On the Use of Alternating Direction Optimization for Imaging Inverse Problems

Mário A. T. Figueiredo

Instituto de Telecomunicações and Instituto Superior Técnico,
Technical University of Lisbon

PORTUGAL

mario.figueiredo@lx.it.pt, www.lx.it.pt/~mtf

Joint work with José Bioucas-Dias and Manya Afonso

instituto de telecomunicações

INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

UNIVERSIDADE TÉCNICA DE LISBOA

Outline

1. Review of some classical imaging inverse problems

2. Alternating direction method of multipliers (ADMM)
   …for sums of two or more functions.

3. Linear/Gaussian image reconstruction/restoration: SALSA

4. Deblurring Poissonian images: PIDAL

5. Other appllications: structured sparsity, hybrid regularization, …

Regularized Solution of Inverse Problems

Many ill-posed inverse problems are addressed by solving

$$\widehat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) + \tau c(\mathbf{x})$$

$f : \mathbb{R}^n \longrightarrow \mathbb{R}$ data fidelity, observation model, negative log-likelihood, …
usually smooth and convex.

$c : \mathbb{R}^n \longrightarrow \bar{\mathbb{R}}$ regularization/penalty function, or negative log-prior;
typically convex non-differentiable (e.g., for sparsity).

Examples: frame-based signal/image restoration/reconstruction,
sparse representations, compressive sensing, …

A Fundamental Dichotomy: Analysis vs Synthesis

[Elad, Milanfar, Rubinstein, 2007], [Selesnick, F, 2010]

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{A}\mathbf{x}) + \tau c(\mathbf{x})$$

Frame-based "synthesis" regularization

$\mathbf{x}$ contains representation coefficients (not the signal/image itself)

$\mathbf{A} = \mathbf{B}\mathbf{W}$, where $\mathbf{B}$ is the observation operator

$\mathbf{W}$ is a synthesis operator (e.g. of a Parseval frame)

$$\mathbf{W}\mathbf{W}^* = \mathbf{I}$$

typical (sparseness-inducing) regularizer

$$c(\mathbf{x}) = \|\mathbf{x}\|_1$$

proper, lsc, convex (not strictly), and coercive.

A Fundamental Dichotomy: Analysis vs Synthesis

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{A}\mathbf{x}) + \tau c(\mathbf{x})$$

Frame-based "analysis" regularization

$\mathbf{x}$ is the signal/image itself, $\mathbf{A}$ is the observation operator

typical frame-based analysis regularizer:

$$c(\mathbf{x}) = \|\mathbf{P}\,\mathbf{x}\|_1$$

analysis operator (e.g., of a Parseval frame)

$$\mathbf{P}^*\mathbf{P} = \mathbf{I}$$

proper, lsc, convex (not strictly), and coercive.

Image Restoration/Reconstruction: General Formulation

All the previous models written as $$f(\mathbf{x}) = \mathcal{L}(\mathbf{A}\mathbf{x})$$

$$\mathcal{L}(\mathbf{z}) = \sum_{i=1}^{m} \xi(z_i, y_i)$$

where ξ is one (e.g.) of these functions:

Gaussian observations: $$\xi_{\mathrm{G}}(z,y) = \frac{1}{2}(z-y)^2 \longrightarrow \mathcal{L}_{\mathrm{G}}$$

Poissonian observations: $$\xi_{\mathrm{P}}(z,y) = z + \iota_{\mathbb{R}_+}(z) - y\log(z_+) \rightarrow \mathcal{L}_{\mathrm{P}}$$

Multiplicative noise: $$\xi_{\mathrm{M}}(z,y) = L(z + e^{y-z}) \longrightarrow \mathcal{L}_{\mathrm{M}}$$

…all proper, lower semi-continuous (lsc), coercive, convex.

$\mathcal{L}_{\mathrm{G}}$ and $\mathcal{L}_{\mathrm{M}}$ are strictly convex. $\mathcal{L}_{\mathrm{P}}$ is strictly convex if $y_i > 0, \ \forall_i$

Proximity Operators and Iterative Shrinkage/Thresholding

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} f(\mathbf{x}) + \tau c(\mathbf{x})$$

The so-called shrinkage/thresholding/denoising function,

$$\mathrm{prox}_{\tau c}(\mathbf{u}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|_2^2 + \tau c(\mathbf{x})$$

or Moreau proximity operator [Moreau 62], [Combettes 01], [Combettes, Wajs, 05].

Classical case: $c(\mathbf{z}) = \|\mathbf{z}\|_1 \;\Rightarrow\; \mathrm{prox}_{\tau c}(\mathbf{u}) = \mathrm{soft}(\mathbf{u}, \tau)$

IST algorithm:
[F and Nowak, 01, 03],
[Daubechies, Defrise, De Mol, 02, 04],
[Combettes and Wajs, 03, 05],
[Starck, Candés, Nguyen, Murtagh, 2003],

$$\boxed{\mathbf{x}_{k+1} = \mathrm{prox}_{\tau c/\alpha}\left(\mathbf{x}_k - \frac{1}{\alpha}\nabla f(\mathbf{x}_k)\right)}$$

Forward-backward splitting
[Bruck, 1977], [Passty, 1979], [Lions and Mercier, 1979],

Drawbacks of IST

$$\boxed{\mathbf{x}_{k+1} = \operatorname{prox}_{\tau c/\alpha}\left(\mathbf{x}_k - \frac{1}{\alpha}\nabla f(\mathbf{x}_k)\right)}$$

Key condition in convergence proofs: ∇f is Lipschtz

…not true with Poisson or multiplicative noise.

Even for the linear/Gaussian case $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|^2$

…IST is known to be slow when $\mathbf{A}$ is ill-conditioned and/or when τ is very small.

Accelerated versions of IST: Two-step IST (TwIST) [Bioucas-Dias, F, 07]

Fast IST (FISTA) [Beck and Teboulle, 09]

Fixed-point continuation (FPC) [Hale, Yin, and Zhang, 07]

GPSR [F, Nowak, Wright, 07]

SpaRSA [Wright, Nowak, F, 08, 09]

several others…

ADMM: Variable Splitting + Augmented Lagrangian View

Unconstrained (convex) optimization problem: $$\min_{\mathbf{z}\in\mathbb{R}^d} \; f_1(\mathbf{z}) + f_2(\mathbf{G}\,\mathbf{z})$$

$c \times d$

Equivalent constrained problem: $$\min_{\mathbf{z}\in\mathbb{R}^d,\mathbf{u}\in\mathbb{R}^c} \; f_1(\mathbf{z}) + f_2(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{u} - \mathbf{G}\,\mathbf{z} = 0$$

Augmented Lagrangian (AL):

$$L_\mu(\mathbf{z},\mathbf{u},\lambda) = f_1(\mathbf{z}) + f_2(\mathbf{u}) + \lambda^T(\mathbf{G}\mathbf{z}-\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{z}-\mathbf{u}\|_2^2$$

AL method, or method of multipliers (MM) [Hestenes, Powell, 1969]

$$\begin{aligned}(\mathbf{z}_{k+1},\mathbf{u}_{k+1}) &= \arg\min_{\mathbf{z},\mathbf{u}} L_\mu(\mathbf{z},\mathbf{u},\lambda_k)\\ \lambda_{k+1} &= \lambda_k + \mu(\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})\end{aligned}$$

equivalent

ADMM corresponds to minimizing alternatingly w.r.t. $\mathbf{z}$ and $\mathbf{u}$

$$\begin{aligned}(\mathbf{z}_{k+1},\mathbf{u}_{k+1}) &= \arg\min_{\mathbf{z},\mathbf{u}} f_1(\mathbf{z}) + f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z}-\mathbf{u}-\mathbf{d}_k\|_2^2\\ \mathbf{d}_{k+1} &= \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})\end{aligned}$$

e, 2011

Alternating Direction Method of Multipliers (ADMM)

Unconstrained (convex) optimization problem: $$\min_{\mathbf{z}\in\mathbb{R}^d} \; f_1(\mathbf{z}) + f_2(\mathbf{G}\,\mathbf{z})$$

ADMM [Glowinski, Marrocco, 75], [Gabay, Mercier, 76]

$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} f_1(\mathbf{z}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z} - \mathbf{u}_k - \mathbf{d}_k\|^2$$

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|^2$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})$$

Interpretations: variable splitting + augmented Lagrangian + NLBGS;

Douglas-Rachford splitting on the dual [Eckstein, Bertsekas, 92];

split-Bregman approach [Goldstein, Osher, 08]

A *Cornerstone* Result on ADMM

[Eckstein, Bertsekas, 1992]

Consider the problem

$$\min_{\mathbf{z}\in\mathbb{R}^d} \; f_1(\mathbf{z}) + f_2(\mathbf{G}\,\mathbf{z})$$

Let f_1 and f_2 be closed, proper, and convex and $\mathbf{G}$ have full column rank.

Let $(\mathbf{z}_k, \; k = 0, 1, 2, ...)$ be the sequence produced by ADMM, with $\mu > 0$; then, if the problem has a solution, say $\bar{\mathbf{z}}$, then

$$\lim_{k\rightarrow\infty} \mathbf{z}_k = \bar{\mathbf{u}}$$

The theorem also allows for inexact minimizations, as long as the errors are absolutely summable.

ADMM for Two or More Functions

[F and Bioucas-Dias, 09]

Consider a more general problem
$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z}) \qquad (P)$$

$g_j : \mathbb{R}^{p_j} \rightarrow \bar{\mathbb{R}}$ — Proper, closed, convex functions

$\mathbf{H}^{(j)} \in \mathbb{R}^{p_j \times d}$ — Arbitrary matrices

There are many ways to write (P) as $\min_{\mathbf{z}\in\mathbb{R}^d} \; f_1(\mathbf{z}) + f_2(\mathbf{G}\,\mathbf{z})$

We adopt:

$$f_1(\mathbf{z}) = 0, \qquad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix}, \qquad f_2\left(\begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix}\right) = \sum_{j=1}^{J} g_j(\mathbf{u}^{(j)})$$

Another approach in [Goldfarb, Ma, 09, 11]

Applying ADMM to More Than Two Functions

$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z}) \qquad \min_{\mathbf{z}\in\mathbb{R}^d} \; f_2(\mathbf{G}\,\mathbf{z}) \qquad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix}$$

$$\mathbf{z}_{k+1} = \arg\min_{\mathbf{z}} f_1(\mathbf{z}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z} - \mathbf{u}_k - \mathbf{d}_k\|^2$$

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|^2$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})$$

Applying ADMM to More Than Two Functions

$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z}) \qquad \min_{\mathbf{z}\in\mathbb{R}^d} \; f_2(\mathbf{G}\,\mathbf{z}) \qquad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix}$$

$$\mathbf{z}_{k+1} = \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^T \mathbf{H}^{(j)}\right]^{-1} \left(\sum_{j=1}^{J} \mathbf{H}^{(j)} \left(\mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}\right)\right)$$

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|^2$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})$$

Applying ADMM to More Than Two Functions

$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z}) \qquad \min_{\mathbf{z}\in\mathbb{R}^d} \; f_2(\mathbf{G}\,\mathbf{z}) \qquad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix}$$

$$\mathbf{z}_{k+1} = \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^T \mathbf{H}^{(j)}\right]^{-1} \left(\sum_{j=1}^{J} \mathbf{H}^{(j)} \left(\mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}\right)\right)$$

$$\mathbf{u}_{k+1}^{(1)} = \arg\min_{\mathbf{u}} g_1(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(1)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(1)}\|^2$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\mathbf{u}_{k+1}^{(J)} = \arg\min_{\mathbf{u}} g_J(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(J)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(J)}\|^2$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})$$

Applying ADMM to More Than Two Functions

$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z}) \qquad \min_{\mathbf{z}\in\mathbb{R}^d} \; f_2(\mathbf{G}\,\mathbf{z}) \qquad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix}$$

$$\mathbf{z}_{k+1} = \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^T \mathbf{H}^{(j)}\right]^{-1} \left(\sum_{j=1}^{J} \mathbf{H}^{(j)}\left(\mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}\right)\right)$$

$$\mathbf{u}_{k+1}^{(1)} = \arg\min_{\mathbf{u}} g_1(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(1)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(1)}\|^2 = \mathrm{prox}_{g_1/\mu}(\mathbf{H}^{(1)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(j)})$$

$$\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots$$

$$\mathbf{u}_{k+1}^{(J)} = \arg\min_{\mathbf{u}} g_J(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(J)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(J)}\|^2 = \mathrm{prox}_{g_J/\mu}(\mathbf{H}^{(J)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(J)})$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})$$

Applying ADMM to More Than Two Functions

$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z}) \qquad \min_{\mathbf{z}\in\mathbb{R}^d} \; f_2(\mathbf{G}\,\mathbf{z}) \qquad \mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix}$$

$$\mathbf{z}_{k+1} = \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^T \mathbf{H}^{(j)}\right]^{-1} \left(\sum_{j=1}^{J} \mathbf{H}^{(j)} \left(\mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}\right)\right)$$

$$\mathbf{u}_{k+1}^{(1)} = \arg\min_{\mathbf{u}} g_1(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(1)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(1)}\|^2 = \text{prox}_{g_1/\mu}(\mathbf{H}^{(1)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(j)})$$

$$\vdots \qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots$$

$$\mathbf{u}_{k+1}^{(J)} = \arg\min_{\mathbf{u}} g_J(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(J)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(J)}\|^2 = \text{prox}_{g_J/\mu}(\mathbf{H}^{(J)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(J)})$$

$$\mathbf{d}_{k+1}^{(1)} = \mathbf{d}_k^{(1)} - (\mathbf{G}^{(1)}\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(1)})$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\mathbf{d}_{k+1}^{(J)} = \mathbf{d}_k^{(J)} - (\mathbf{G}^{(J)}\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(J)})$$

Applying ADMM to More Than Two Functions

$$\mathbf{z}_{k+1} = \boxed{\left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^* \, \mathbf{H}^{(j)}\right]^{-1}} \left(\sum_{j=1}^{J} (\mathbf{H}^{(j)})^* \left(\mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}\right)\right)$$

$$\mathbf{u}_{k+1}^{(1)} = \arg\min_{\mathbf{u}} g_1(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(1)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(1)}\|^2 = \mathrm{prox}_{g_1/\mu}(\mathbf{H}^{(1)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(j)})$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$\mathbf{u}_{k+1}^{(J)} = \arg\min_{\mathbf{u}} g_J(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{H}^{(J)}\mathbf{z}_{k+1} + \mathbf{d}_k^{(J)}\|^2 = \mathrm{prox}_{g_1/\mu}(\mathbf{H}^{(1)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(j)})$$

$$\mathbf{d}_{k+1}^{(1)} = \mathbf{d}_k^{(1)} - (\mathbf{H}^{(1)}\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(1)})$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$\mathbf{d}_{k+1}^{(J)} = \mathbf{d}_k^{(J)} - (\mathbf{H}^{(J)}\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(J)})$$

Conditions for easy applicability:

inexpensive proximity operators

inexpensive matrix inversion

Linear/Gaussian Observations: Frame-Based Analysis

Problem: $$\widehat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \tau\, \|\mathbf{P}\mathbf{x}\|_1$$

Template: $$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z})$$

Mapping: $J = 2$, $g_1(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{y}\|_2^2$, $g_2(\mathbf{z}) = \tau\, \|\mathbf{z}\|_1$

$$\mathbf{H}^{(1)} = \mathbf{A}, \qquad \mathbf{H}^{(2)} = \mathbf{P},$$

Convergence conditions: g_1 and g_2 are closed, proper, and convex.

$$\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{P} \end{bmatrix}$$ has full column rank.

Resulting algorithm: SALSA

(*split augmented Lagrangian shrinkage algorithm*) [Afonso, Bioucas-Dias and F., 09, 10]

Linear/Gaussian Observations: Frame-Based Analysis

Key steps of SALSA (both for analysis and synthesis):

Moreau proximity operator of $g_1(\mathbf{z}) = \frac{1}{2}\|\mathbf{z}-\mathbf{y}\|_2^2,$

$$\mathrm{prox}_{g_1/\mu}(\mathbf{u}) = \arg\min_{\mathbf{z}} \frac{1}{2\mu}\|\mathbf{z}-\mathbf{y}\|_2^2 + \frac{1}{2}\|\mathbf{z}-\mathbf{u}\|_2^2 = \frac{\mathbf{y}+\mu\,\mathbf{u}}{1+\mu}$$

Moreau proximity operator of $g_2(\mathbf{z}) = \tau\|\mathbf{z}\|_1,$

$$\mathrm{prox}_{g_2/\mu}(\mathbf{u}) = \mathrm{soft}\Big(\mathbf{u}, \tau/\mu\Big)$$

Linear step (next slide):

$$\mathbf{z}_{k+1} = \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^T \mathbf{H}^{(j)}\right]^{-1} \left(\sum_{j=1}^{J} \mathbf{H}^{(j)} \left(\mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}\right)\right)$$

Handling the Matrix Inversion: Frame-Based Analysis

Frame-based analysis: $$\left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^* \mathbf{H}^{(j)}\right]^{-1} = \left[\mathbf{A}^*\mathbf{A} + \mathbf{P}^*\mathbf{P}\right]^{-1} = \left[\mathbf{A}^*\mathbf{A} + \mathbf{I}\right]^{-1}$$

$\mathbf{P}^*\mathbf{P} = \mathbf{I}$

Parseval frame

diagonal

DFT (FFT)

Periodic deconvolution: $\mathbf{A} = \mathbf{U}^*\mathbf{D}\mathbf{U}$

$O(n \log n)$

$$\left[\mathbf{A}^*\mathbf{A} + \mathbf{I}\right]^{-1} = \mathbf{U}^*\left[|\mathbf{D}|^2 + \mathbf{I}\right]^{-1}\mathbf{U}$$

subsampling matrix: $\mathbf{M}\mathbf{M}^* = \mathbf{I}$

Compressive imaging (MRI): $\mathbf{A} = \mathbf{M}\mathbf{U}$

$O(n \log n)$

$$\left[\mathbf{U}^*\mathbf{M}^*\mathbf{M}\mathbf{U} + \mathbf{I}\right]^{-1} = \mathbf{I} - \frac{1}{2}\mathbf{U}^*\mathbf{M}^*\mathbf{M}\mathbf{U}$$

matrix inversion lemma

subsampling matrix: $\mathbf{S}^*\mathbf{S}$ is diagonal

Inpainting (recovery of lost pixels): $\mathbf{A} = \mathbf{S}$

$O(n)$

$\left[\mathbf{S}^*\mathbf{S} + \mathbf{I}\right]^{-1}$ is a diagonal inversion

SALSA for Frame-Based Synthesis

Problem: $\widehat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{BWx} - \mathbf{y}\|_2^2 + \tau \, \|\mathbf{x}\|_1$

Template: $\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{z})$

Mapping: $J = 2$, $\quad g_1(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{y}\|_2^2, \quad g_2(\mathbf{z}) = \tau \, \|\mathbf{z}\|_1$

$$\mathbf{H}^{(1)} = \mathbf{BW} \qquad \mathbf{H}^{(2)} = \mathbf{I},$$

Convergence conditions: g_1 and g_2 are closed, proper, and convex.

$$\mathbf{G} = \begin{bmatrix} \mathbf{B}\,\mathbf{W} \\ \mathbf{I} \end{bmatrix}$$ has full column rank.

Handling the Matrix Inversion: Frame-Based Analysis

Frame-based analysis: $$\left[\sum_{j=1}^{J}(\mathbf{H}^{(j)})^*\,\mathbf{H}^{(j)}\right]^{-1} = \left[\mathbf{W}^*\mathbf{B}^*\mathbf{B}\mathbf{W} + \mathbf{I}\right]^{-1}$$

DFT

Periodic deconvolution: $\mathbf{B} = \mathbf{U}^*\mathbf{D}\mathbf{U}$

diagonal matrix

$O(n \log n)$ $$\left[\mathbf{W}^*\mathbf{B}^*\mathbf{B}\mathbf{W} + \mathbf{I}\right]^{-1} = \mathbf{I} - \mathbf{W}^*\mathbf{U}^*\boxed{\mathbf{D}^*\left[|\mathbf{D}|^2 + \mathbf{I}\right]^{-1}\mathbf{D}}\mathbf{U}\mathbf{W}$$

matrix inversion lemma + $\mathbf{W}\mathbf{W}^* = \mathbf{I}$

subsampling matrix: $\mathbf{M}\mathbf{M}^* = \mathbf{I}$

Compressive imaging (MRI): $\mathbf{B} = \mathbf{M}\mathbf{U}$

$O(n \log n)$ $$\left[\mathbf{W}^*\mathbf{U}^*\mathbf{M}^*\mathbf{M}\mathbf{U}\mathbf{W} + \mathbf{I}\right]^{-1} = \mathbf{I} - \frac{1}{2}\mathbf{W}^*\mathbf{U}^*\mathbf{M}^*\mathbf{M}\mathbf{U}\mathbf{W}$$

subsampling matrix: $\mathbf{S}\mathbf{S}^* = \mathbf{I}$

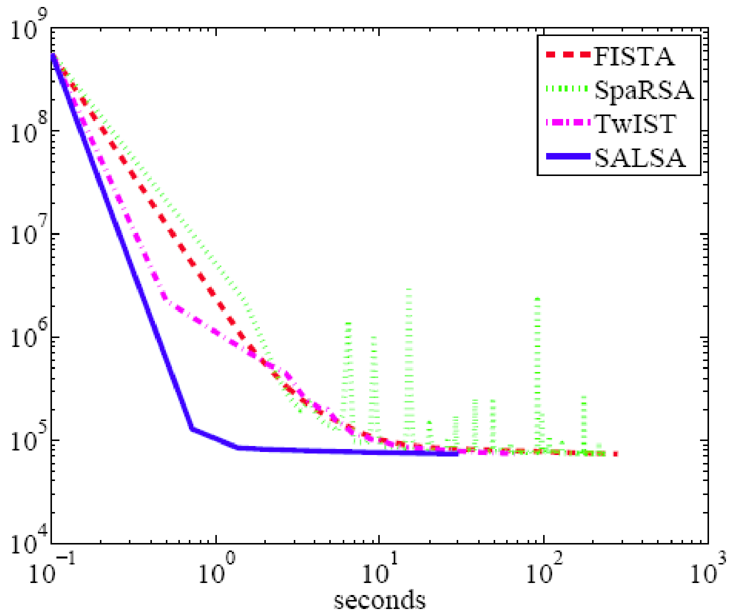
Inpainting (recovery of lost pixels): $\mathbf{B} = \mathbf{S}$

$O(n \log n)$ $$\left[\mathbf{W}^*\mathbf{S}^*\mathbf{S}\mathbf{W} + \mathbf{I}\right]^{-1} = \mathbf{I} - \frac{1}{2}\mathbf{W}^*\mathbf{S}^*\mathbf{S}\mathbf{W}^*$$

SALSA Experiments

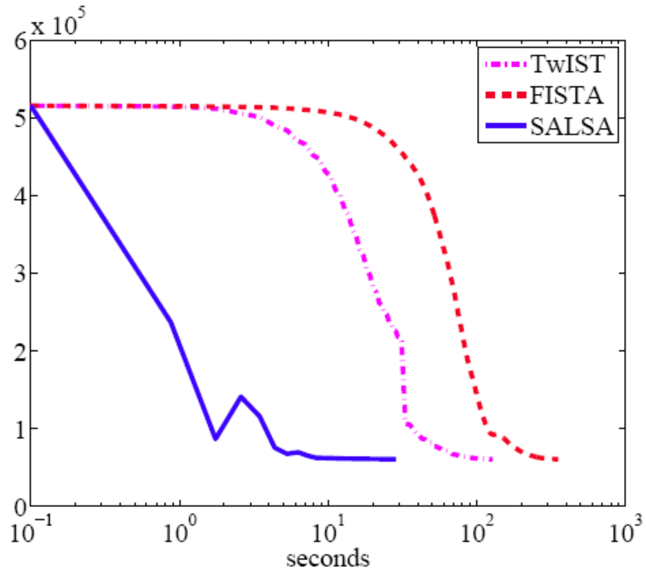
Benchmark problem: image deconvolution (9x9 uniform blur, 40dB BSNR)

undecimated Haar wavelets, ℓ_1 synthesis regularization.



SALSA Experiments

Image inpainting (50% pixels missing)



Alg.	Calls to $\mathbf{B}, \mathbf{B}^H$	Iter.	CPU time (sec.)	MSE MSE	ISNR (dB)
FISTA	1022	340	263.8	92.01	18.96
TwIST	271	124	112.7	100.92	18.54
SALSA	84	28	20.88	77.61	19.68

Frame-Based Analysis Deconvolution of Poissonian Images

Problem template: $$\min_{\mathbf{u}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{u}) \qquad (P1)$$

positivity constraint

Frame-analysis regularization: $$\widehat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \mathcal{L}_{\mathrm{P}}(\mathbf{B}\,\mathbf{x}) + \lambda\|\mathbf{P}\,\mathbf{x}\|_1 + \iota_{\mathbb{R}^n_+}(\mathbf{x})$$

Same form as $(P1)$ with: $J = 3, \;\; g_1 = \mathcal{L}_{\mathrm{P}}, \;\; g_2 = \|\cdot\|_1, \;\; g_3 = \iota_{\mathbb{R}^n_+}$

Convergence conditions: g_1, g_2, and g_3 are closed, proper, and convex.

$$\mathbf{G} = \begin{bmatrix} \mathbf{B} \\ \mathbf{P} \\ \mathbf{I} \end{bmatrix}$$ has full column rank

Required inversion: $$\left[\sum_{j=1}^{J}(\mathbf{H}^{(j)})^*\,\mathbf{H}^{(j)}\right]^{-1} = \left[\mathbf{B}^*\mathbf{B} + \mathbf{P}^*\mathbf{P} + \mathbf{I}\right]^{-1} = \left[\mathbf{B}^*\mathbf{B} + 2\mathbf{I}\right]^{-1}$$

…again, easy in deconvolution, inpainting, **tomography**.

Proximity Operator of the Poisson Log-Likelihood

Proximity operator of the Poisson log-likelihood

$$\mathbf{u}_{k+1}^{(1)} \leftarrow \arg\min_{\mathbf{v}} \frac{\mu}{2}\|\mathbf{v} - \boldsymbol{\nu}_k^{(1)}\|_2^2 + \sum_{i=1}^{m} \xi(v_i, y_i)$$

$$\xi(z, y) = z + \iota_{\mathbb{R}_+}(z) - y\log(z_+)$$

Separable problem with closed-form (non-negative) solution

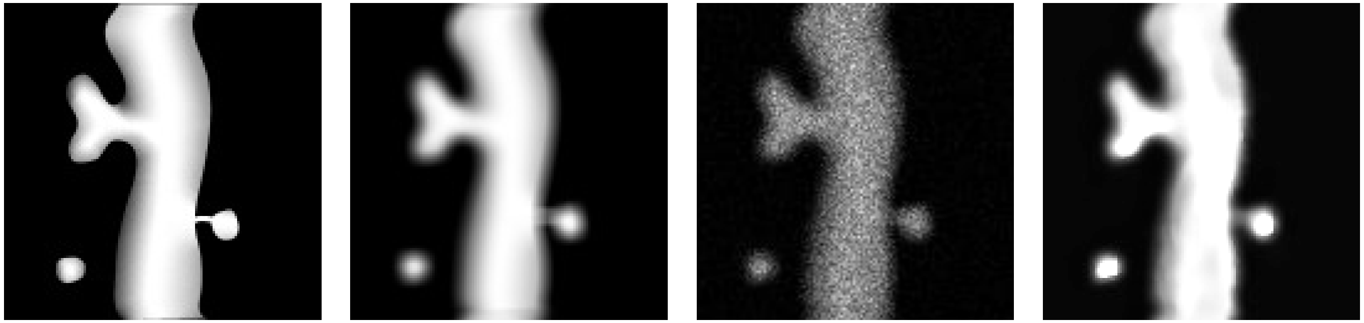
$$u_{i,k+1}^{(1)} = \frac{1}{2}\left(\nu_{i,k}^{(1)} - \frac{1}{\mu} + \sqrt{\left(\nu_{i,k}^{(1)} - \frac{1}{\mu}\right)^2 + \frac{4\,y_i}{\mu}}\right)$$

Proximity operator of $g_3 = \iota_{\mathbb{R}_+^n}$ is simply $\mathrm{prox}_{\iota_{\mathbb{R}_+^n}}(\mathbf{x}) = (\mathbf{x})_+$

Poisson Image Deconvolution by AL (PIDAL): Experiments

Comparison with [Dupé, Fadili, Starck, 09] and [Starck, Bijaoui, Murtagh, 95]

		PIDAL-TV			PIDAL-FA			[Dupé, Fadili, Starck, 09]			[Starck et al, 95]
Image	M	MAE	iterations	time	MAE	iterations	time	MAE	iterations	time	MAE
Cameraman	5	0.27	120	22	**0.26**	70	13	0.35	6	4.5	0.37
Cameraman	30	1.29	51	9.1	**1.22**	39	7.4	1.47	98	75	2.06
Cameraman	100	3.99	33	6.0	**3.63**	36	6.8	4.31	426	318	5.58
Cameraman	255	8.99	32	5.8	**8.45**	37	7.0	10.26	480	358	12.3
Neuron	5	**0.17**	117	3.6	0.18	66	2.9	0.19	6	3.9	0.19
Neuron	30	**0.68**	54	1.8	0.77	44	2.0	0.82	161	77	0.95
Neuron	100	**1.75**	43	1.4	2.04	41	1.8	2.32	427	199	2.88
Neuron	255	3.52	43	1.4	**3.47**	42	1.9	5.25	202	97	6.31
Cell	5	0.12	56	10	**0.11**	36	7.6	0.12	6	4.5	0.12
Cell	30	0.57	31	6.5	0.54	39	8.2	0.56	85	64	**0.47**
Cell	100	1.71	85	15	1.46	31	6.4	1.72	215	162	**1.37**
Cell	255	3.77	89	17	3.33	34	7.0	5.45	410	308	**3.10**



$$\text{MAE} \equiv \frac{\|\widehat{\mathbf{x}} - \mathbf{x}\|_1}{n}$$

Constrained Optimization Formulation

Unconstrained optimization formulation: $$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \tau c(\mathbf{x})$$

Constrained optimization formulation: $$\min_{\mathbf{x}} \quad c(\mathbf{x})$$

basis pursuit denoising (BPN) $$\text{s.t.} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \leq \varepsilon$$

[Chen, Donoho, Saunders, 1998]

Both analysis and synthesis can be used:

- frame-based analysis, $$c(\mathbf{x}) = \|\mathbf{P}\mathbf{x}\|_1$$
- frame-based synthesis $$c(\mathbf{x}) = \|\mathbf{x}\|_1$$

$$\mathbf{A} = \mathbf{B}\,\mathbf{W}$$

Proposed Approach for Constrained Formulation

Constrained problem: $$\min_{\mathbf{x}} \; c(\mathbf{x}) \quad \text{s.t. } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \le \varepsilon$$

…can be written as $$\min_{\mathbf{x}} \; c(\mathbf{x}) + \iota_{E(\varepsilon,\mathbf{y})}(\mathbf{A}\,\mathbf{x})$$

$$E(\varepsilon, \mathbf{y}) = \{\mathbf{x} \in \mathbb{R}^n \; : \; \|\mathbf{x} - \mathbf{y}\|_2 \le \varepsilon\} \qquad \iota_S(z) = \begin{cases} 0 & \Leftarrow \quad z \in S \\ +\infty & \Leftarrow \quad z \notin S \end{cases}$$

…which has the form $$\min_{\mathbf{u}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{u}) \qquad (P1)$$

with $J = 2, \quad g_1(\mathbf{z}) = c(\mathbf{z}), \qquad \mathbf{H}^{(1)} = \mathbf{I} \qquad \mathbf{G} = \begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix}$

$g_2(\mathbf{z}) = \iota_{E(\varepsilon,\mathbf{y})}(\mathbf{z}), \qquad \mathbf{H}^{(2)} = \mathbf{A}$

full column rank

Resulting algorithm: C-SALSA (constrained-SALSA) [Afonso, Bioucas-Dias, F, 09, 11]

Some Aspects of C-SALSA

Moreau proximity operator of $\iota_{E(\varepsilon,\mathbf{y})}$ is simply a projection on an ℓ_2 ball:

$$\operatorname{prox}_{\iota_{E(\varepsilon,\mathbf{y})}}(\mathbf{u}) = \arg\min_{\mathbf{z}} \iota_{E(\varepsilon,\mathbf{y})} + \frac{1}{2}\|\mathbf{z}-\mathbf{u}\|_2^2$$

$$= \begin{cases} \mathbf{u} & \Leftarrow \quad \|\mathbf{u}-\mathbf{y}\|_2 \leq \varepsilon \\ \mathbf{y} + \frac{\varepsilon(\mathbf{u}-\mathbf{y})}{\|\mathbf{u}-\mathbf{y}\|_2} & \Leftarrow \quad \|\mathbf{u}-\mathbf{y}\|_2 > \varepsilon \end{cases}$$

As SALSA, also C-SALSA involves inversion of the form

$$\left[\mathbf{W}^*\mathbf{B}^*\mathbf{B}\mathbf{W} + \mathbf{I}\right]^{-1} \quad \text{or} \quad \left[\mathbf{A}^*\mathbf{A} + \mathbf{P}^*\mathbf{P}\right]^{-1}$$

…all the same tricks as above.

C-SALSA Experiments: Image Deblurring

Benchmark experiments:

Experiment	blur kernel	σ^2
1	9×9 uniform	0.56^2
2A	Gaussian	2
2B	Gaussian	8
3A	$h_{ij} = 1/(1+i^2+j^2)$	2
3B	$h_{ij} = 1/(1+i^2+j^2)$	8

Comparison with

NESTA [Bobin, Becker, Candès, 09]

SPGL1 [van den Berg, Friedlander, 08]

Frame-synthesis

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)			Iterations			CPU time (seconds)		
	SPGL1	NESTA	C-SALSA	SPGL1	NESTA	C-SALSA	SPGL1	NESTA	C-SALSA
1	1029 (659/1290)	3520 (3501/3541)	398 (388/406)	340	880	134	441.16	590.79	100.72
2A	511 (279/663)	4897 (4777/4981)	451 (442/460)	160	1224	136	202.67	798.81	98.85
2B	377 (141/532)	3397 (3345/3473)	362 (355/370)	98	849	109	120.50	557.02	81.69
3A	675 (378/772)	2622 (2589/2661)	172 (166/175)	235	656	58	266.41	423.41	42.56
3B	404 (300/475)	2446 (2401/2485)	134 (130/136)	147	551	41	161.17	354.59	29.57

Frame-analysis

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)		Iterations		CPU time (seconds)	
	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA
1	2881 (2861/2889)	413 (404/419)	720	138	353.88	80.32
2A	2451 (2377/2505)	362 (344/371)	613	109	291.14	62.65
2B	2139 (2065/2197)	290 (278/299)	535	87	254.94	50.14
3A	2203 (2181/2217)	137 (134/143)	551	42	261.89	23.83
3B	1967 (1949/1985)	116 (113/119)	492	39	236.45	22.38

TV

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)		Iterations		CPU time (seconds)	
	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA
1	7783 (7767/7795)	695 (680/710)	1945	232	311.98	62.56
2A	7323 (7291/7351)	559 (536/578)	1830	150	279.36	38.63
2B	6828 (6775/6883)	299 (269/329)	1707	100	265.35	25.47
3A	6594 (6513/6661)	176 (98/209)	1649	59	250.37	15.08
3B	5514 (5417/5585)	108 (104/110)	1379	37	210.94	9.23

(Overlapping) Group Regularization (Structured Sparsity)

$$\widehat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i=1}^{k} \lambda_i\, \phi_i(\mathbf{x}_{G_i})$$

$G_i \subseteq \{1, ..., n\}$
groups (may overlap)

If groups have hierarchical structure and the ϕ_i are ℓ_1, ℓ_2, or ℓ_∞ norms, then $\mathrm{prox}_{\sum_i \phi_i}$ can be computed efficiently [Jenatton, Audibert, Bach, 2009]

Algorithm for arbitrary groups with $\phi_i = \|\cdot\|_2$ (FoGLASSO) [Liu and Ye, 2010]

ADMM allows addressing this problem, if…
[F and Bioucas-Dias, SPARS'2011]

…the functions ϕ_i have simple prox_{ϕ_i}

…a certain matrix inversion can be efficiently handled

See [Qin and Goldfarb, 2011] for another ADM method for this problem.

(Overlapping) Group Regularization

$$\widehat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 + \sum_{i=1}^{k} \lambda_i\, \phi_i(\mathbf{x}_{G_i})$$

Template: $$\min_{\mathbf{u}} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{u})$$

Mapping: $J = k+1,$

$$g_1(\mathbf{z}) = \frac{1}{2}\|\mathbf{z}-\mathbf{y}\|_2^2,$$

$$g_j(\mathbf{z}) = \lambda_j\, \phi_j(\mathbf{z}),\ j = 2, ..., k+1$$

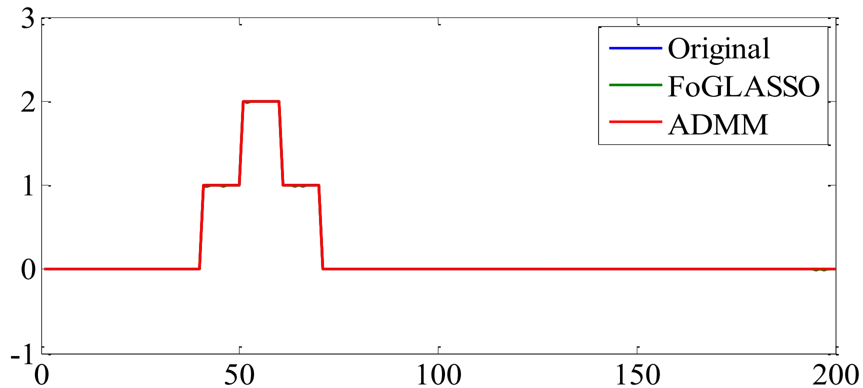
$$\mathbf{H}^{(1)} = \mathbf{A},$$

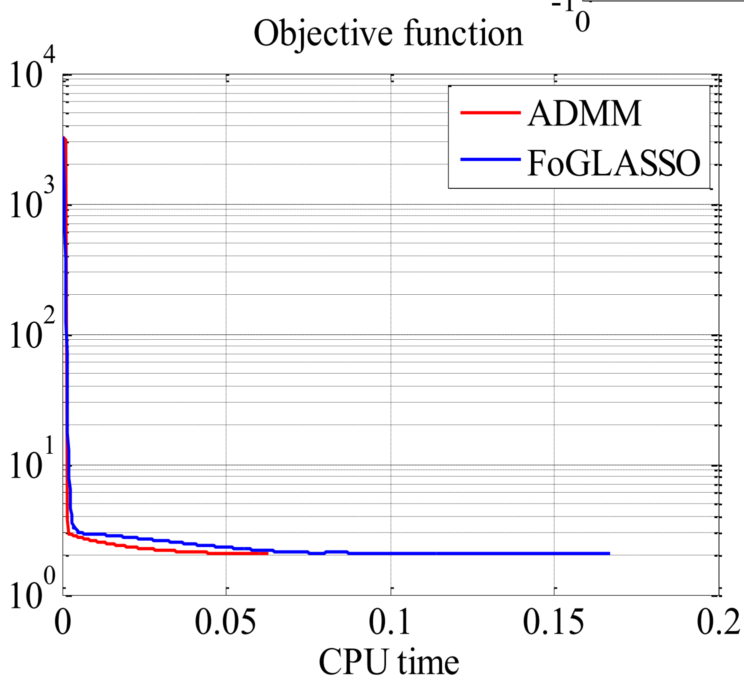
$$\mathbf{H}^{(j)} = \mathrm{diag}(\mathbb{I}_{1\in G_j}, \mathbb{I}_{2\in G_j}, ..., \mathbb{I}_{n\in G_j}),\ j = 2, ..., k+1$$

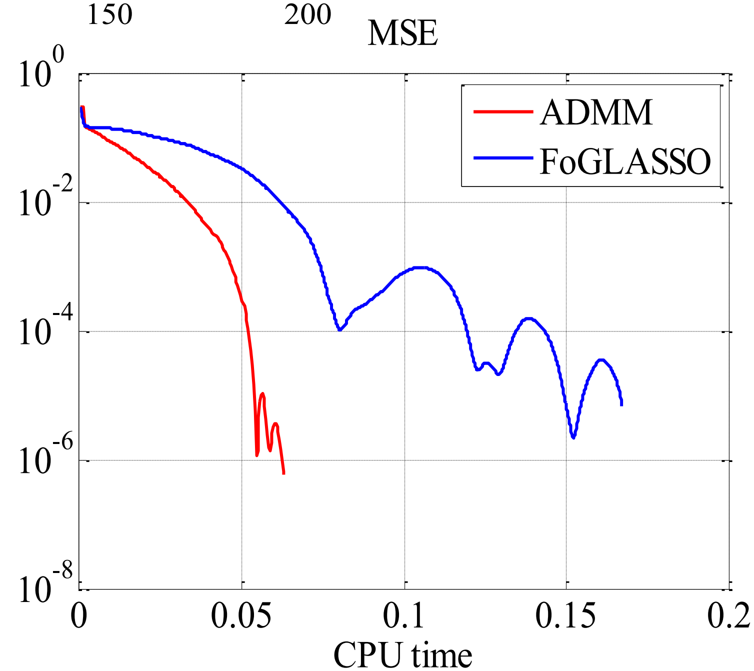
(Overlapping) Group Regularization: Toy Example

$$n = 200,\ \mathbf{A} \in \mathbb{R}^{100\times 200}\ (\text{i.i.d.}\, \mathcal{N}(0,1)) \qquad \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n},\ \ \mathbf{n} \sim \mathcal{N}(0, 10^{-2})$$

$$\phi_i = \|\cdot\|_2,\ k = 19,\ G_1 = \{1, .., 20\},\ G_2 = \{11, ..., 30\}, ...$$







Hybrid: Analysis + Synthesis Regularization

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{B}\,\mathbf{W}\,\mathbf{x} - \mathbf{y}\|_2^2 + \tau_1\,\|\mathbf{x}\|_1 + \tau_2\|\mathbf{P}\,\mathbf{W}\,\mathbf{x}\|_1$$

Observation matrix

synthesis matrix
of a Parseval frame

analysis matrix of
another Parseval frame

As in frame-based “synthesis” regularization,

$\mathbf{x}$ contains representation coefficients (not the signal itself)

these coefficients are under regularization

As in frame-based “analysis” regularization,

the signal $\mathbf{z} = \mathbf{W}\,\mathbf{x}$ is “analyzed”: $\mathbf{P}\,\mathbf{z}$

the result of the analysis is under regularization

Hybrid: Analysis + Synthesis Regularization

Problem: $$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{B}\,\mathbf{W}\,\mathbf{x} - \mathbf{y}\|_2^2 + \tau_1\,\|\mathbf{x}\|_1 + \tau_2\|\mathbf{P}\,\mathbf{W}\,\mathbf{x}\|_1$$

Template: $$\min_{\mathbf{u}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{u}) \qquad (P1)$$

Mapping: $J = 3$, $\quad g_1(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{y}\|_2^2,$ $\quad g_2(\mathbf{z}) = \tau_1\,\|\mathbf{z}\|_1$, $\quad g_3(\mathbf{z}) = \tau_2\,\|\mathbf{z}\|_1$

$$\mathbf{H}^{(1)} = \mathbf{B}\,\mathbf{W}, \quad \mathbf{H}^{(2)} = \mathbf{I}, \quad \mathbf{H}^{(3)} = \mathbf{P}\,\mathbf{W},$$

Convergence conditions: all g_i are closed, proper, and convex.

$$\mathbf{G} = \begin{bmatrix} \mathbf{B}\,\mathbf{W} \\ \mathbf{I} \\ \mathbf{P}\,\mathbf{W} \end{bmatrix}$$ has full column rank.

Experiments: Image Deconvolution

Benchmark experiments:

Experiment	blur kernel	σ^2
1	9×9 uniform	0.56^2
2A	Gaussian	2
2B	Gaussian	8
3A	$h_{ij} = 1/(1 + i^2 + j^2)$	2
3B	$h_{ij} = 1/(1 + i^2 + j^2)$	8

Two different frames (undecimated Daubechies 2 and 6); hand-tuned parameters.

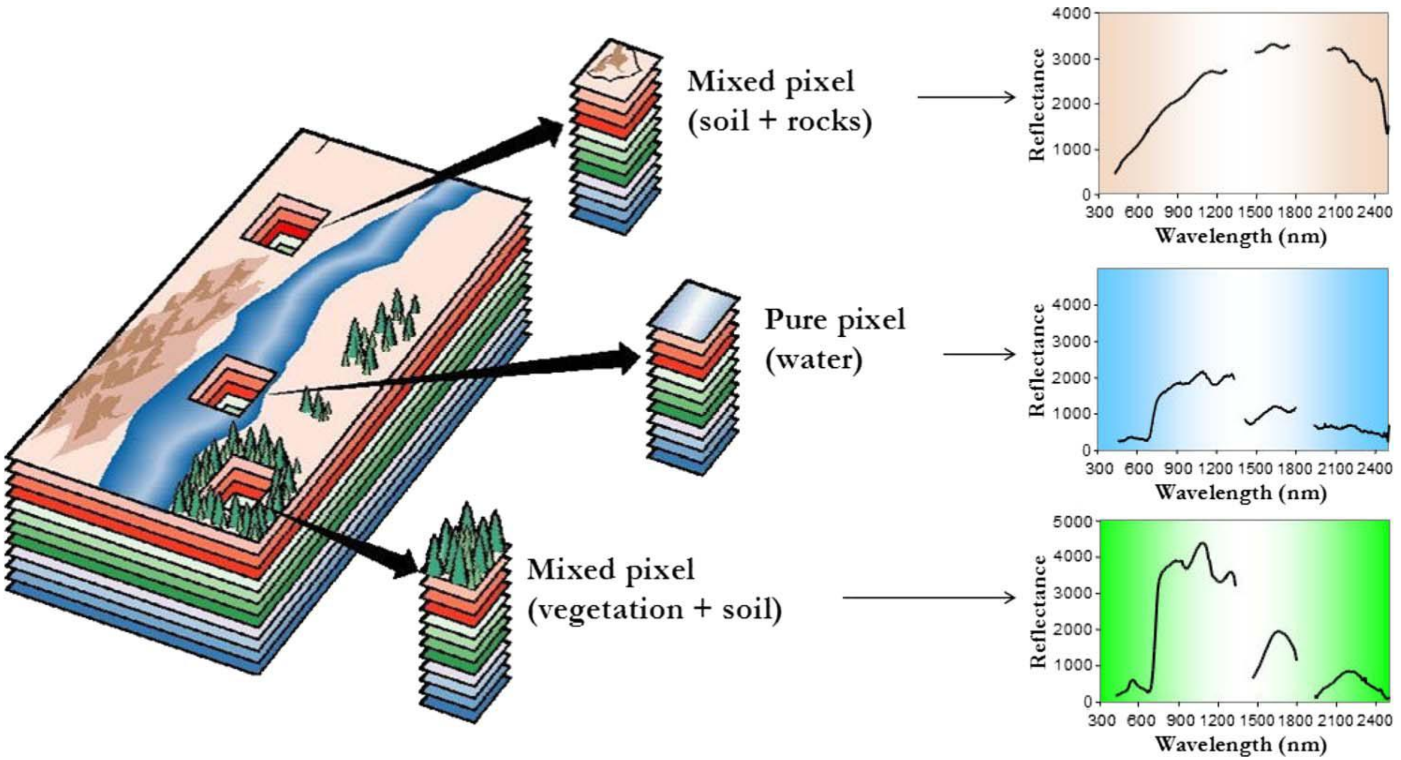
Exp.	Analysis ISNR	Synthesis ISNR	Hybrid ISNR	Analysis time (sec)	Synthesis time (sec)	Hybrid time (sec)
1	8.52 dB	7.13 dB	8.61 dB	34.1	4.1	11.1
2A	5.38 dB	4.49 dB	5.48 dB	21.3	1.4	3.8
2B	5.27 dB	4.48 dB	5.39 dB	20.2	1.6	3.4
3A	7.33 dB	6.32 dB	7.46 dB	17.9	1.6	3.7
3B	4.93 dB	4.37 dB	5.31 dB	20.1	2.5	3.9

Preliminary conclusions: analysis is better than synthesis

hybrid is slightly better than analysis

hybrid is faster than analysis

Yet Another Application: Spectral Unmixing



Goal: find the relative abundance of each "material" in each pixel.

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\,\mathbf{x} - \mathbf{y}\|_2^2 + \boxed{\iota_{\mathbb{R}^n_+}(\mathbf{x}) + \iota_{\{1\}}(\mathbf{1}^T\mathbf{x})}$$

Given library of spectra

Indicator of the canonical simplex

Spectral Unmixing

Problem: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\,\mathbf{x} - \mathbf{y}\|_2^2 + \iota_{\mathbb{R}_+^n}(\mathbf{x}) + \iota_{\{1\}}(\mathbf{1}^T\mathbf{x})$

Template: $\min_{\mathbf{u}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{u})$

Mapping: $J = 3$, $g_1(\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - \mathbf{y}\|_2^2$, $g_2(\mathbf{z}) = \iota_{\mathbb{R}_+^n}(\mathbf{z})$, $g_3(z) = \iota_{\{1\}}(z)$

$$\mathbf{H}^{(1)} = \mathbf{A}, \qquad \mathbf{H}^{(2)} = \mathbf{I}, \qquad \mathbf{H}^{(3)} = \mathbf{1}^T$$

Proximity operators are trivial.

Matrix inversion can be precomputed (typical sizes 200~300 x 500~1000)

$$\left[\sum_{j=1}^{J}(\mathbf{H}^{(j)})^*\,\mathbf{H}^{(j)}\right]^{-1} = \left[\mathbf{A}^T\mathbf{A} + \mathbf{I} + \mathbf{1}\mathbf{1}^T\right]^{-1}$$

Spectral unmixing by split augmented Lagrangian (SUNSAL) [Bioucas-Dias, F, 2010]

Related algorithm (split-Bregman view) in [Szlam, Guo, Osher, 2010]

Summary, Open Questions, and Ongoing Work

Summary: ADMM is a very flexible and efficient tool, for a variety of optimization problems arising in imaging inverse problems…

…if a certain matrix can be cheaply inverted.

Ongoing work: efficiently handling (large) problems where the matrix inversion can't be sidestepped (L-BFGS [Afonso, Bioucas-Dias, F, 2010]) (alternating linearization [Goldfarb, Ma, Scheinberg, 2010], primal-dual methods [Chambolle, Pock, 2009], [Esser, Zhang, Chan, 2009])

hyperspectral imaging with spatial regularization

(overlapping) group regularization
[F, Bioucas-Dias, SPARS'11]

MAP inference in graphical models (dual decomposition + ADMM)
[Martins, Smith, Xing, Aguiar, F, ICML'2011]

logistic regression, …

Some Publications

M. Figueiredo and J. Bioucas-Dias, “Restoration of Poissonian images using alternating direction optimization”, *IEEE Transactions on Image Processing*, 2010.

J. Bioucas-Dias and M. Figueiredo, ““Multiplicative noise removal using variable splitting and constrained optimization”, *IEEE Transactions on Image Processing*, 2010.

M. Afonso, J. Bioucas-Dias, M. Figueiredo, “An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems”, *IEEE Transactions on Image Processing*, 2011.

M. Afonso, J. Bioucas-Dias, M. Figueiredo, “Fast image recovery using variable splitting and constrained optimization”, *IEEE Transactions on Image Processing*, 2010.

M. Afonso, J. Bioucas-Dias, M. Figueiredo, “An augmented Lagrangian approach to linear inverse problems with compound regularization”, *IEEE International Conference on Image Processing – ICIP’2010*, Hong Kong, 2010.

A. Martins, M. Figueiredo, N. Smith, E. Xing, P. Aguiar, “An augmented Lagrangian approach to constrained MAP inference”, *International Conference on Machine Learning – ICML’2011, Bellevue, WA,* 2011.

J. Bioucas-Dias, M. Figueiredo, “Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing”, *Workshop on Hyperspectral Image and Signal Processing: evolution in Remote Sensing - WHISPERS’2010*, Reykjavik, Iceland, 2010.

M. Figueiredo, J. Bioucas-Dias, “An alternating direction algorithm for (overlapping) group regularization”, *Workshop on Signal Processing with Adaptive Sparse Structured Representations – SPARS’2011*, Edinburgh, UK, 2011.