Why CLEAN when you can PURIFY? a new algorithmic framework for next-generation radio-interferometric imaging

Rafael E. Carrillo

Signal Processing Laboratory (LTS5)
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Joint work with J. D. McEwen and Y. Wiaux

BASP Frontiers January 27, 2015

Outline

Outline

Introduction

Scalable Algorithms

Numerical Results

Summary

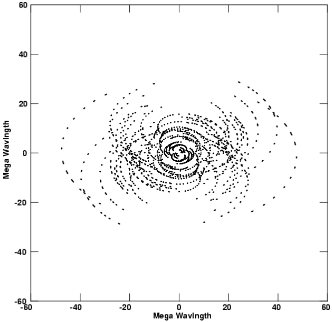
Introduction

- Astronomical studies require high resolution, high sensitivity imaging devices
- A radio interferometer is an array of spatially separated antennas that takes measurements of the radio emissions of the sky
- It allows observation of the radio emission from the sky with high angular resolution and sensitivity

RI Inverse Problem

Interferometers provide incomplete Fourier measurements of the observed object (complex visibilities)

$$y(\mathbf{u}) = \int A(\mathbf{l}, \mathbf{u})\, x(\mathbf{l})\, \mathrm{e}^{-2\mathrm{i}\pi \mathbf{u}\cdot\mathbf{l}}\, \mathrm{d}^2\mathbf{l}$$



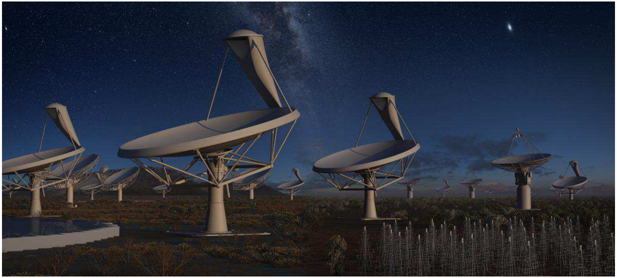
- $A(\mathbf{l}, \mathbf{u})$: direction dependent effects

Image recovery poses a linear inverse problem:

$$\mathbf{y} = \Phi\mathbf{x}, \text{ with } \Phi \in \mathbb{C}^{M\times N}$$

Next Generation Instruments

Next generation telescopes, such as the SKA, have triggered an intense research to reformulate imaging techniques for radio interferometry.



Motivation

Main challenges for next generation telescopes

- High resolution and dynamic range
- Large number of continuous visibilities (M orders of magnitude larger than N)

Motivation

Main challenges for next generation telescopes

- High resolution and dynamic range
- Large number of continuous visibilities (M orders of magnitude larger than N)

Our solution

- Leverage recent advances in sparse signal recovery and convex optimization to address these challenging problems
- Effectiveness of sparse regularization applied to radio interferometric imaging already demonstrated (Wiaux et al. 2009, Wenger et al. 2010, McEwen & Wiaux 2011, Li et al. 2011, Carrillo et al. 2012, Hardy 2013, Garsden et al. 2014)

Sparse Signal Recovery

- Suppose $\mathbf{x}$ is expressed in terms of a dictionary $\Psi \in \mathbb{C}^{N \times D}$, $D \geq N$, as $\mathbf{x} = \Psi \boldsymbol{\alpha}$, $\boldsymbol{\alpha} \in \mathbb{C}^D$
- Noisy model:

$$\mathbf{y} = \Phi \mathbf{x} + \mathbf{n}$$

- Two different approaches
  - Synthesis based problem:

$$\min_{\bar{\boldsymbol{\alpha}} \in \mathbb{R}^N} \|\bar{\boldsymbol{\alpha}}\|_1 \text{ subject to } \|\mathbf{y} - \Phi \Psi \bar{\boldsymbol{\alpha}}\|_2 \leq \epsilon$$

  - Analysis based problem:

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}^N} \|\Psi^{\dagger} \bar{\mathbf{x}}\|_1 \text{ subject to } \|\mathbf{y} - \Phi \bar{\mathbf{x}}\|_2 \leq \epsilon$$

Average Sparsity

- We recently propose the SARA algorithm based on the average sparsity model
- It uses a dictionary composed of several coherent frames:

$$\Psi = [\Psi_1, \Psi_2, \ldots, \Psi_q]$$

- Optimization problem:

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}_+^N} \|\Psi^\dagger \bar{\mathbf{x}}\|_0 \text{ subject to } \|\mathbf{y} - \Phi \bar{\mathbf{x}}\|_2 \leq \epsilon$$

$$\|\Psi^\dagger \bar{\mathbf{x}}\|_0 = \sum_{i=1}^{q} \|\Psi_i^\dagger \bar{\mathbf{x}}\|_0 \rightarrow \text{average sparsity}$$

- A reweighting scheme solving a sequence of (convex) weighted ℓ_1-problems is used to approximate the ℓ_0 problem

Constrained Optimization

Thus we focus on solving problems of the form:

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}_+^N} \|\mathsf{W}\Psi^\dagger \bar{\mathbf{x}}\|_1 \text{ subject to } \|\mathbf{y} - \Phi \bar{\mathbf{x}}\|_2 \leq \epsilon$$

- $\epsilon = \sigma_n \sqrt{M + 2\sqrt{M}}$ →statistical bound
- $\bar{\mathbf{x}} \in \mathbb{R}_+^N$ →positivity constraint
- $\Phi = \mathsf{GFDA}$
  - G : convolutional interpolation operator
  - F : fast Fourier transform
  - D : deconvolution operator
  - A : primary beam

Outline

Proximal Splitting Methods

- Solve problems of the form

$$\min_{\mathbf{x}\in\mathbb{R}^N} f_1(\mathbf{x}) + \ldots + f_S(\mathbf{x})$$

- $f_1(\mathbf{x}), \ldots, f_S(\mathbf{x})$ are proper convex lower semicontinuous functions from $\mathbb{R}^N$ to $\mathbb{R}$ (not necessarily differentiable)
- Key idea: split a complicated problem into several simpler problems
- Each non-smooth function is incorporated in the optimization via its proximity operator:

$$\mathrm{prox}_f(\mathbf{x}) \triangleq \arg\min_{\mathbf{z}\in\mathbb{R}^N} f(\mathbf{z}) + \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|_2^2$$

Solving the Weighted ℓ_1 Problem

The ℓ_1 problem can be reformulated as:

$$\min_{\mathbf{x}\in\mathbb{R}^N} f_1(\mathsf{L}_1\mathbf{x}) + \ldots + f_S(\mathsf{L}_S\mathbf{x})$$

with $S = 3$

- $\mathsf{L}_1 = \Psi^H$, $\mathsf{L}_2 = \mathsf{I}$ and $\mathsf{L}_3 = \Phi$
- $f_1(\mathbf{r}_1) = \|\mathsf{W}\mathbf{r}_1\|_1$ for $\mathbf{r}_1 \in \mathbb{R}^D$
- $f_2(\mathbf{r}_2) = i_C(\mathbf{r}_2)$ with $C = \mathbb{R}^N_+$
- $f_3(\mathbf{r}_k) = i_B(\mathbf{r}_3)$ with $B = \{\mathbf{r}_3 \in \mathbb{R}^M : \|\mathbf{y} - \mathbf{r}_3\|_2 \leq \epsilon\}$

Simultaneous-Direction Method of Multipliers (SDMM)

SDMM uses the following equivalent problem

$$\min f_1(\mathbf{r}_1) + \ldots + f_S(\mathbf{r}_S)$$
$$\text{subject to } \mathsf{L}_k \mathbf{x} = \mathbf{r}_k, \text{ for } k = 1, \ldots, S$$

- SDMM decouples the problems for $f_1, \ldots, f_S$, offering a parallel algorithmic structure
- Subproblems optimizing $f_1, \ldots, f_S$ no longer involve linear operators
- Optimization based on an alternate primal-dual approach

Alternating Minimization Approach

SDMM uses the augmented Lagragian function

$$\mathcal{L}_\gamma(\mathbf{x},\mathbf{r}_1,\ldots,\mathbf{r}_S,\mathbf{z}_1,\ldots,\mathbf{z}_S) = \sum_{i=1}^{S} f_i(\mathbf{r}_i) + \frac{1}{\gamma}\mathbf{z}_i^H(\mathsf{L}_i\mathbf{x}-\mathbf{r}_i) + \frac{1}{2\gamma}\|\mathsf{L}_i\mathbf{x}-\mathbf{r}_i\|_2^2,$$

and then solves for each variable in an alternating fashion:

$$\mathbf{x}^{(t)} = \arg\min_{\mathbf{x}} \mathcal{L}_\gamma(\mathbf{x},\mathbf{r}_1^{(t-1)},\ldots,\mathbf{r}_S^{(t-1)},\mathbf{z}_1^{(t-1)},\ldots,\mathbf{z}_S^{(t-1)})$$

$$\mathbf{r}_i^{(t)} = \arg\min_{\mathbf{r}_i} \mathcal{L}_\gamma(\mathbf{x}^{(t)},\mathbf{r}_1,\ldots,\mathbf{r}_S,\mathbf{z}_1^{(t-1)},\ldots,\mathbf{z}_S^{(t-1)})$$

$$\mathbf{z}_i^{(t)} = \mathbf{z}_i^{(t-1)} + \mathsf{L}_i\mathbf{x}^{(t)} - \mathbf{r}_i^{(t)}$$

Scalability to High Data Dimensions

- Large-scale data problems, i.e. $M \gg N$ and large N
- Partition $\mathbf{y}$ and Φ into R blocks:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_R \end{bmatrix} \text{ and } \Phi = \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_R \end{bmatrix}$$

- Each $\mathbf{y}_i$ is modeled as $\mathbf{y}_i = \Phi_i \mathbf{x} + \mathbf{n}_i$
- Reconstruction problem reformulated as

$$\min_{\bar{\mathbf{x}} \in \mathbb{R}_+^N} \|\mathrm{W}\Psi^H \bar{\mathbf{x}}\|_1 \text{ subject to } \|\mathbf{y}_i - \Phi_i \bar{\mathbf{x}}\|_2 \leq \epsilon_i, i = 1, \ldots, R$$

Problem Reformulation

The ℓ_1 problem can be reformulated as:

$$\min_{\mathbf{x}\in\mathbb{R}^N} f_1(\mathsf{L}_1\mathbf{x}) + \cdots + f_S(\mathsf{L}_S\mathbf{x})$$

with $S = R + 2$

- $\mathsf{L}_1 = \Psi^H$, $\mathsf{L}_2 = \mathsf{I}$ and $\mathsf{L}_{k+2} = \Phi_k$ for $k = 1, \ldots, S$
- $f_1(\mathbf{r}_1) = \|\mathsf{W}\mathbf{r}_1\|_1$ for $\mathbf{r}_1 \in \mathbb{R}^D$
- $f_2(\mathbf{r}_2) = i_C(\mathbf{r}_2)$ with $C = \mathbb{R}^N_+$
- $f_k(\mathbf{r}_k) = i_{B_k}(\mathbf{r}_k)$ with $B_k = \{\mathbf{r}_k \in \mathbb{R}^{M_k} : \|\mathbf{y_k} - \mathbf{r}_k\|_2 \leq \epsilon_k\}$, $k = 3, \ldots, S$

SDMM Algorithm

1: Initialize $\gamma > 0$, $\hat{\mathbf{x}}^{(0)}$, $\mathbf{r}_i^{(0)}$ and $\mathbf{z}_i^{(0)}$, for $i = 1, \ldots, S$.
2: **while** No convergence criteria **do**
3: $\quad \hat{\mathbf{x}}^{(t)} = \left(\sum_{i=1}^{S} \mathsf{L}_i^H \mathsf{L}_i\right)^{-1} \sum_{i=1}^{S} \mathsf{L}_i^H (\mathbf{r}_i^{(t)} - \mathbf{z}_i^{(t)})$
4: $\quad$ **for all** $i = 1, \ldots, S$ **do**
5: $\quad\quad \mathbf{r}_i^{(t)} = \mathrm{prox}_{\gamma f_i}(\mathsf{L}_i \hat{\mathbf{x}}^{(t)} + \mathbf{z}_i^{(t-1)})$
6: $\quad\quad \mathbf{z}_i^{(t)} = \mathbf{z}_i^{(t-1)} + \mathsf{L}_i \hat{\mathbf{x}}^{(t)} - \mathbf{r}_i^{(t)}$
7: $\quad$ **end for**
8: **end while**
9: **return** $\hat{\mathbf{x}}^{(t)}$

- CORE MESSAGE: Steps 5 and 6 can be done in parallel $\forall i$

Implementation Details

Linear system

$$\mathbf{x}^{(t)} = (\sum_{i=1}^{S} \mathsf{L}_i^H \mathsf{L}_i)^{-1} \sum_{i=1}^{S} \mathsf{L}_i^H (\mathbf{r}_i^{(t-1)} - \mathbf{z}_i^{(t-1)})$$

- Solved iteratively using a conjugate gradient algorithm
- For the problem in hand $\sum_{i=1}^{S} \mathsf{L}_i^H \mathsf{L}_i = \Phi^H \Phi + 2\mathsf{I}$
- Bottleneck of the algorithm!
- Need simpler methods

Inexact ADMM-based approach

ADMM uses the following equivalent problem

$$\min_{\mathbf{x},\mathbf{z}} f(\mathbf{x}) + h(\mathbf{z}) \text{ subject to } \Phi\mathbf{x} + \mathbf{z} = \mathbf{y},$$

where

- $f(\mathbf{x}) = \|\mathsf{W}\Psi^H\mathbf{x}\|_1 + i_C(\mathbf{x})$, where $C = \mathbb{R}_+^N$
- $h(\mathbf{z}) = i_B(\mathbf{z})$, where $B = \{\mathbf{z} \in \mathbb{R}^M : \|\mathbf{z}\|_2 \leq \epsilon\}$
- It uses the augmented Lagrangian function

$$f(\mathbf{x}) + h(\mathbf{z}) + \frac{1}{\gamma}\boldsymbol{\lambda}^H(\Phi\mathbf{x} + \mathbf{z} - \mathbf{y}) + \frac{1}{2\gamma}\|\Phi\mathbf{x} + \mathbf{z} - \mathbf{y}\|_2^2$$

- Update for $\mathbf{x}$ based on a proximal linear approximation of the augmented Lagrangian

ADMM-based Algorithm

1: Initialize $\gamma, \mu, \beta > 0$, $\mathbf{x}^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$
2: **while** No convergence criteria **do**
3: $\quad \mathbf{z}^{(t+1)} = \mathrm{prox}_{\gamma h}(\mathbf{y} - \Phi\mathbf{x}^{(t)} - \boldsymbol{\lambda}^{(t)})$
4: $\quad \mathbf{x}^{(t+1)} = \mathrm{prox}_{\mu\gamma f}(\mathbf{x}^{(t)} - \mu\Phi^H(\boldsymbol{\lambda}^{(t)} + \Phi\mathbf{x}^{(t)} - \mathbf{y} + \mathbf{z}^{(t+1)}))$
5: $\quad \boldsymbol{\lambda}^{(t+1)} = \boldsymbol{\lambda}^{(t)} + \beta(\Phi\mathbf{x}^{(t+1)} - \mathbf{y} + \mathbf{z}^{(t+1)})$
6: **end while**
7: **return** $\mathbf{x}^{(t+1)}$

- Updates for $\mathbf{z}$ and $\boldsymbol{\lambda}$ are separable
- The gradient in 4 can be computed using a sum reduction approach since $\Phi^H\mathbf{y} = \sum_{i=1}^{R} \Phi_i^H \mathbf{y}_i$

Parallel Algorithm

Initialize $\gamma, \mu, beta > 0$, $\mathbf{x}^{(0)}$, $\mathbf{z}^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$

$\mathbf{g}_k^{(0)} = \Phi_k^H(\boldsymbol{\lambda}_k^{(0)} + \Phi_k \mathbf{x}^{(0)} - \mathbf{y} - \mathbf{z}_k^{(0)})$, for $k = 1, \ldots, R$

while No convergence criteria **do**

    $\mathbf{x}^{(t+1)} = \mathrm{prox}_{\mu\gamma f}(\mathbf{x}^{(t)} - \mu \sum_{k=1}^{R} \mathbf{g}_k^{(t)})$

    for all $k = 1, \ldots, R$ **do**

        $\mathbf{r}_k^{(t+1)} = \Phi_k \mathbf{x}^{(t+1)} - \mathbf{y}_k$

        $\mathbf{z}_k^{(t+1)} = \mathrm{prox}_{\gamma h_k}(-\mathbf{r}_k^{(t+1)} - \boldsymbol{\lambda}_k^{(t)})$

        $\boldsymbol{\lambda}_k^{(t+1)} = \boldsymbol{\lambda}_k^{(t)} + \beta(\mathbf{r}_k^{(t+1)} - \mathbf{z}_k^{(t+1)})$

        $\mathbf{g}_k^{(t+1)} = \Phi_k^H(\boldsymbol{\lambda}_k^{(t+1)} + \mathbf{r}_k^{(t+1)} - \mathbf{z}_k^{(t+1)})$

    end for

end while

return $\mathbf{x}^{(t+1)}$

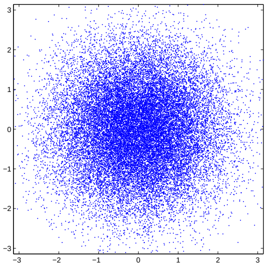
PURIFY

- PURIFY is an open-source code that provides functionality to perform radio interferometric imaging
- SDMM solvers implemented in C
- ADMM solvers implemented in MATLAB
- Implements the following sparsity priors:
  - Daubechies orthogonal wavelets
  - Total variation
  - Sparsity averaging
- Code available at github (http://basp-group.github.io/purify/)

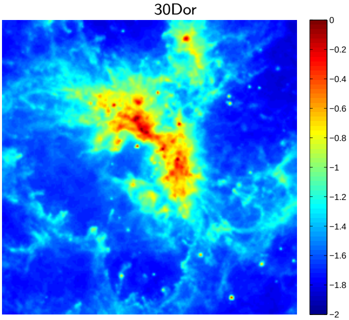
Outline

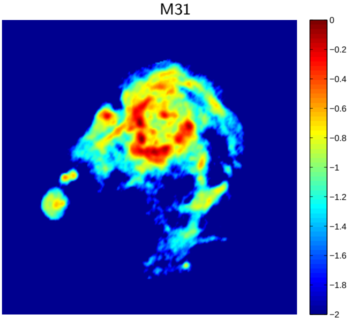
Simulation Setup

- M31 and 30Dor 256×256 test images
- Continuous visibilities with random Gaussian profile
- $\Phi = \text{GFZA}$
  - G : convolutional interpolation operator
  - F : fast Fourier transform
  - Z : upsampling operator
  - A = I : primary beam
- 30dB noise
- $0.2N \leq M \leq 2N$

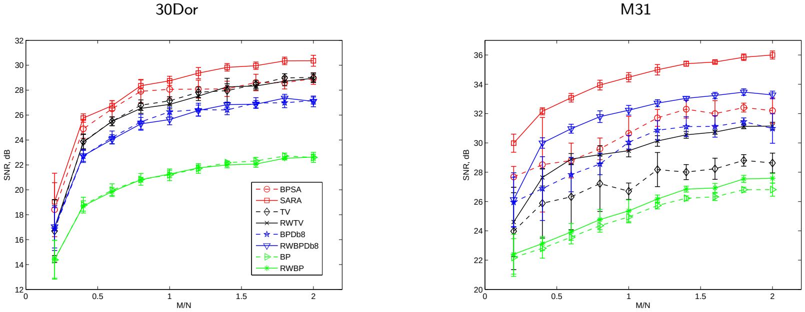


Test Images

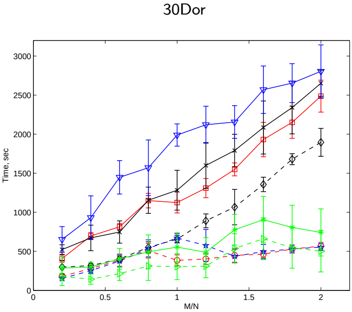


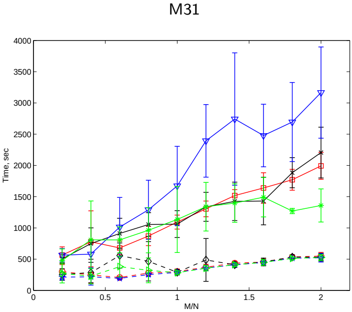


Reconstruction Quality Results (SDMM)

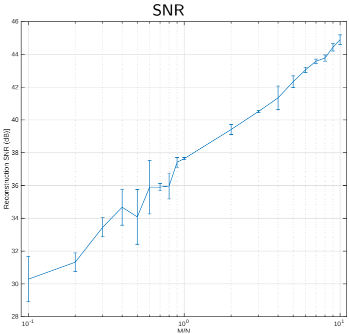


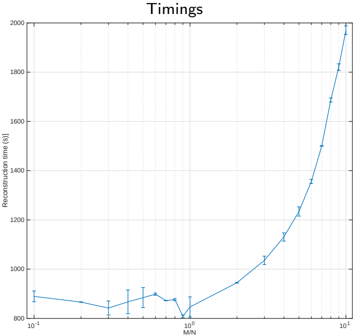
Timing Results (SDMM)





ADMM Results





- 40 dB noise
- Scalable to higher dimensions ($10N \approx 650K$ visibilities)

Outline

Summary

- We developed an open source code (PURIFY) that implements several convex imaging algorithms
- The proposed algorithms offer a parallel implementation structure

Future work:

- Direction dependent effects can be incorporated in the model as convolutional kernels in the operator G (Wolz et al. 2013)
- New ways to improve the computational efficiency of the algorithm have to be explored:
  - Stochastic ADMM approaches (Azadi et al 2014)
  - Faster implementations for the sparsity operators
  - Dimensionality reduction techniques

Thank You!