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
1 Regularization of inverse problems

2 Regularization by penalization and early stopping

3 Iterative regularization for general models
An ill-posed inverse problem
Given $A : X \to Y$, and $\bar{y} \in Y$ we want to solve

$$Ax = \bar{y} \quad (P)$$
An ill-posed inverse problem
Given \( A : X \rightarrow Y \), and \( \bar{y} \in Y \) we want to solve

\[
Ax = \bar{y} \quad (P)
\]

- 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, \( A = (a_1; \ldots; a_i; \ldots) \)
- Non-linear/Kernel regression/SVM: same but send the \( a_i \)'s in a feature space
An ill-posed inverse problem

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

\[
Ax = \bar{y}
\]

(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, \( A = (a_1; \ldots; a_i; \ldots) \)
- Non-linear/Kernel regression/SVM: same but send the \( a_i \)'s in a feature space
An ill-posed inverse problem
Given $A : X \rightarrow Y$, and $\bar{y} \in Y$ we want to solve

$$Ax = \bar{y}$$

(P)

(P) might be ill-posed!

- (P) might have no solutions $\rightarrow$ introduce a discrepancy
An ill-posed inverse problem
Given $A : X \to Y$, and $\tilde{y} \in Y$ we want to solve

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

(P) might be ill-posed!

- (P) might have no solutions $\rightarrow$ introduce a discrepancy
$$D(Ax; \tilde{y}) = \|Ax - \tilde{y}\|, \|Ax - \tilde{y}\|_1, \text{ or } D_{KL}(Ax; \tilde{y}) ...$$
An ill-posed inverse problem
Given $A : X \rightarrow Y$, and $\bar{y} \in Y$ we want to solve

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

(P) might be ill-posed!
- (P) might have no solutions $\rightarrow$ introduce a discrepancy

$$D(Ax; \bar{y}) = \|Ax - \bar{y}\|, \|Ax - \bar{y}\|_1, \text{ or } D_{KL}(Ax; \bar{y}) \ldots$$
An ill-posed inverse problem

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

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

(P) might be ill-posed!

- (P) might have no solutions $\rightarrow$ introduce a discrepancy
  $$D(Ax; \tilde{y}) = \|Ax - \tilde{y}\|, \|Ax - \tilde{y}\|_1, \text{ or } D_{KL}(Ax; \tilde{y}) \ldots$$
- the solution $x^\dagger$ might be not unique $\rightarrow$ introduce a prior
An ill-posed inverse problem

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

\[
x^\dagger = \arg\min_{x} \arg\min_{\bar{y}} D(Ax; \bar{y}) \quad \text{(P)}
\]

(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 \( D_{KL}(Ax; \bar{y}) \) ...
- the solution \( x^\dagger \) might be not unique → introduce a prior
  \( R(x) \) is a convex functional (\( \|x\|^2, \|Wx\|_1, \|\nabla x\|,\ldots\)
An ill-posed inverse problem
Given $A : X \to Y$, and $\tilde{y} \in Y$ we want to solve

$$x^\dagger = \arg\min_{x} R(x) \arg\min_{\min \ D(Ax;\tilde{y})}$$

(P)

(P) might be ill-posed!

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

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

A noisy example

$$\hat{x}^\dagger = \arg\min_{\arg\min} R(x) \quad (P)$$

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

$x$  $\bar{y} = A\bar{x}$  $\hat{y}$  $\hat{x}^\dagger$
Intro: Inverse Problems

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

A noisy example

\[
x^\dagger = \arg \min_{x} R(x) \\
\arg \min_{x} D(Ax; \bar{y})
\]  

\( (P) \)

\( \bar{x} \) \hspace{1cm} \( \bar{y} = A\bar{x} \) \hspace{1cm} \( \hat{y} \) \hspace{1cm} \( \hat{x}^\dagger \)

We need to impose well-posedness!
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. $$\|\hat{y} - \bar{y}\| \leq \delta \Rightarrow \exists \lambda_\delta \in \mathcal{P}, \quad \|x_{\lambda_\delta}(\hat{y}) - x^\dagger\| = O(\delta^\alpha)$$

Guillaume Garrigos

5/21
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. \(\lim_{\lambda \in \mathcal{P}} x_\lambda(\tilde{y}) = x^\dagger\)

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

A good regularization method is a method for which \(\alpha\) is big.
1. Regularization of inverse problems

2. Regularization by penalization and early stopping

3. Iterative regularization for general models
Which regularization method for our model problem?

\[
x^\dagger = \arg\min_{\arg\min D(Ax;\tilde{y})} R(x)
\] (P)
Penalization method

\[ x_\lambda(y) := \arg \min_{x \in X} \lambda R(x) + D(Ax; y) \quad (P_\lambda) \]
**Penalization method**

\[ x_\lambda(y) := \arg \min_{x \in \mathcal{X}} \lambda R(x) + D(Ax; y) \quad (P_\lambda) \]

In practice

\[ \lambda \to 0 \quad (P_{\lambda_1}) \xrightarrow{\text{optim}} x_{\lambda_1} \xrightarrow{\text{param. selec.}} x_{\lambda_\delta} \]

\[ \lambda \to 0 \quad (P_{\lambda_3}) \xrightarrow{\text{optim}} x_{\lambda_3} \xrightarrow{\text{reg. path}} \]

\[ \lambda \to 0 \quad (P_{\lambda_2}) \xrightarrow{\text{optim}} x_{\lambda_2} \xrightarrow{\text{optim}} x_{\lambda_3} \xrightarrow{\text{optim}} x_{\lambda_\delta} \]
Penalization method

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

(\( P_\lambda \))

Example

\[ \lambda = 1 \quad \lambda = 0.3 \quad \lambda = 0.01 \]
Penalization

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

(Tikhonov regularization is a regularization method (linear case)

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 \( \hat{x}_\lambda \) be generated by the data \( \hat{y} \).

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

\[ x_\lambda(y) := \arg \min_{x \in X} \lambda R(x) + D(Ax; y) \]  \hspace{1cm} (P_\lambda)

Tikhonov regularization is a regularization method (linear case)

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

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

- the exponent 1/2 is optimal
- very few results for other models...
Early stopping
Take any (robust) algorithm solving directly \((P)\):

\[
\arg\min_x R(x) \quad \arg\min_{\tilde{y}} D(Ax;\tilde{y})
\]

The regularization path is \((x_n)_{n \in \mathbb{N}}\), the parameter is \(n\).
Iterative Regularization (Early stopping)

Early stopping
Take any (robust) algorithm solving directly \((P)\): \[
\arg \min_{x} R(x) \quad \arg \min_{x} D(Ax; \bar{y})
\]
The regularization path is \((x_n)_{n \in \mathbb{N}}\), the parameter is \(n\).

In practice
\[(P) \xrightarrow{\text{optim}} (x_n)_{n \in \mathbb{N}} \rightarrow \text{reg. path} \xrightarrow{\text{param. selec.}} x_{n_\delta}\]
Iterative Regularization (Early stopping)

Early stopping
Take any (robust) algorithm solving directly \((P)\):

\[
\arg \min_{x \in \arg \min_{x \in \arg \min} D(A \cdot ; y)} R(x)
\]

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

Example

\[
\begin{align*}
n &= 300 \\
n &= 500 \\
n &= 1000
\end{align*}
\]
Early stopping
take any (robust) algorithm solving directly \((P)\):
\[
\arg\min_x R(x)
\]
arg\min \(D(Ax; \tilde{y})\)

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

The algorithm(s)
If \(D(Ax; y) = \|Ax - y\|_2^2\) 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:
\[
x_n = \nabla R^*(-A^*u_n)
\]
\[
u_{n+1} = u_n + \tau(Ax_n - y).
\]

NB: If \(R = \| \cdot \|_2^2\) it becomes the Landweber algorithm
\[
x_{n+1} = x_n - \tau A^*(Ax_n - y).
\]
Early stopping
Take any (robust) algorithm solving directly \((P)\):
\[
\arg \min_{x \in \text{arg min} D(A; y)} R(x)
\]
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 \(\|\hat{y} - \tilde{y}\| \leq \delta\) and \(\hat{x}_n\) be generated by the data \(\hat{y}\) via
\[
\hat{x}_{n+1} = \hat{x}_n - \gamma A^*(A\hat{x}_n - y).
\]
If \(n_\delta = O(\delta^{-1})\), then \(\left\|\hat{x}_{n_\delta} - x^\dagger\right\| \lesssim \delta^{\frac{1}{2}}\).
Early stopping
Take any (robust) algorithm solving directly \((P)\): 
\[
\arg\min_x R(x) \quad \arg\min \quad \arg\min \quad \arg\min D(Ax; \bar{y})
\]
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.

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

What about other models for \(D \ldots\?)
The learning setting

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

$$\arg \min_{w \in H} \phi \int_{X \times Y} (\langle w, \phi(x) \rangle - y)^2 d\rho(x, y).$$

But we actually only have access to a sample of the data $(x_i, y_i)_{m=1}^m$. 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 $\|\hat{w}_\lambda(m) - w^*\|$ or $\|\hat{w}_n(m) - w^*\|$. [Caponetto, De Vito - 2006].

Other algorithms are regularizing, and other parameters are regularizers ([Rosasco, Villa - 2015]).
The learning setting

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

$$\arg \min_{f: \mathcal{X} \to \mathcal{Y}} \int_{\mathcal{X} \times \mathcal{Y}} (f(x) - y)^2 d\rho(x, y)$$

(P)
The learning setting

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

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

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 $\|\hat{w}_\lambda^{(m)} - w^\ast\|$ or $\|\hat{w}_n^{(m)} - w^\ast\|$. Other algorithms are regularizing, and other parameters are regularizers [Caponetto, De Vito - 2006]. Other parameters are regularizers [Rosasco, Villa - 2015].
The learning setting

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

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

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 $\|X^mw - Y^m\|$. 

Guillaume Garrigos

10/21
The learning setting

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

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

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 $\|\hat{w}_\chi(m) - w^\dagger\|$ or $\|\hat{w}_n(m) - w^\dagger\|$ [Caponetto, De Vito - 2006].
The learning setting

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

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

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

\[ \arg\min_{x} R(x) \]
\[ \arg\min_{D} D(Ax; \tilde{y}) \]

- 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!!
1. Regularization of inverse problems

2. Regularization by penalization and early stopping

3. Iterative regularization for general models
When $D(Ax; y) \neq \|Ax - y\|^2$, how to solve $\arg\min_{x} R(x)$?

$\arg\min_{D(Ax; \bar{y})} D(Ax; \bar{y})$

→ we cannot use the dual of $(P)$
→ 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

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

(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

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

(\(P_\lambda\))

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

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

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

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

$$(P_{\lambda})$$

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 \((P_\lambda)\) \(\min_x \lambda R(x) + D(Ax; y)\):

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

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

\[
\begin{align*}
x_n &= \nabla R^*(-A^*u_n) \quad \text{(Dual-to-primal step)} \\
w_{n+1} &= u_n + \tau Ax_n \quad \text{(Forward step)} \\
u_{n+1} &= w_{n+1} - \tau \text{prox}_{\frac{1}{\tau \lambda_n} D(\cdot; y)} \left( \tau^{-1} w_{n+1} \right) \quad \text{(Backward step)}
\end{align*}
\]
Our approach: Diagonal method on Dual problem

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

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

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

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

- Activates \(D\) only via its prox
- If \(R = J + (1/2)\|\cdot\|^2\) then \(\nabla R^* = \text{prox}_J\)
- Does it work?
Main result on Diagonal Dual Descent method

Assumptions

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

→ Qualification condition holds if $R$ continuous at $x^\dagger$ and $\text{Range}(A^*)$ closed.
Assumptions

- \( R \) is strongly convex and \( \bar{x} \in \text{dom } R \)
- \( D(\cdot; \bar{y}) \) 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. \( \lambda_n \in \ell^{1/p-1}(\mathbb{N}) \)). 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 $\bar{x} \in \text{dom} \: R$
- $D(\cdot; \bar{y})$ 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 $\bar{x} \in \text{dom } R$
- $D(\cdot; \bar{y})$ 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. $\|\hat{x}_{n_\delta} - x^\dagger\| = O(\delta^{1/3})$.

- Similar results for other discrepancies like $D_{KL}(Ax; y)$
Main result on Diagonal Dual Descent method

Assumptions

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

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. $\|\hat{x}_{n_\delta} - x^\dag\| = O(\delta^{1/3})$.

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

$512 \times 512$ images blurred and corrupted by impulse noise (35\% intensity)

$D(X_w, y) = \|Ax - y\|_1 \text{ and } F(x) = \|Wx\|_1 \text{ or } \|x\|_{TV}$
512 × 512 image blurred and corrupted by impulse noise (35% intensity)
512 × 512 image blurred and corrupted by impulse noise (35% intensity)
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 ($P_\lambda$), 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?)
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 ($P_\lambda$), 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!