/2$ is optimal
- very few results for other models...

Iterative Regularization (Early stopping)

Early stopping

Take any (robust) algorithm solving directly (P): $\underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x)$

The regularization path is $(x_n)_{n\in\mathbb{N}}$, the parameter is n.

Early stopping

Take any (robust) algorithm solving directly (P): $\underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x)$

The regularization path is $(x_n)_{n\in\mathbb{N}}$, the parameter is n.

In practice

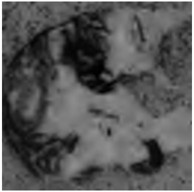
$$(P) \quad \overset{\text{optim}}{\longrightarrow} \quad (x_n)_{n\in\mathbb{N}} \quad \rightarrow \quad \text{reg. path} \quad \overset{\text{param. selec.}}{\longrightarrow} \quad x_{n_\delta}$$

Early stopping

Take any (robust) algorithm solving directly (P): $\underset{x \in \arg\min D(A\cdot;y)}{\arg\min} R(x)$

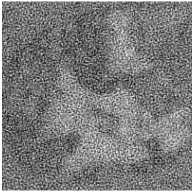
The regularization path is $\{x_n\}$, the parameter is n.

Example



$n = 300$

$n = 500$



$n = 1000$

Early stopping

Take any (robust) algorithm solving directly (P): $\underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x)$

The regularization path is $\{x_n\}$, the parameter is n.

The algorithm(s)

If $D(Ax;y) = \|Ax - y\|^2$ the constraint is linear so the dual of (P) is:

$$\min_u R^*(-A^*u) + \langle u, y \rangle,$$

which could be solved by gradient on the dual:

$$x_n = \nabla R^*(-A^*u_n)$$
$$u_{n+1} = u_n + \tau(Ax_n - y).$$

<u>NB</u>: If $R = \|\cdot\|^2$ it becomes the Landweber algorithm

$$x_{n+1} = x_n - \tau A^*(Ax_n - y).$$

Iterative Regularization (Robust Optimization)

Early stopping

Take any (robust) algorithm solving directly (P): $\underset{x \in \arg\min D(A\cdot;y)}{\arg\min} R(x)$

The regularization path is $\{x_n\}$, the parameter is n.

Gradient descent is a regularization method

Assume $R(x) = \|x\|^2$, $D(Ax; y) = \|Ax - y\|^2$ and $x^\dagger \in \text{Range}(A^*)$.
Let $\|\hat{y} - \bar{y}\| \leq \delta$ and $\hat{x}_n$ be generated by the data $\hat{y}$ via

$$\hat{x}_{n+1} = \hat{x}_n - \gamma A^*(A\hat{x}_n - y).$$

$$\text{If } n_\delta = O(\delta^{-1}), \text{ then } \left\| \hat{x}_{n_\delta} - x^\dagger \right\| \lesssim \delta^{\frac{1}{2}}$$

Iterative Regularization (Robust Optimization)

Early stopping

Take any (robust) algorithm solving directly (P): $\underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x)$.

The regularization path is $\{x_n\}$, the parameter is n.

Gradient Descent on the dual is a regularization [Matet et al., 2016]

Assume $R(x)$ to be strongly convex, $D(Ax; y) = \|Ax - y\|^2$ and $\partial R(x^\dagger) \cap \text{Range}(A^*) \neq \emptyset$. Let $\|\hat{y} - \bar{y}\| \leq \delta$ and $\hat{x}_n$ be generated by the data $\hat{y}$, via Gradient descent on the dual.

$$\text{If } n_\delta = O(\delta^{-1}), \text{ then } \left\|\hat{x}_{n_\delta} - x^\dagger\right\| \lesssim \delta^{\frac{1}{2}}$$

What about other models for D ..?

The learning setting

The learning setting

Let ρ be a distribution on $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{Y} \subset \mathbb{R}$). We want to solve

$$\operatorname*{arg\,min}_{f:\mathcal{X}\to\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} (f(x) - y)^2 d\rho(x, y) \tag{P}$$

The learning setting

Let ρ be a distribution on $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{Y} \subset \mathbb{R}$). We want to solve

$$\operatorname*{arg\,min}_{w \in H_\phi} \int_{\mathcal{X}\times\mathcal{Y}} (\langle w, \phi(x) \rangle - y)^2 d\rho(x, y) \tag{P}$$

The learning setting

Let ρ be a distribution on $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{Y} \subset \mathbb{R}$). We want to solve

$$\operatorname*{arg\,min}_{w \in H_\phi} \int_{\mathcal{X} \times \mathcal{Y}} (\langle w, \phi(x) \rangle - y)^2 d\rho(x, y) \tag{P}$$

But we actually only have access to a sample of the data $(x_i, y_i)_{i=1}^m$.

This means that we pass from minimizing $\|Xw - Y\|$ to $\|X^m w - Y^m\|$.

The learning setting

Let ρ be a distribution on $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{Y} \subset \mathbb{R}$). We want to solve

$$\underset{w \in H_\phi}{\arg\min} \int_{\mathcal{X} \times \mathcal{Y}} (\langle w, \phi(x) \rangle - y)^2 d\rho(x, y) \tag{P}$$

But we actually only have access to a sample of the data $(x_i, y_i)_{i=1}^m$.

This means that we pass from minimizing $\|Xw - Y\|$ to $\|X^m w - Y^m\|$.

- Define a regularization method by looking at $m \to +\infty$ instead of $\delta \to 0$.
- Under reasonable assumptions, the same type of results hold: both Tikhonov and Gradient descent give optimal rates for $\|\hat{w}_{\lambda(m)} - w^\dagger\|$ or $\|\hat{w}_{n(m)} - w^\dagger\|$ [Caponetto, De Vito - 2006].

The learning setting

Let ρ be a distribution on $\mathcal{X} \times \mathcal{Y}$ ($\mathcal{Y} \subset \mathbb{R}$). We want to solve

$$\operatorname*{arg\,min}_{w \in H_\phi} \int_{\mathcal{X} \times \mathcal{Y}} (\langle w, \phi(x) \rangle - y)^2 d\rho(x, y) \tag{P}$$

But we actually only have access to a sample of the data $(x_i, y_i)_{i=1}^m$.

This means that we pass from minimizing $\|Xw - Y\|$ to $\|X^m w - Y^m\|$.

- Define a regularization method by looking at $m \to +\infty$ instead of $\delta \to 0$.
- Under reasonable assumptions, the same type of results hold: both Tikhonov and Gradient descent give optimal rates for $\|\hat{w}_{\lambda(m)} - w^\dagger\|$ or $\|\hat{w}_{n(m)} - w^\dagger\|$ [Caponetto, De Vito - 2006].
- Other algorithms are regularizing, and other parameters are regularizers (passes over the data [Rosasco, Villa - 2015]).

Let's make the point

$$\underset{\arg\min D(Ax;\bar{y})}{\arg\min} \quad R(x) \tag{P}$$

- Penalization and Early stopping are two different regularization methods
- Early stopping seems to have a better complexity in practice
- Penalization lacks theoretical guarantees for general models.
- It is not even clear which algorithm to use for early stopping in general !!

When $D(Ax; y) \neq \|Ax - y\|^2$, how to solve $\underset{\arg\min D(Ax;\bar{y})}{\arg\min} R(x)$?

$\rightarrow$ we cannot use the dual of (P)
$\rightarrow$ Diagonal approach ! (Old idea, see e.g. Lemaire in the 80's)

Diagonal method (heuristic)

Consider any algorithm $x_{n+1} = \text{Algo}(x_n, y, \lambda)$ for solving

$$x_\lambda(y) := \underset{x \in X}{\arg\min}\ \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Instead, do $x_{n+1} = \text{Algo}(x_n, y, \lambda_n)$ with $\lambda_n \to 0$.

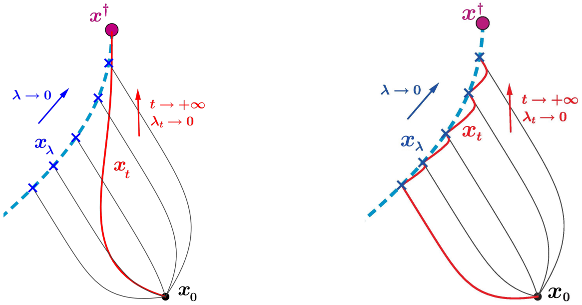
Diagonal method (heuristic)

Consider any algorithm $x_{n+1} = \text{Algo}(x_n, y, \lambda)$ for solving

$$x_\lambda(y) := \arg\min_{x \in X} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Instead, do $x_{n+1} = \text{Algo}(x_n, y, \lambda_n)$ with $\lambda_n \to 0$.

- Includes the warm restart strategy



Diagonal method (heuristic)

Consider any algorithm $x_{n+1} = \text{Algo}(x_n, y, \lambda)$ for solving

$$x_\lambda(y) := \underset{x \in X}{\arg\min}\ \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Instead, do $x_{n+1} = \text{Algo}(x_n, y, \lambda_n)$ with $\lambda_n \to 0$.

- Includes the warm restart strategy
- See [Attouch, Czarnecki, Peypouquet,...] about diagonal FB

Diagonal method (heuristic)

Consider any algorithm $x_{n+1} = \text{Algo}(x_n, y, \lambda)$ for solving

$$x_\lambda(y) := \arg\min_{x \in X} \; \lambda R(x) + D(Ax; y) \qquad (P_\lambda)$$

Instead, do $x_{n+1} = \text{Algo}(x_n, y, \lambda_n)$ with $\lambda_n \to 0$.

- Includes the warm restart strategy
- See [Attouch, Czarnecki, Peypouquet,...] about diagonal FB

Issues

- How to deal with $D(A\cdot; y)$ if D nonsmooth and $A \neq Id$?
- require to know the conditioning of $D(A\cdot; y)$. Might not exist.

Our approach: Diagonal method on Dual problem

Take the dual of (P_λ) $\min_x \lambda R(x) + D(Ax; y)$:

$$\min_u R^*(-A^* u) + \frac{1}{\lambda} D^*(\lambda u; y). \qquad (D_\lambda)$$

Do a diagonal proximal-gradient (Forward-Backward) method on (D_λ), with $\lambda_n \to 0$:

$$x_n = \nabla R^*(-A^* u_n) \qquad \text{(Dual-to-primal step)}$$

$$w_{n+1} = u_n + \tau A x_n \qquad \text{(Forward step)}$$

$$u_{n+1} = w_{n+1} - \tau \operatorname{prox}_{\frac{1}{\tau \lambda_n} D(\cdot;y)} \left(\tau^{-1} w_{n+1}\right) \qquad \text{(Backward step)}$$

Our approach: Diagonal method on Dual problem

Take the dual of (P_λ) $\min_x \lambda R(x) + D(Ax; y)$:

$$\min_u R^*(-A^* u) + \frac{1}{\lambda} D^*(\lambda u; y). \qquad (D_\lambda)$$

Do a diagonal proximal-gradient (Forward-Backward) method on (D_λ), with $\lambda_n \to 0$:

$$\begin{aligned} x_n &= \nabla R^*(-A^* u_n) && \text{(Dual-to-primal step)} \\ w_{n+1} &= u_n + \tau A x_n && \text{(Forward step)} \\ u_{n+1} &= w_{n+1} - \tau \operatorname{prox}_{\frac{1}{\tau\lambda_n} D(\cdot;y)}\left(\tau^{-1} w_{n+1}\right) && \text{(Backward step)} \end{aligned}$$

- Activates D only via its prox
- If $R = J + (1/2)\|\cdot\|^2$ then $\nabla R^* = \operatorname{prox}_J$
- Does it work?

Main result on Diagonal Dual Descent method

Assumptions

- R is strongly convex and $\bar{x} \in \operatorname{dom} R$
- $D(\cdot; \bar{y})$ coercive and p-conditioned
- Qualification condition: $\partial R(x^\dagger) \cap \text{Range}(A^*) \neq \emptyset$

$\rightarrow$ Qualification condition holds if R continuous at $x^\dagger$ and Range(A^*) closed.

Main result on Diagonal Dual Descent method

Assumptions

- R is strongly convex and $\bar{x} \in \operatorname{dom} R$
- $D(\cdot; \bar{y})$ coercive and p-conditioned
- Qualification condition: $\partial R(x^\dagger) \cap \text{Range}(A^*) \neq \emptyset$

Theorem: Optimization (aka no-noise case) [G., Rosasco, Villa - 2017]

Assume that $\lambda_n \to 0$ fast enough (i.e. $\lambda_n \in \ell^{\frac{1}{p-1}}(\mathbb{N})$). Let x_n generated from the true data $\bar{y}$. Then $\|x_n - x^\dagger\| = o(1/\sqrt{n})$.

Main result on Diagonal Dual Descent method

Assumptions

- R is strongly convex and $\bar{x} \in \operatorname{dom} R$
- $D(\cdot; \bar{y})$ coercive and p-conditioned
- Qualification condition: $\partial R(x^\dagger) \cap \text{Range}(A^*) \neq \emptyset$

Assume an additive discrepancy: $D(Ax; y) = L(Ax - y)$.

Theorem: Regularization [G., Rosasco, Villa - 2017]

Assume that $\lambda_n \to 0$ fast enough (i.e. $\lambda_n \in \ell^{\frac{1}{p-1}}(\mathbb{N})$), and $\|\hat{y} - \bar{y}\| \leq \delta$. Let $\hat{x}_n$ generated from the noisy data $\hat{y}$.
Then $\exists n_\delta = O(\delta^{-2/3})$ s.t. $\|\hat{x}_{n_\delta} - x^\dagger\| = O(\delta^{1/3})$.

Main result on Diagonal Dual Descent method

Assumptions

- R is strongly convex and $\bar{x} \in \mathrm{dom}\, R$
- $D(\cdot; \bar{y})$ coercive and p-conditioned
- Qualification condition: $\partial R(x^\dagger) \cap \mathrm{Range}(A^*) \neq \emptyset$

Assume an additive discrepacy: $D(Ax; y) = L(Ax - y)$.

Theorem: Regularization [G., Rosasco, Villa - 2017]

Assume that $\lambda_n \to 0$ fast enough (i.e. $\lambda_n \in \ell^{\frac{1}{p-1}}(\mathbb{N})$), and $\|\hat{y} - \bar{y}\| \leq \delta$. Let $\hat{x}_n$ generated from the noisy data $\hat{y}$.
Then $\exists n_\delta = O(\delta^{-2/3})$ s.t. $\|\hat{x}_{n_\delta} - x^\dagger\| = O(\delta^{1/3})$.

- Similar results for other discrepancies like $D_{KL}(Ax; y)$

Main result on Diagonal Dual Descent method

Assumptions

- R is strongly convex and $\bar{x} \in \operatorname{dom} R$
- $D(\cdot; \bar{y})$ coercive and p-conditioned
- Qualification condition: $\partial R(x^\dagger) \cap \mathrm{Range}(A^*) \neq \emptyset$

Assume an additive discrepacy: $D(Ax; y) = L(Ax - y)$.

Theorem: Regularization [G., Rosasco, Villa - 2017]

Assume that $\lambda_n \to 0$ fast enough (i.e. $\lambda_n \in \ell^{\frac{1}{p-1}}(\mathbb{N})$), and $\|\hat{y} - \bar{y}\| \leq \delta$. Let $\hat{x}_n$ generated from the noisy data $\hat{y}$.
Then $\exists n_\delta = O(\delta^{-2/3})$ s.t. $\|\hat{x}_{n_\delta} - x^\dagger\| = O(\delta^{1/3})$.

- Similar results for other discrepancies like $D_{KL}(Ax; y)$
- Less sharp results suggest that slower $\lambda_n \to 0$ leads to larger n_δ but more accurate $\hat{x}_{n_\delta}$.

Experiments

512×512 images blurred and corrupted by impulse noise (35% intensity)

$D(Xw, y) = \|Ax - y\|_1$ and $F(x) = \|Wx\|_1$ or $\|x\|_{TV}$

512×512 image blurred and corrupted by impulse noise (35% intensity)

t=0

Experiments

512×512 image blurred and corrupted by impulse noise (35% intensity)

t=10

Comments on the experiments

Early stopping VS penalization : who wins?

- Early stopping achieves the same error reconstruction than the Penalization method
- Early stopping requires way less computations than 'stupid' Penalization, but comparable to warm restart strategy
- With Early stopping we have a direct control on the computations, but not on the quality error: fix a budget of iterations, pick the best solution
- With Penalization it is the reverse: fix a stopping criterion for the problems (P_λ), and let run the algorithm

Comments on the experiments

Early stopping VS penalization : who wins?

- Early stopping achieves the same error reconstruction than the Penalization method
- Early stopping requires way less computations than 'stupid' Penalization, but comparable to warm restart strategy
- With Early stopping we have a direct control on the computations, but not on the quality error: fix a budget of iterations, pick the best solution
- With Penalization it is the reverse: fix a stopping criterion for the problems (P_λ), and let run the algorithm

How to choose the parameters ??

- Any technique used for Penalization applies to Early stopping
- In learning, cross-validation works very well
- In imaging it's more delicate (SURE? Discrepancy principle?)

Conclusions

- Early stopping is not limited to linear inverse problems but applies to general models
- Allows for better control of the computational costs than penalization methods

What's next? (work in progress)

- Learning scenario (what if $A \leftrightarrow A_m$?)
- Accelerated method: same reconstruction bound, but faster?
- Removing the strong convexity assumption by using an other algorithm?

Thanks for your attention !