Inverse Problems with Poisson Noise
A Sparsity and Optimization Tour

Jalal Fadili
GREYC, CNRS-ENSICAEN-Université de Caen,
http://www.greyc.ensicaen.fr/~jfadili

Joint work with F.-X. Dupé, B. Zhang, J. Schmitt and J.-L. Starck
ADA VII 2012
Poisson noise in astronomy
Poisson noise in astronomy

Ubiquitous in astronomical data sets.
Poisson noise in astronomy

- Ubiquitous in astronomical data sets.
- Stems from a photon counting process.
Poisson noise in astronomy

- Ubiquitous in astronomical data sets.
- Stems from a photon counting process.
- Low-count situations.
Poisson noise in astronomy

- Ubiquitous in astronomical data sets.
- Stems from a photon counting process.
- Low-count situations.
- Typically isotropic sources but could be anisotropic.

![Intensity, Counts, Isotropic+line-like sources](image)
Poisson noise in astronomy

- Ubiquitous in astronomical data sets.
- Stems from a photon counting process.
- Low-count situations.
- Typically isotropic sources but could be anisotropic.
- Monochannel or multispectral.
Poisson noise in astronomy

- Ubiquitous in astronomical data sets.
- Stems from a photon counting process.
- Low-count situations.
- Typically isotropic sources but could be anisotropic.
- Monochannel or multispectral.
- On Cartesian or spherical grids.
Poisson noise in astronomy

- Ubiquitous in astronomical data sets.
- Stems from a photon counting process.
- Low-count situations.
- Typically isotropic sources but could be anisotropic.
- Monochannel or multispectral.
- On Cartesian or spherical grids.
- Typical instruments: XMM, GLAST/Fermi.
What is an inverse problem?

Building blocks to solve an inverse problem.

Data fidelity.

Regularization: SparseLand.

Optimization problems and algorithms.

Parameter(s) choice.

Take-away messages.
What is an inverse problem?

Building blocks to solve an inverse problem.

Data fidelity.

Regularization: SparseLand.

Optimization problems and algorithms.

Parameter(s) choice.

Take-away messages.
What is an inverse problem?

$H$ is a bounded (here linear) degradation operator (typically alters content).

$\Xi$ is a non-linear mapping not necessarily linear nor invertible (e.g. sensor-specific, VST, etc).

$\odot$ is any composition of two arguments (e.g. ‘+’, ‘.’).

$\epsilon$ is the noise process (e.g. additive white Gaussian, speckle, Poisson, etc.).

Throughout the lecture: finite-dimensional setting, typically $\mathbb{R}^N$. 

**Forward model**

$$ y = \Xi \left( \begin{array}{c} \text{Measurement/degradation} \\ \odot \\ \epsilon \end{array} \right)$$

**Prior knowledge (regularization)**
What is an inverse problem?

\[ y = \Xi(\mathbf{H}f + \varepsilon) \]

- \( \mathbf{H} \) is a bounded (here linear) degradation operator (typically alters content).
- \( \Xi \) is a non-linear mapping not necessarily linear nor invertible (e.g. sensor-specific, VST, etc).
- \( \circ \) is any composition of two arguments (e.g. '+', '·').
- \( \varepsilon \) is the noise process (e.g. additive white Gaussian, speckle, Poisson, etc.).
- Throughout the lecture: finite-dimensional setting, typically \( \mathbb{R}^N \).

Objective

Recover \( f \) from \( y \) is an ill-posed inverse problem.
Denoising with Poisson noise

\[ \Xi : x \mapsto x, \quad H = I, \quad \varepsilon \sim \mathcal{P}(f) \]
Inpainting with Poisson noise

\[ \Xi : x \mapsto x, \quad H : f \mapsto f_\Omega, \quad \varepsilon \sim \mathcal{P}(f_\Omega) \]
Deconvolution with Poisson noise

\[ \Xi : x \mapsto x, \quad H : f \mapsto f \ast h, \quad \varepsilon \sim \mathcal{P}(f \ast h) \]
Deconvolution with AWGN

\[ \Xi: x \mapsto x, \quad H: f \mapsto f \ast h \]
\[ \circ: +, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2) \]
Image decomposition

Galaxy SBS 0335-052 from Gemini instrument

\[ \Xi : x \mapsto x \]

\[ H : \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} \mapsto f_1 + f_2 \]

\[ \odot : + \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2) \]
A problem is well-posed in Hadamard sense [Hadamard 1923] if the following holds:

(i) **Existence**: there is at least one solution.

(ii) **Uniqueness**: the set of solutions converge to a unique solution.

(iii) **Stability**: the solution depends continuously on the measurements.

Stability is most often violated.

Simple example: a linear noiseless forward model $y = Hf$.

- Determined or overdetermined: stability depends on conditioning of $H$:

$$\frac{\|f - f_\varepsilon\|_p}{\|f\|_p} \leq \|H\|_p \|H^{-1}\|_p \frac{\|\varepsilon\|_p}{\|y\|_p}, \quad p \geq 1.$$ 

- Underdetermined case: no unique solution by the fundamental theorem of linear algebra (more unknowns than equations).
Outline

- What is an inverse problem?
- Building blocks to solve an inverse problem.
  - Data fidelity.
  - Regularization: SparseLand.
  - Optimization problems and algorithms.
  - Parameter(s) choice.
- Take-away messages.
Building blocks

Forward model

Inverse problems in image processing

Prior knowledge on the solution

Optimization theory
Building blocks

Forward model

Inverse problems in image processing

- Degradation model.
- Prior image model.

Prior knowledge on the solution

Optimization theory
Building blocks

Forward model

Inverse problems in image processing

- Degradation model.
- Prior image model.
- Optimize to estimate.

Prior knowledge on the solution

Optimization theory
Building blocks

Forward model

Inverse problems in image processing

Prior knowledge on the solution

Optimization theory

Solution properties (uniqueness, stability, etc.)

Degradation model.

Prior image model.

Optimize to estimate.
Building blocks: Example

\[ y = h \ast f + \varepsilon \]

\( \varepsilon \), e.g. additive white Gaussian noise (AWGN)
Building blocks: Example

\[ y = h \ast f + \varepsilon \]

\( \varepsilon \), e.g. additive white Gaussian noise (AWGN)
Building blocks: Example

Forward model

Blur PSF and noise properties

Deconvolution

Data fidelity.

Image $f$

Deconvolution

$y = h \ast f + \varepsilon$

$\varepsilon$, e.g. additive white Gaussian noise (AWGN)
Building blocks: Example

Forward model

Blur PSF and noise properties
*Data fidelity.*

Image piecewise smooth
*Prior.*

Deconvolution

Regularization

\[ y = h \ast f + \varepsilon \]

\(\varepsilon\), e.g. additive white Gaussian noise (AWGN)
Building blocks: Example

Forward model

- Deconvolution
- Objective to optimize (convergence, global local, rate)

Regularization

- Image piecewise smooth (Prior)

Optimization theory

- Blur PSF and noise properties (Data fidelity)

Deconvolution

\[ y = h \ast f + \varepsilon \]

- \( \varepsilon \), e.g. additive white Gaussian noise (AWGN)

Image, Deconvolution, PSF, Noise
Building blocks: Example

Forward model

Deconvolution

Blur PSF and noise properties
*Data fidelity.*

Image piecewise smooth
*Prior.*

Properties
*Uniqueness, recovery, other guarantees.*

Objective to optimize
*Algorithm (convergence, global local, rate).*

Regularization

Optimization theory

Image \( f \)

\( y = h \ast f + \varepsilon \)

\( \varepsilon \), e.g. additive white Gaussian noise (AWGN)
Outline

- What is an inverse problem?
- Building blocks to solve an inverse problem.
  - Data fidelity.
  - Regularization: SparseLand.
- Optimization problems and algorithms.
- Parameter(s) choice.
- Take-away messages.
Data fidelity and forward model

\[ -\log p_\epsilon(y|f) = F_\epsilon \circ H(f) \]
Data fidelity

Data fidelity and forward model

\[- \log p_\varepsilon(y|f) = F_\varepsilon \circ H(f)\]

Additive noise \( Y|f \sim \mathcal{N}(Hf, \Sigma_\varepsilon) \), \( \Sigma_\varepsilon \succ 0 \).

\[
F_{\text{Gaussian}}(g) = \frac{1}{2} (y - g)^T \Sigma_\varepsilon^{-1} (y - g).
\]
Data fidelity

Data fidelity and forward model

\[- \log p_\varepsilon(y|f) = F_\varepsilon \circ H(f)\]

Poisson noise \(Y|f \sim \mathcal{P}(Hf), \forall i, (Hf)[i] \geq 0\).

\[
F_{\text{Poisson}}(g) = \sum_{i=1}^{N} F_{\text{p}}(g[i]), \text{ if } y[i] > 0,
\]

\[
F_{\text{p}}(g[i]) = \begin{cases} 
- y[i] \log(g[i]) + g[i] & \text{if } g[i] > 0, \\
+ \infty & \text{otherwise,}
\end{cases}
\]

if \(y[i] = 0\),

\[
F_{\text{p}}(g[i]) = \begin{cases} 
g[i] & \text{if } g[i] \in [0, +\infty), \\
+ \infty & \text{otherwise.}
\end{cases}
\]
Outline

- What is an inverse problem?
- Building blocks to solve an inverse problem.
- Data fidelity.
- Regularization: SparseLand.
- Optimization problems and algorithms.
- Parameter(s) choice.
- Take-away messages.
Analysis and Synthesis Priors

**Synthesis**

Generative linear model

\[ f = \Phi x = \sum_{i=1}^{P} \varphi_i x[i] \]

- Typical examples: X-let (synthesis) dictionary.
- The images \( f \) are confined to the column space of the dictionary \( \Phi \).
- A constructive form providing an explicit description to synthesize rich and complex signals.

**Analysis**

Correlation model

\[ x = D^* f = (\langle d_i, f \rangle)_{i=1}^{P} \]

- Typical examples: X-let (analysis), discrete derivatives, fused.
- The signals \( f \) are arbitrary vectors in \( \mathbb{R}^N \).
- For redundant \( D \), much fewer samples than coefficients (less unknowns).
Analysis vs Synthesis sparse recovery

**Synthesis**

\[
\min_{x \in \mathbb{R}^P} \frac{1}{2} \|y - H\Phi x\|_2^2 + \lambda \|x\|_1
\]

\[P = 3\]

**Analysis**

\[
\min_{f \in \mathbb{R}^N} \frac{1}{2} \|y - Hf\|_2^2 + \lambda \|D^* f\|_1
\]

\[N = 2\]
Sparsity and functional spaces

\[ f = \Phi x = \sum_{i=1}^{P} \varphi_i x[i] \]
\[ x = D^* f = (\langle d_i, f \rangle)_{i=1}^{P} \]

Strictly sparse signals: \( \|x\|_0 = |\text{supp}(x)| = K \ll N. \)
Sparsity and functional spaces

\[ f = \Phi x = \sum_{i=1}^{P} \varphi_i x[i] \]

\[ x = D^* f = (\langle d_i, f \rangle)_{i=1}^{P} \]

Compressible signals: \( x \in w\ell_q(C) \)

\[ |x(i)| \leq C i^{-1/q} \]

\[ \|x - x_K\|_2 \leq C_q K^{1/2-1/q}, q < 2 \]

\( \Phi \) a (dual) frame

\[ \|f - f_K\|_2 \leq C_q A^{-1} K^{1/2-1/q}, q < 2 \]
Wavelets for isotropic structures

\[ f = \sum_{k=0}^{2^{J_c} - 1} \langle f, \phi_{J_c,k} \rangle \tilde{\phi}_{J_c,k} + \sum_{j=J_c}^{\infty} \sum_{k=0}^{2^j - 1} \langle f, \psi_{j,k} \rangle \tilde{\psi}_{j,k}, \]

- \((\phi, \psi, \tilde{\phi}, \tilde{\psi})\) are defined by a perfect reconstruction FB \((h, g, \tilde{h}, \tilde{g})\).
- A (tight) frame expansion (oversampled) or a (bi)orthogonal basis (critically sampled).
- Optimally sparse over Besov spaces, images with pointwise singularities.
The curvelet transform provides a multiresolution, directional representation with basis elements well localized in both space and frequency.

Oscillating behavior: \( \varphi_j \) is a little needle whose envelope is a specified "ridge" of effective length \( 2^{-j/2} \) and width \( 2^{-j} \), and which displays an oscillatory behavior across the main "ridge".

Optimally sparse representation of piecewise \( C^2 \) images with \( C^2 \) edges.
Wave atoms for oscillating textures

A basis of wavelet packets obeying the parabolic scaling wavelength = (diameter)^2.

Wave Atoms provide a multiresolution, directional representation with basis elements well localized in both space and frequency.

Oscillating behavior: an oscillatory pattern with support 2^{-j} and frequency 2^{2j}.

Optimally sparse representation of oriented warped oscillatory patterns.
Morphological diversity

\[ f = \sum_{k=1}^{K} f_k \]

Isotropic structures

Anisotropic structures

Oscillating textures

e tc.
Highly redundant dictionaries

- **Morphological diversity** ⇒ Overcompleteness.
- **Fast** ⇒ Implicit analysis and synthesis operators.

- Wavelets
  (Pointwise singularities, isotropic structures)
- Local DCT, WaveAtom
  (Locally oscillatory, stationary textures)
- Curvelets
  (Piecewise smooth with $C^2$ contours)
- Others
  \[ \text{width} = \text{length}^2 \]
- Others
Sparsity penalties

The graph illustrates various sparsity penalties with different parameters. The y-axis represents the penalty value, and the x-axis represents the input value. The penalties are categorized into two groups: Convex and Non-convex.

- Convex penalties: These include the L0, L1, L2, and L-infinity norms, represented by blue, red, green, and cyan lines, respectively.
- Non-convex penalties: These include the Lp norms for p = 0.1, 0.5, 0.8, and 1.2, represented by green, red, teal, and magenta lines, respectively.

The Huber penalty is also shown in black, which is a hybrid of the L1 and L2 norms.
This talk
Structured sparsity

Capture structure beside sparsity, e.g.:
- Local dependencies in clusters.
- Dependencies inherited from scale persistence.
- Tree structures.

Models:
- Sparsity penalties with localizing operators (penalized estimators).
- Stochastic models: join, MRF and more generally graphical models (Bayesian inference).

Non-overlapping blocks or groups

Overlapping blocks or groups

More complex dependencies
Structured sparsity penalties

\[ J(x) = \sum_{i \in I} \psi_i(B_i x) = \Psi \circ B(x) \]

- \( B_i : \mathbb{R}^N \rightarrow \mathbb{R}^{N_i} \) is a countable family of localization operators \( i \in I \) such that

\[ Bx = (B_i x)_{i \in I} \in \Omega = \prod_{i \in I} \mathbb{R}^{N_i} \]

- \( \psi_i : \mathbb{R}^{N_i} \rightarrow \mathbb{R}^+ \) is a sparsity-promoting penalty with

\[ \forall u = (u_i)_{i \in I} \in \Omega, \quad \Psi(u) = \sum_{i \in I} \psi_i(u_i). \]

- The classical example is the \( \ell_p \)-norm

\[ \psi_i(v) = \|v\|_p, \]

where typically \( p > 1 \) to promote group-sparsity.
Outline

- What is an inverse problem?
- Building blocks to solve an inverse problem.
- Data fidelity.
- Regularization: SparseLand.
- Optimization problems and algorithms.
- Parameter(s) choice.
- Take-away messages.
Class of problems

Inverse problems with mixed regularization, e.g.:

\[
\min_{x \in \mathcal{H}} \left( F(x) + G_1(x) + \cdots + G_n(x) \right)
\]

- **Data fidelity**
- **Regularization, constraints**

**Assumption** All functions are proper lsc convex, with appropriate domain qualification conditions.

Covers many other applications beyond signal/image processing: machine learning, statistical estimation, etc.

**Inverse problem**

\[
y = A x + \epsilon
\]

**Forward model**

\[
y = A x + \epsilon
\]
Example: Deconvolution with Poisson noise

- Sparse in $\Phi$.
- Positive.
- Preserves total flux.

$y = h \ast f + \varepsilon \sim P(f \ast h)$

Forward model
Example: Deconvolution with Poisson noise

- Sparse in $\Phi$.
- Positive.
- Preserves total flux.

Forward model

$$\min_f - \log p_\varepsilon (y | f) + \lambda \| \Phi^* f \|_1 + \nu\{u \geq 0\}(f) + \nu\{\sum_i u[i] = \text{cst}\}(f)$$
Example: Deconvolution with Poisson noise

\[ y = h \ast f + \varepsilon \sim \mathcal{P}(f \ast h) \]

- Sparse in \( \Phi \).
- Positive.
- Preserves total flux.

Forward model

\[
\begin{aligned}
\min_f & \quad - \log p_\varepsilon(y | f) + \lambda \| \Phi^* f \|_1 + \nu\{u \geq 0\}(f) + \nu\{\sum_i u[i] = \text{cst}\}(f) \\
F(f) & \\
\end{aligned}
\]
Example: Deconvolution with Poisson noise

Inverse problem

Inverse problem

Sparse in $\Phi$.

Positive.

Preserves total flux.

Forward model

$\min_f \ - \ \log p_\varepsilon(y|f) + \lambda \| \Phi^* f \|_1 + \nu\{u_\geq 0\}(f) + \nu\{\sum_i u[i]=\text{cst}\}(f)$

Data fidelity

$F(f)$

Sparsity

(Analysis)

$G_1(f)$

$\varepsilon \sim \mathcal{P}(f \star h)$
Example: Deconvolution with Poisson noise

- Sparse in $\Phi$.
- Positive.
- Preserves total flux.

\[
\min_f \quad - \log p_\varepsilon(y|f) + \lambda \left\| \Phi^* f \right\|_1 + \nu_{\{u \geq 0\}}(f) + \nu_{\{\sum_i u[i] = \text{cst}\}}(f)
\]

Data fidelity

\[
F(f)
\]

Sparsity (Analysis)

\[
G_1(f)
\]

Positivity

\[
G_2(f)
\]
Example: Deconvolution with Poisson noise

- Sparse in $\Phi$.
- Positive.
- Preserves total flux.

**Forward model**

$$
\min_f \left[ - \log p_\varepsilon(y|f) + \lambda \| \Phi^* f \|_1 + \iota\{u \geq 0\}(f) + \iota\{\sum_i u[i] = \text{cst}\}(f) \right]
$$

- Data fidelity: $F(f)$
- Sparsity (Analysis): $G_1(f)$
- Positivity
- Flux preservation: $G_3(f)$
Example: Deconvolution with Poisson noise

Inverse problem

Main challenge

How to solve such (non-smooth) optimization problems

\[
\min_f - \log p_\varepsilon(y|f) + \lambda \| \Phi^* f \|_1 + \nu\{ u \geq 0 \}(f) + \nu\{ \sum_i u[i] = \text{cst} \}(f)
\]

Data fidelity

\( F(f) \)

Sparsity

(Analysis)

\( G_1(f) \)

Positivity

\( G_2(f) \)

Flux preservation

\( G_3(f) \)

Sparse in \( \Phi \).

Positive.

Preserves total flux.
Proximal splitting: A glimpse

\[
\min_{x \in \mathcal{H}} \left( F(x) + G_1(x) + \cdots + G_n(x) \right)
\]

Data fidelity \hspace{1cm} \text{Regularization, constraints}

- Methods designed to solve non-smooth but structured convex optimization problems.
- Goals:
  - Exploit the structure of the problem (composite additive): sequence of calculations involving only each function at a time.
  - Exploit the properties of the individual functions: e.g. simple (to be defined shortly), smooth, separable, etc..
  - Deal with large scale data.
  - Avoid nested algorithms.
- A whole field in optimization theory.
Proximal splitting: Example

\[ \min_{x \in C \subset \mathcal{H}} F(x) \iff \min_{x \in \mathcal{H}} F(x) + \nu_C(x) \]

- Proper, lsc, convex.
- \( \text{dom}(F) \cap C \neq \emptyset. \)
- \( \nabla F \) is \( \beta \)-Lipschitz.
- Non empty.
- Closed.
- Convex.
Proximal splitting: Example

\[
\min_{x \in C \subset \mathcal{H}} F(x) \iff \min_{x \in \mathcal{H}} F(x) + \nu_C(x)
\]

- Proper, lsc, convex.
- \( \text{dom}(F) \cap C \neq \emptyset \).
- \( \nabla F \) is \( \beta \)-Lipschitz.
- Non empty.
- Closed.
- Convex.

\[
\mathbf{x}_{k+1} = \text{proj}_C \left( \mathbf{x}_k - \mu_k \nabla F(\mathbf{x}_k) \right)
\]

Projection

Gradient descent (Landweber)

\[
\text{proj}_C(x) = \arg \min_{z \in C} \frac{1}{2} \|x - z\|_2^2
\]
Proximal splitting: Example

\[
\min_{x \in \mathcal{H}} F(x) + G(x)
\]

- Proper, lsc, convex.
- \( \text{dom}(F) \cap \text{dom}(G) \neq \emptyset \).
- \( \nabla F \) is \( \beta \)-Lipschitz.

\[
\begin{align*}
\mathbf{x}_{k+1} &= \text{prox}_{\mu_k G} (\mathbf{x}_k - \mu_k \nabla F(\mathbf{x}_k)) \\
&= \text{prox}_{\gamma G} (\mathbf{x}) = \arg\min_{z \in \mathcal{H}} \frac{1}{2} \| \mathbf{x} - z \|_2^2 + \gamma G(\mathbf{x}), \quad \gamma > 0
\end{align*}
\]

\( G \) is simple if \( \text{prox}_{\gamma G} \) has closed-form.

Implicit step

Gradient descent (explicit step)
Example: Iterative Soft thresholding

$$\min_{x \in \mathcal{H}} \frac{1}{2} \| y - A x \|_2^2 + \lambda \| x \|_1$$
Example: Iterative Soft thresholding

\[
\min_{x \in \mathcal{H}} \frac{1}{2} \| y - Ax \|^2_2 + \lambda \| x \|_1
\]

\[
x_{k+1} = \text{SoftThresh}_{\mu_k \lambda}(x_k + \mu_k A^*(y - Ax_k))
\]
Many, many others ...

Many, many other schemes for more complicated functionals, e.g.:
- Forward-Bakcward-Forward.
- Generalized Forward-Backward.
- Douglas-Rachford.
- ADMM and variants.
- Primal-Dual splitting.
... and Poisson noise?

$$\min_f -\log p_\varepsilon(y|f) + \lambda \|\Phi^* f\|_1 + \nu\{u \geq 0\}(f) + \nu\{\sum_i u[i] = \text{cst}\}(f)$$

- Data fidelity: $F(f)$
- Sparsity (Analysis): $G_1(f)$
- Positivity: $G_2(f)$
- Flux preservation: $G_3(f)$

Several alternatives e.g. [Dupe-F.-Starck 08-11]:

- Douglas-Rachford (on product spaces) or,
- Primal-Dual splitting.

Simple (non smooth) functions (up to linear operators):

- Simple data fidelity ($F$).
- Soft thresholding ($G_1$).
- Simple projections ($G_2$ and $G_3$).
Outline

- What is an inverse problem?
- Building blocks to solve an inverse problem.
- Data fidelity.
- Regularization: SparseLand.
- Optimization problems and algorithms.
- Parameter(s) choice.
- Take-away messages.
The priors depend on parameters (e.g. $\lambda$).

How to choose or estimate good ones?

Important questions:

- What is a good parameter?
- Quantitative definition?
- Guarantees on $f$ or $Hf$?
- Handle complicated noise distribution (here Poisson)?
Parameter(s) selection

- Equivalent formulations (e.g. constrained form).
- Learned from exemplars: classical estimation theory (e.g. MLE).
- Hierarchical Bayesian models: put priors on hyperparameters and infer them along with $f$ (stochastic sampling).
- Asymptotics: what asymptotic properties the estimator of $f$ or $Hf$ should enjoy.
- A posteriori rules:
  - Mozorov discrepancy principle.
  - $L$-curve.
- Model selection and unbiased risk estimation:
  - On prediction $Hf$: e.g. GCV, SURE.
  - On $f$: GSURE.
- But things get more involved with Poisson noise.
Outline

- What is an inverse problem?
- Building blocks to solve an inverse problem.
- Data fidelity.
- Regularization: SparseLand.
- Optimization problems and algorithms.
- Stylized examples.
- Take-away messages.
Take-away messages

Building blocks to solve Poisson inverse problems:
- Data fidelity (forward model).
- Regularization and constraints.
- Optimization algorithms.

Many issues to tackle:
- Wise priors: sparsity, dictionary, positivity.
- Provably convergent and fast solvers for large-scale problems (proximal splitting).
- Parameter(s) choice.

A wide variety of applications: denoising, inpainting, deconvolution, tomography, multispectral data, etc..

Other problems.

Accelerated algorithms.

Theoretical recovery guarantees remain to be investigated.
Extended experiments, toolboxes available
http://www.greyc.ensicaen.fr/~jfadili
http://www.sparsesignalrecipes.info
http://www.ceremade.dauphine.fr/~peyre/numerical-tour/tours/

Thanks
Any questions?