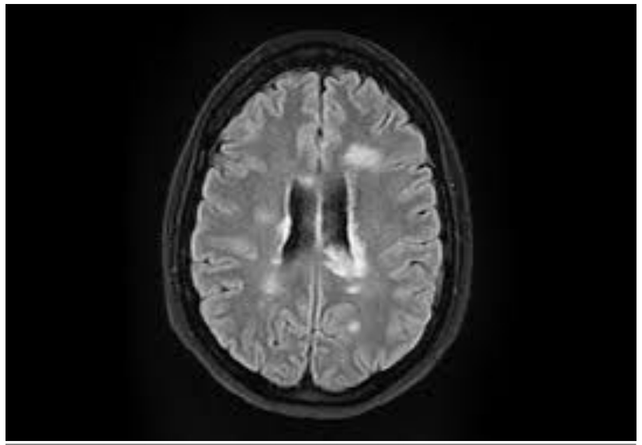
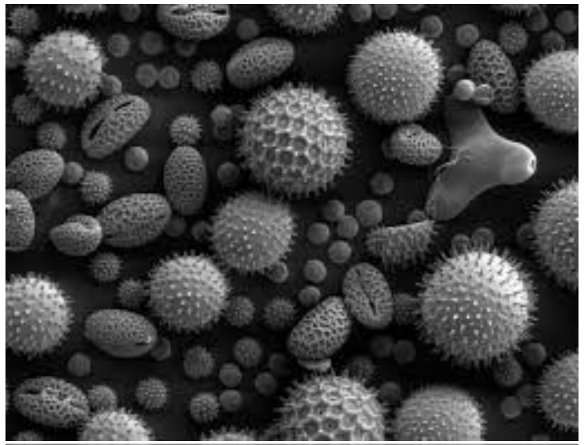
The COSMIC Project

(Tackling Problems in Biomedical and Astrophysical Imaging)





Samuel Farrens CEA

Bât 709 Pe 279

Outline

- The COSMIC Project
- Inverse Problems
- Compressed Sensing
- PySAP
  - Sparse2D
  - ModOpt
  - Plug-Ins
- The Future

Outline

❖ The COSMIC Project

The COSMIC Project

Compressed Sensing for Magnetic Resonance Imaging & Cosmology

DRF funded collaboration between CosmoStat and NeuroSpin to share knowledge and develop software tools for compressed sensing and image reconstruction.

Philippe Ciuciu (PI)

+

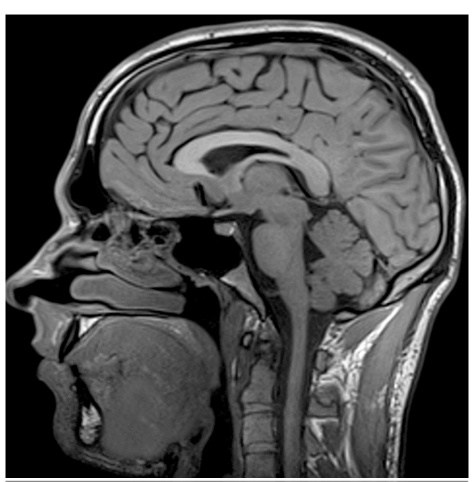
Jean-Luc Starck

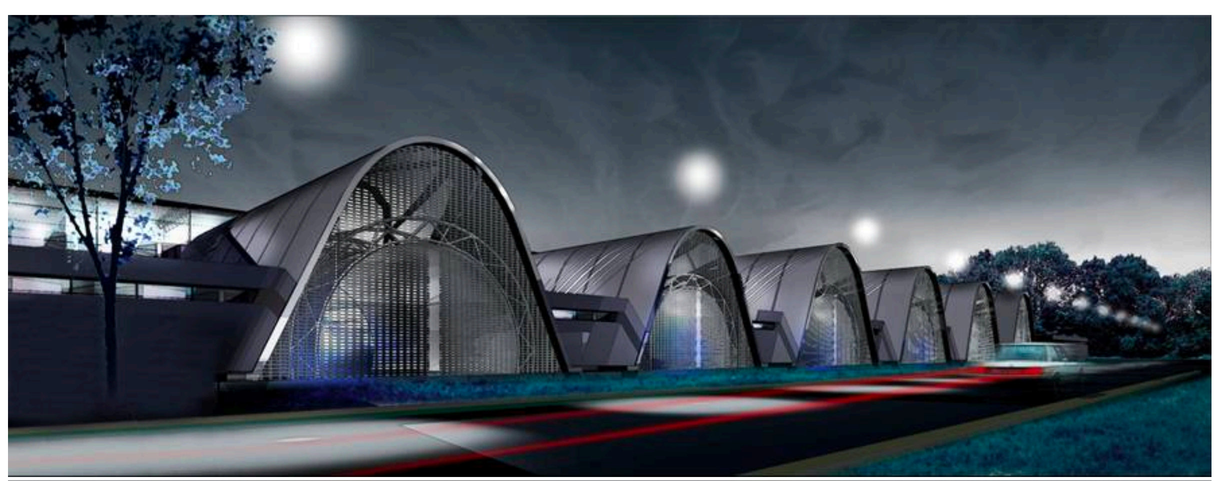
http://cosmic.cosmostat.org/

NeuroSpin

Cutting-edge neuroimaging research centre based at CEA

- Magnetic Resonance Imaging (MRI)
- Cognitive Neuroscience
- Preclinical and Clinical Neuroscience
- Image Processing



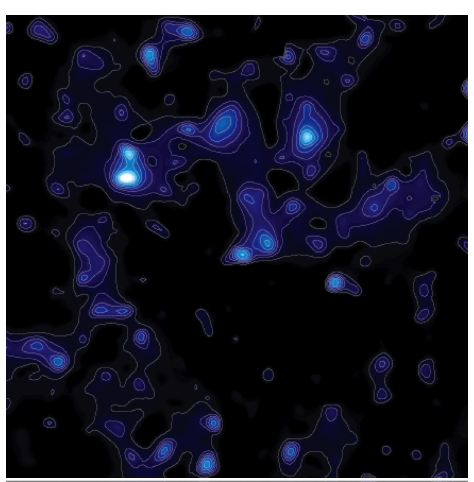


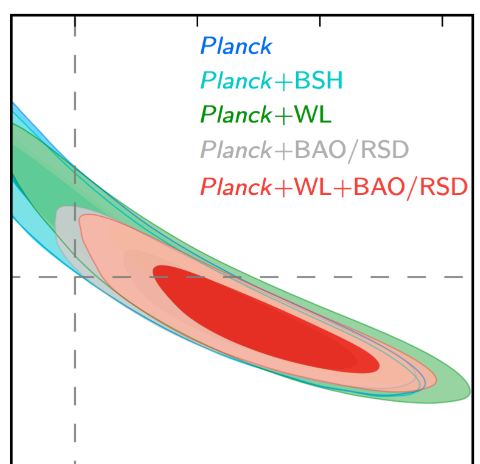
The COSMIC Project

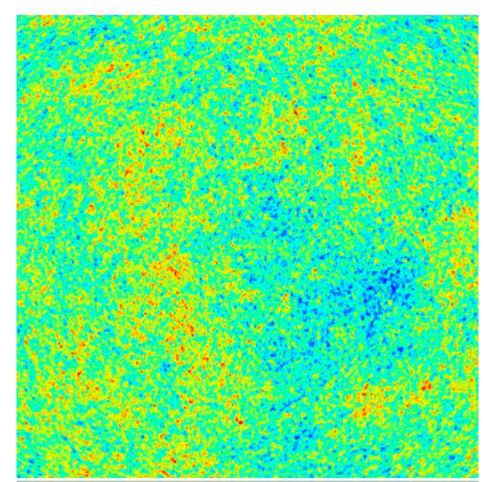
CosmoStat

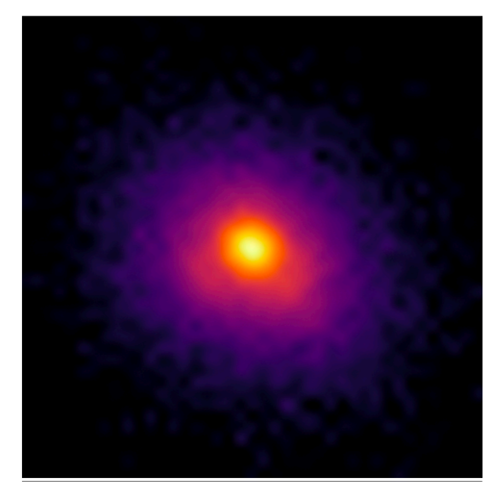
Interdisciplinary lab at DAP combining cosmology and statistical methods.

- Cosmology
- Signal Processing
- Machine Learning









http://www.cosmostat.org/

Biomedical Context

Just under a third of Europeans are affected by neurocognitive disorders and around 5% of children suffer from learning difficulties.

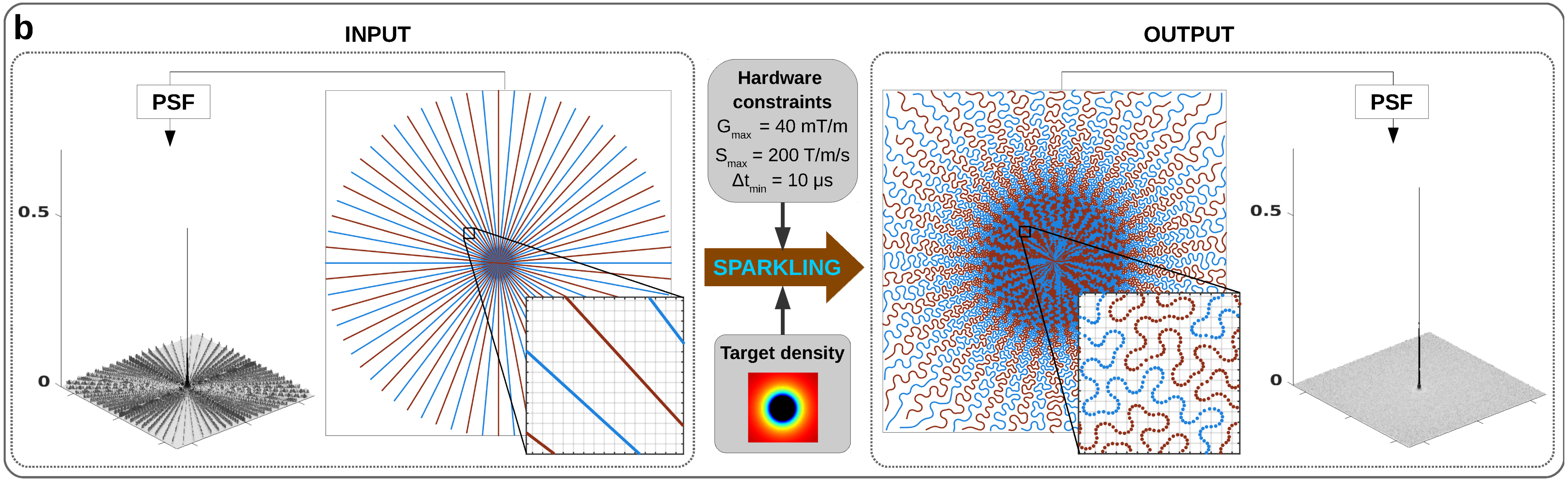
Characterising injuries in the brains of premature infants could potentially avoid long-term consequences for brain development. This, however, is extremely challenging given the long scan times required to acquire high resolution images.

COSMIC aims to significantly reduce MRI scan times while retaining full HR resolution in the reconstructed images. This will be achieved with combination of novel scanning schemes and cutting edge image reconstruction software.

The COSMIC Project

Biomedical Context

Spreading Projection Algorithm for Rapid K-space sampLING (**SPARKLING**)



[Lazarus, et al MRM 2019]

Astrophysical Context

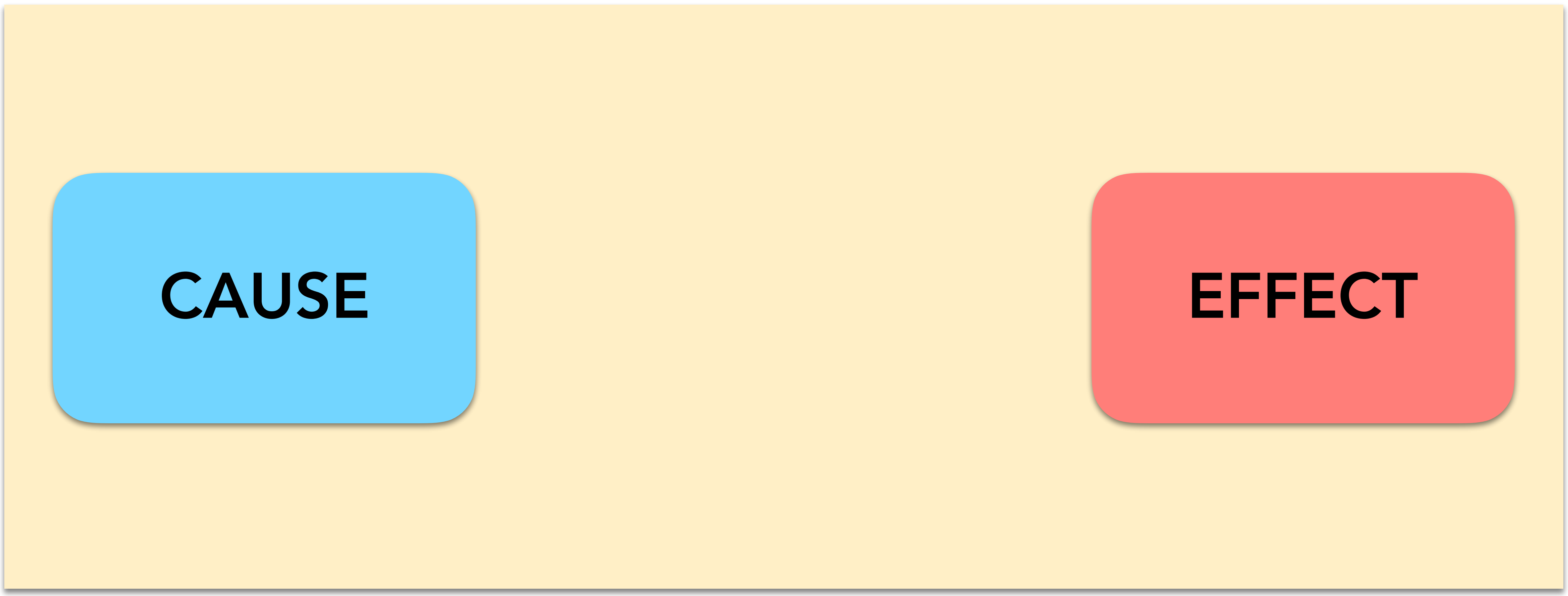
Upcoming cosmological surveys (*e.g.* Euclid, LSST, SKA, *etc.*) will have to process petabyte scales of data in an efficient and automated way. Extracting the most from the data produced will require the best possible image analysis tools.

For radio data, the problem of reconstructing masked images is mathematically identical that of reconstructing subsampled MR images. This means that tools jointly developed by CosmoStat and NeuroSpin can also be applied to astrophysical data.

Outline

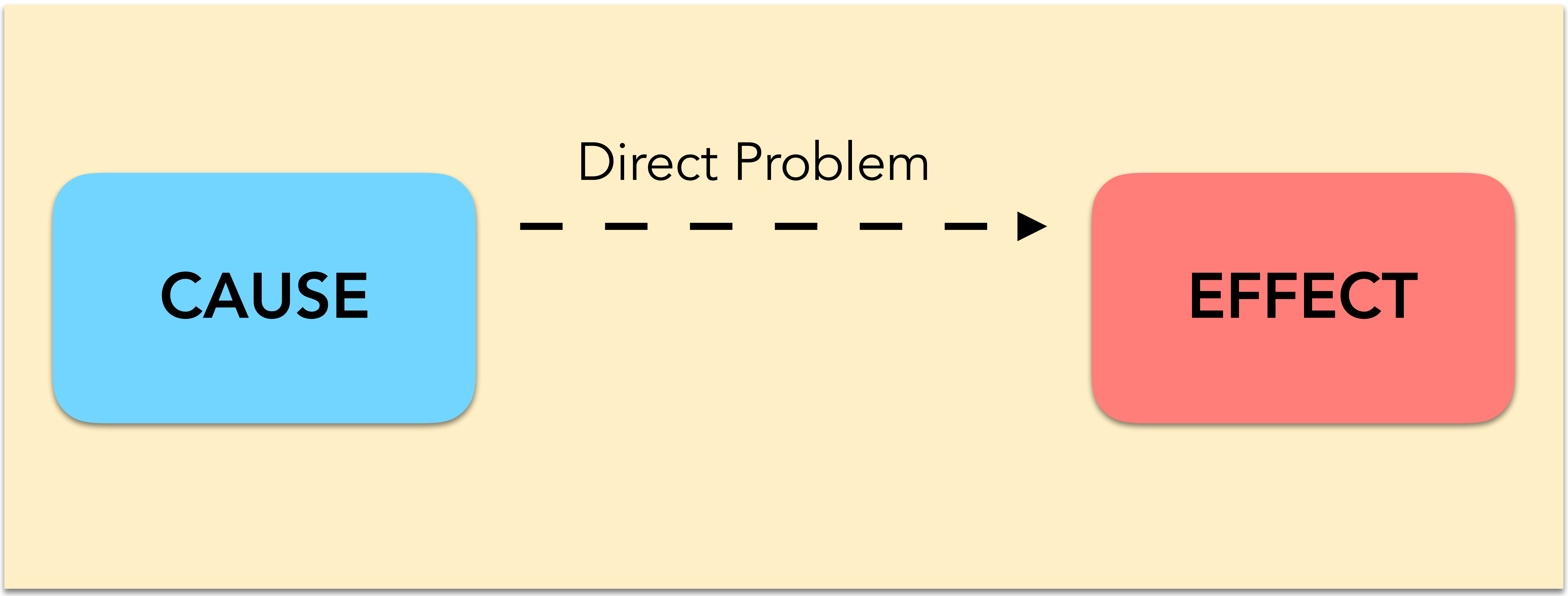
❖ Inverse Problems

Inverse Problems



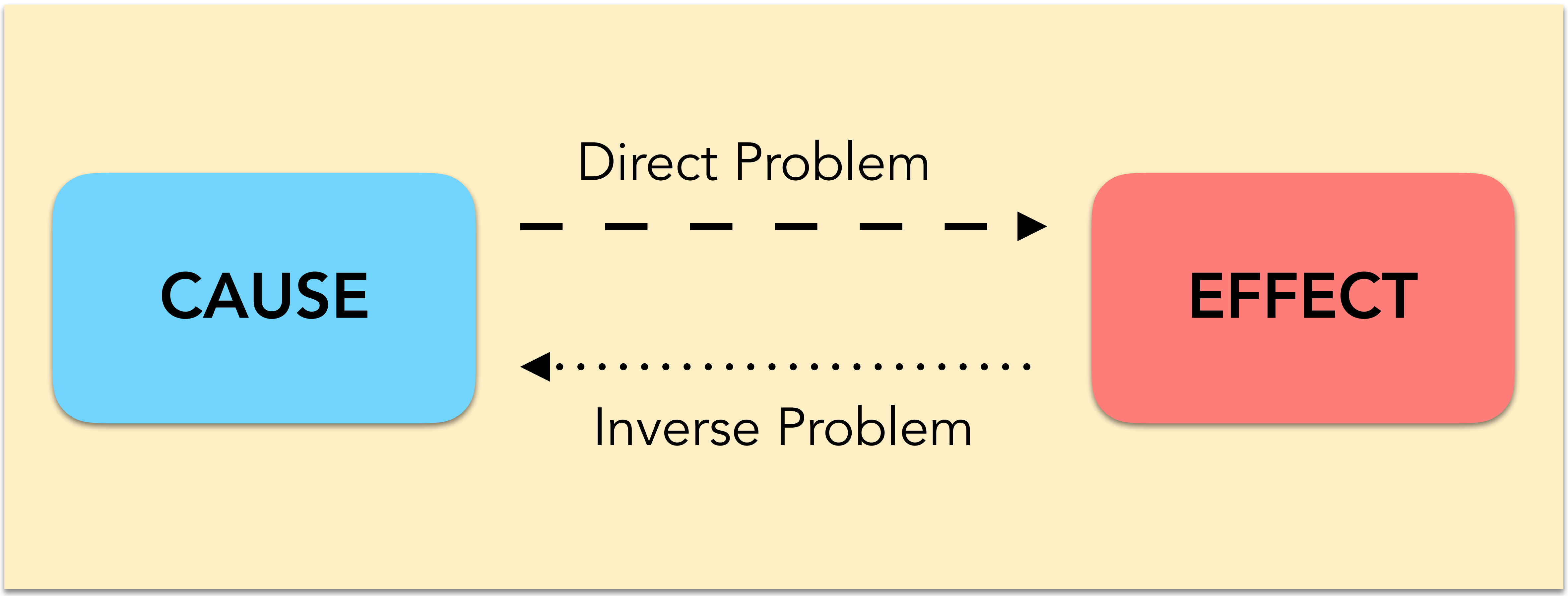
With an inverse problem one attempts to obtain information about a physical system from observed measurements.

Inverse Problems



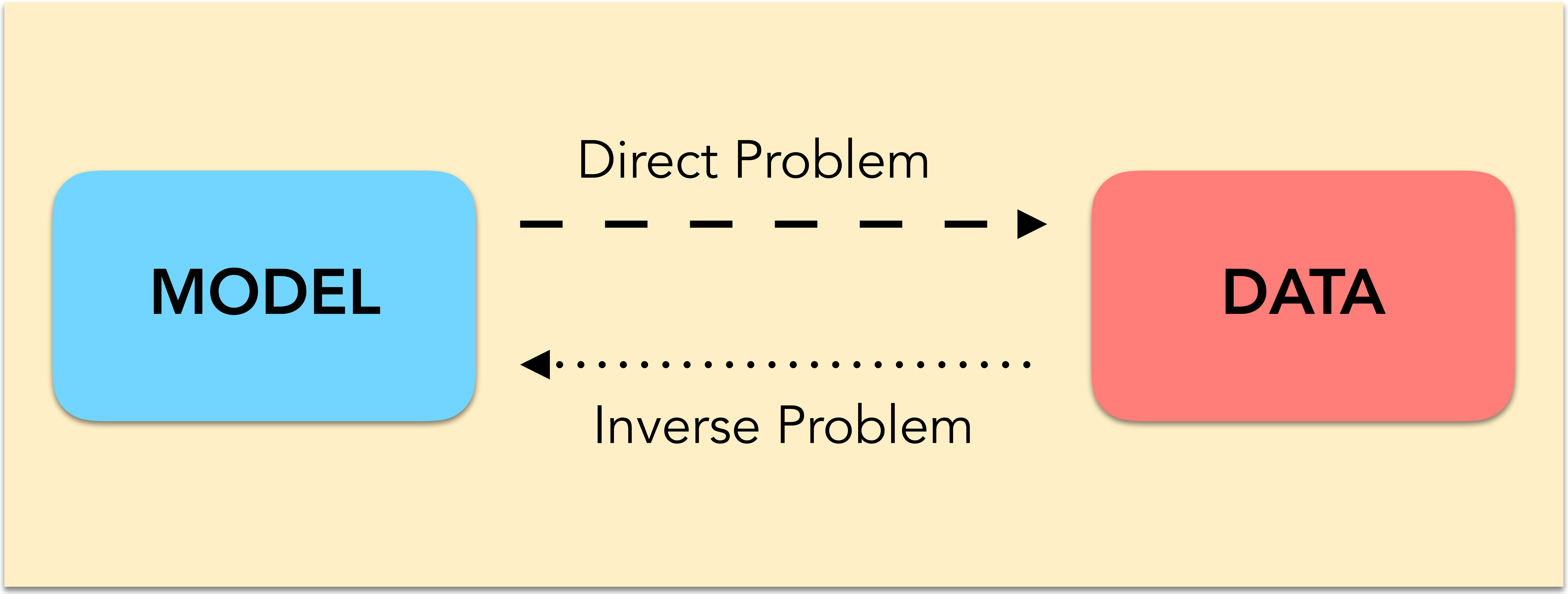
With an inverse problem one attempts to obtain information about a physical system from observed measurements.

Inverse Problems



With an inverse problem one attempts to obtain information about a physical system from observed measurements.

Inverse Problems



With an inverse problem one attempts to obtain information about a physical system from observed measurements.

Linear Inverse Problem

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

$\mathbf{y} \in \mathbb{R}^m$	$\mathbf{A} \in \mathbb{R}^{m \times n}$	$\mathbf{x} \in \mathbb{R}^n$
Data Vector	Operation Matrix	Model Vector

Inverse Problems

Straight Line : Direct Problem

$$y = mx + b$$

$$x = \begin{bmatrix} 8 & 2 & 11 & 6 & 5 & 4 & 12 & 9 & 6 & 11 \end{bmatrix}$$

Model

$$m = -1.1$$

$$b = 14$$

Model Vector

$$\mathbf{x} = \begin{bmatrix} 14.0 & -1.1 \end{bmatrix}$$

Operation Matrix

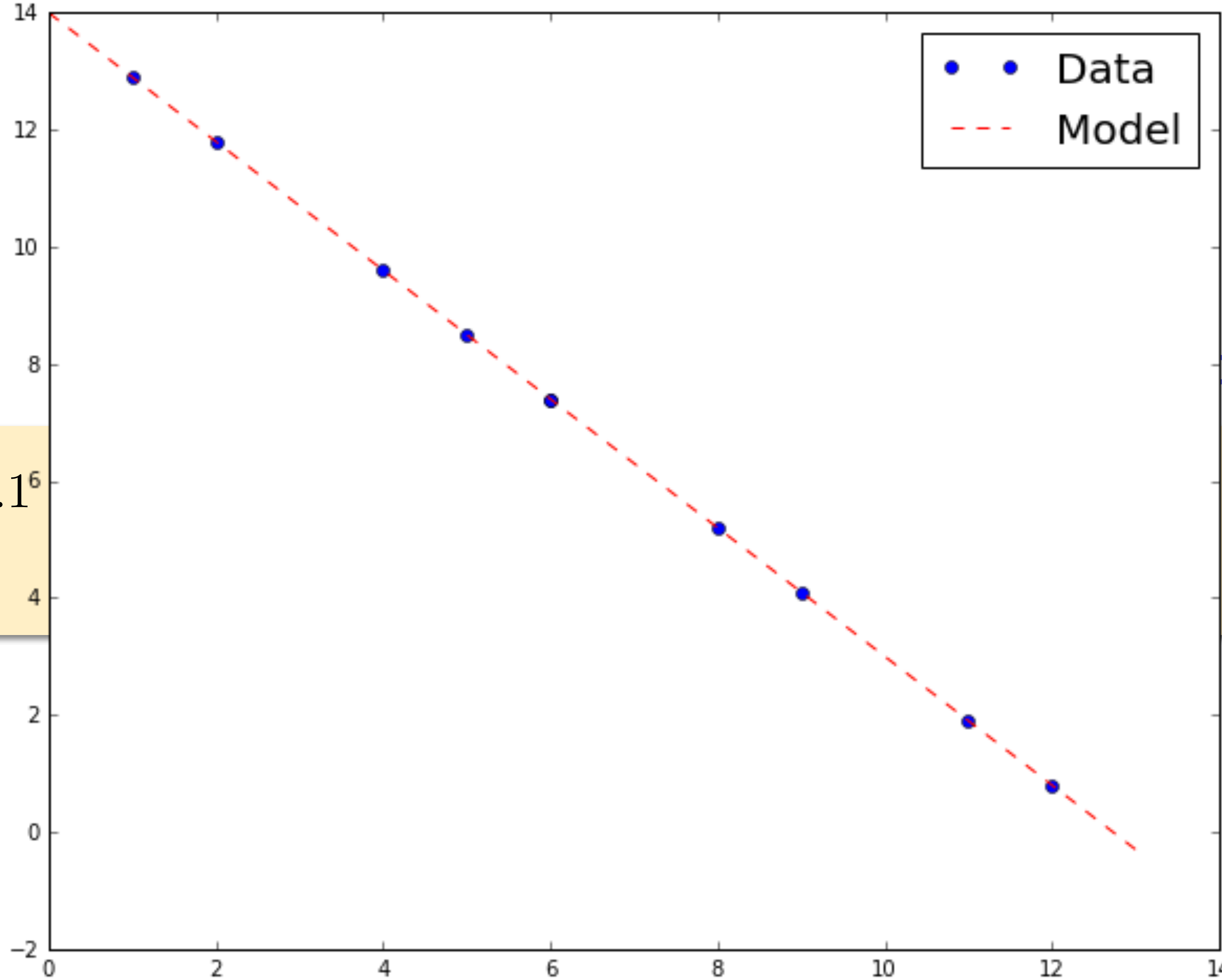
$$\mathbf{A} = \begin{bmatrix} 1 & 8 \\ 1 & 2 \\ 1 & 11 \\ 1 & 6 \\ 1 & 5 \\ 1 & 4 \\ 1 & 12 \\ 1 & 9 \\ 1 & 6 \\ 1 & 11 \end{bmatrix}$$

Data Vector

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

Inverse Problems

Straight Line : Direct Problem



Model

$$m = -1.1$$
$$b = 14$$

Data Vector

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

Straight Line : Inverse Problem

$$y = mx + b$$

$$x = \begin{bmatrix} 8 & 2 & 11 & 6 & 5 & 4 & 12 & 9 & 6 & 11 \end{bmatrix}$$
$$y = \begin{bmatrix} 3 & 10 & 3 & 6 & 8 & 12 & 1 & 4 & 9 & 14 \end{bmatrix}$$

Data Vector

$$\mathbf{y} = \begin{bmatrix} 3 \\ 10 \\ 3 \\ 6 \\ 8 \\ 12 \\ 1 \\ 4 \\ 9 \\ 14 \end{bmatrix}$$

Operation Matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 8 \\ 1 & 2 \\ 1 & 11 \\ 1 & 6 \\ 1 & 5 \\ 1 & 4 \\ 1 & 12 \\ 1 & 9 \\ 1 & 6 \\ 1 & 11 \end{bmatrix}$$

Model Vector

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

Inverse Problems

Straight Line : Inverse Problem

$$y = mx + b$$

$$x = \begin{bmatrix} 8 & 2 & 11 & 6 & 5 & 4 & 12 & 9 & 6 & 11 \end{bmatrix}$$
$$y = \begin{bmatrix} 3 & 10 & 3 & 6 & 8 & 12 & 1 & 4 & 9 & 14 \end{bmatrix}$$

Data Vector

$$\mathbf{y} = \begin{bmatrix} 3 \\ 10 \\ 3 \\ 6 \\ 8 \\ 12 \\ 1 \\ 4 \\ 9 \\ 14 \end{bmatrix}$$

Operation Matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 8 \\ 1 & 2 \\ 1 & 11 \\ 1 & 6 \\ 1 & 5 \\ 1 & 4 \\ 1 & 12 \\ 1 & 9 \\ 1 & 6 \\ 1 & 11 \end{bmatrix}$$

Model Vector

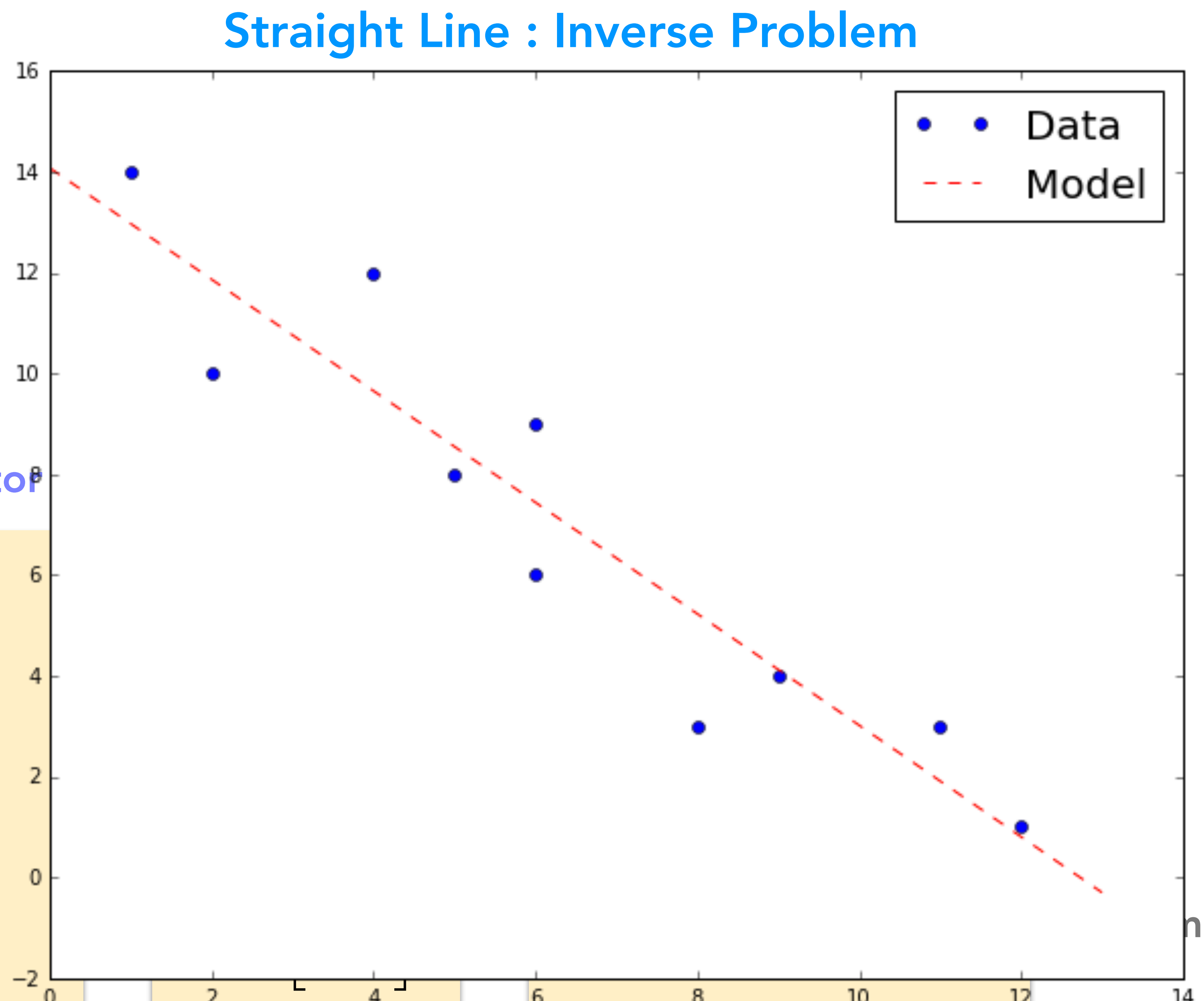
$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$$

$$\downarrow$$

$$\mathbf{x} = \left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{y}$$

(Normal Equation)

Inverse Problems



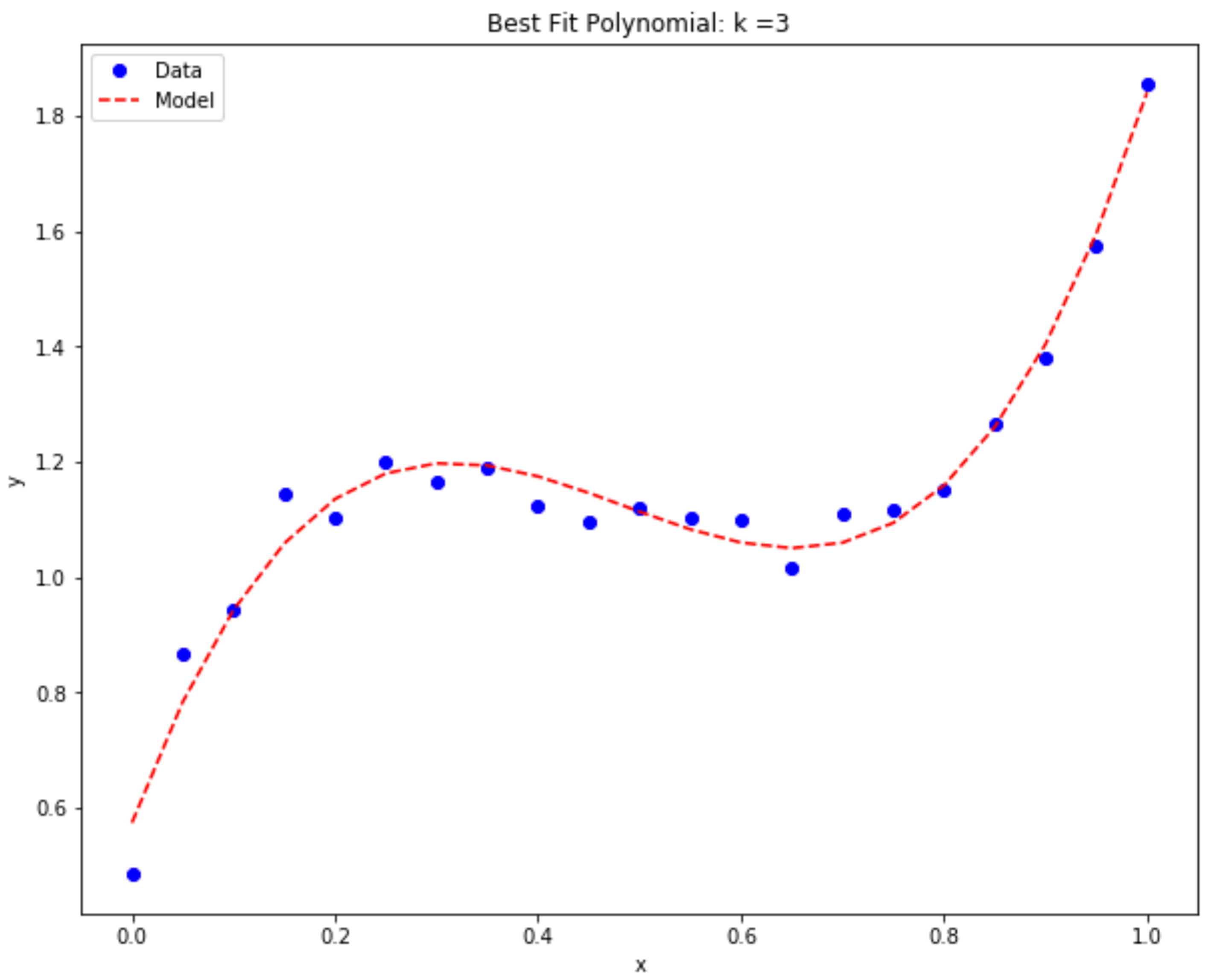
Data Vector

$$\mathbf{y} = \begin{bmatrix} 3 \\ 10 \\ 3 \\ 6 \\ 8 \\ 12 \\ 1 \\ 4 \\ 9 \\ 14 \end{bmatrix}$$

nal Equation)

Polynomial Line : Inverse Problem

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_k x^k$$



Inverse Problems

Well-Posed Problem

1. A solution exists
2. The solution is unique
3. The solution's behaviour changes continuously with the initial conditions

Ill-Posed Problem

1. No solution exists
2. The solution is not unique
3. The problem is ill-conditioned

Well-Conditioned Problem

$$\overset{\mathbf{y}}{\begin{bmatrix} 4 \\ 7 \end{bmatrix}} = \overset{\mathbf{A}}{\begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}} \overset{\mathbf{x}}{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}$$

$$\begin{bmatrix} 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2.01 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Ill-Conditioned Problem

$$\begin{bmatrix} 3 \\ 1.47 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0.48 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 1.47 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0.49 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Inverse Problems

Well-Conditioned Problem

$$\begin{array}{cccc} \mathbf{y} & & \mathbf{A} & \mathbf{x} \\ \begin{bmatrix}4\\7\end{bmatrix} & = & \begin{bmatrix}1 & 2\\2 & 3\end{bmatrix} & \begin{bmatrix}x_1\\x_2\end{bmatrix} \end{array} \quad \rightarrow \quad \begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}2\\1\end{bmatrix}$$

$$\begin{bmatrix}4\\7\end{bmatrix} = \begin{bmatrix}1 & 2\\2.01 & 3\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

Ill-Conditioned Problem

$$\begin{bmatrix}3\\1.47\end{bmatrix} = \begin{bmatrix}1 & 2\\0.48 & 0.99\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \quad \rightarrow \quad \begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix}$$

$$\begin{bmatrix}3\\1.47\end{bmatrix} = \begin{bmatrix}1 & 2\\0.49 & 0.99\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

Well-Conditioned Problem

$$\begin{array}{ccc} \mathbf{y} & \mathbf{A} & \mathbf{x} \end{array}$$

$$\begin{bmatrix} 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2.01 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.96 \\ 1.02 \end{bmatrix}$$

Ill-Conditioned Problem

$$\begin{bmatrix} 3 \\ 1.47 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0.48 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 1.47 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0.49 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Inverse Problems

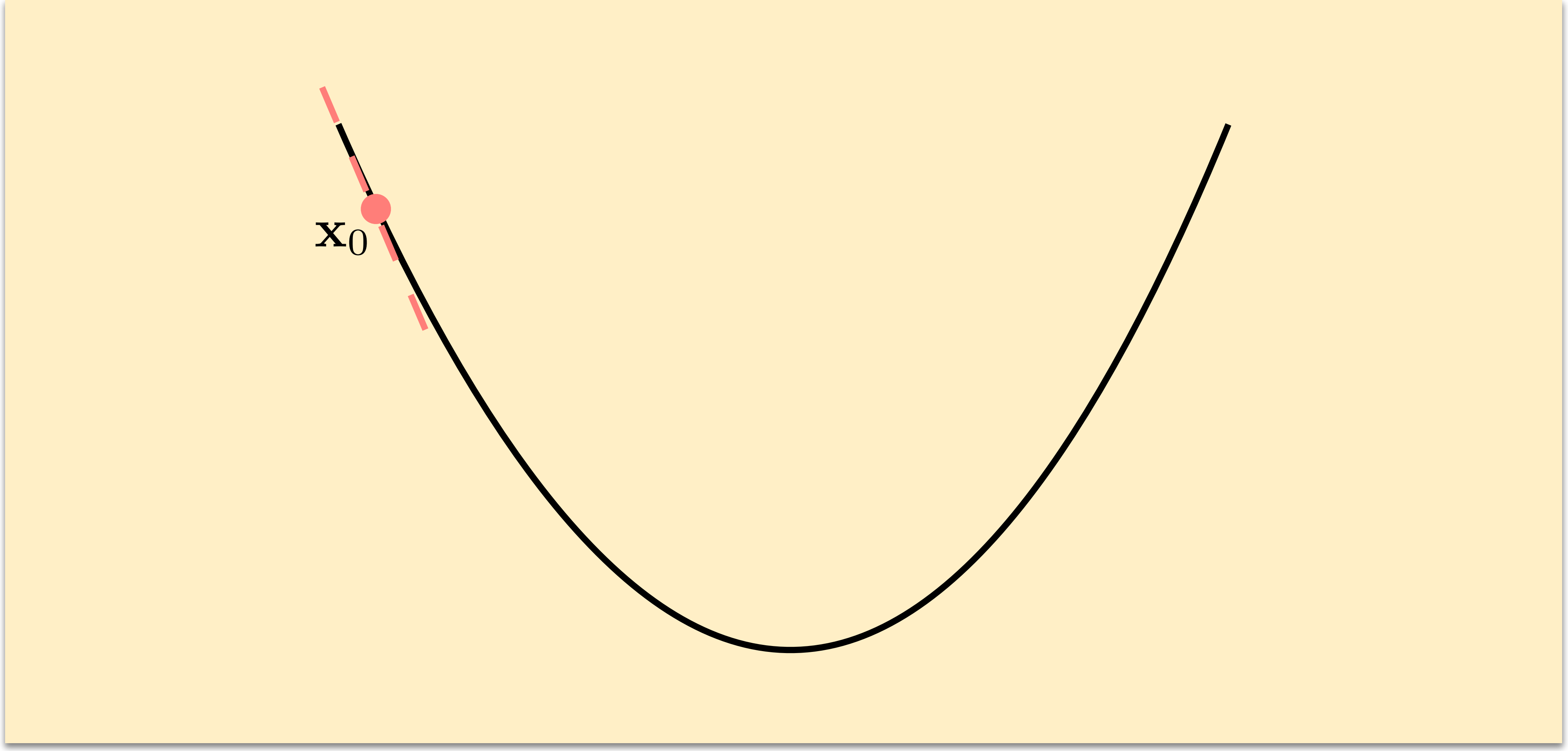
$$\underset{\mathbf{x}}{\mathrm{argmin}} \quad \frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda R(\mathbf{x})$$

1. Find **x** such that **y-Ax** is small
2. We have some prior knowledge about the properties of **x** given by R(**x**)

Inverse Problems

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$$

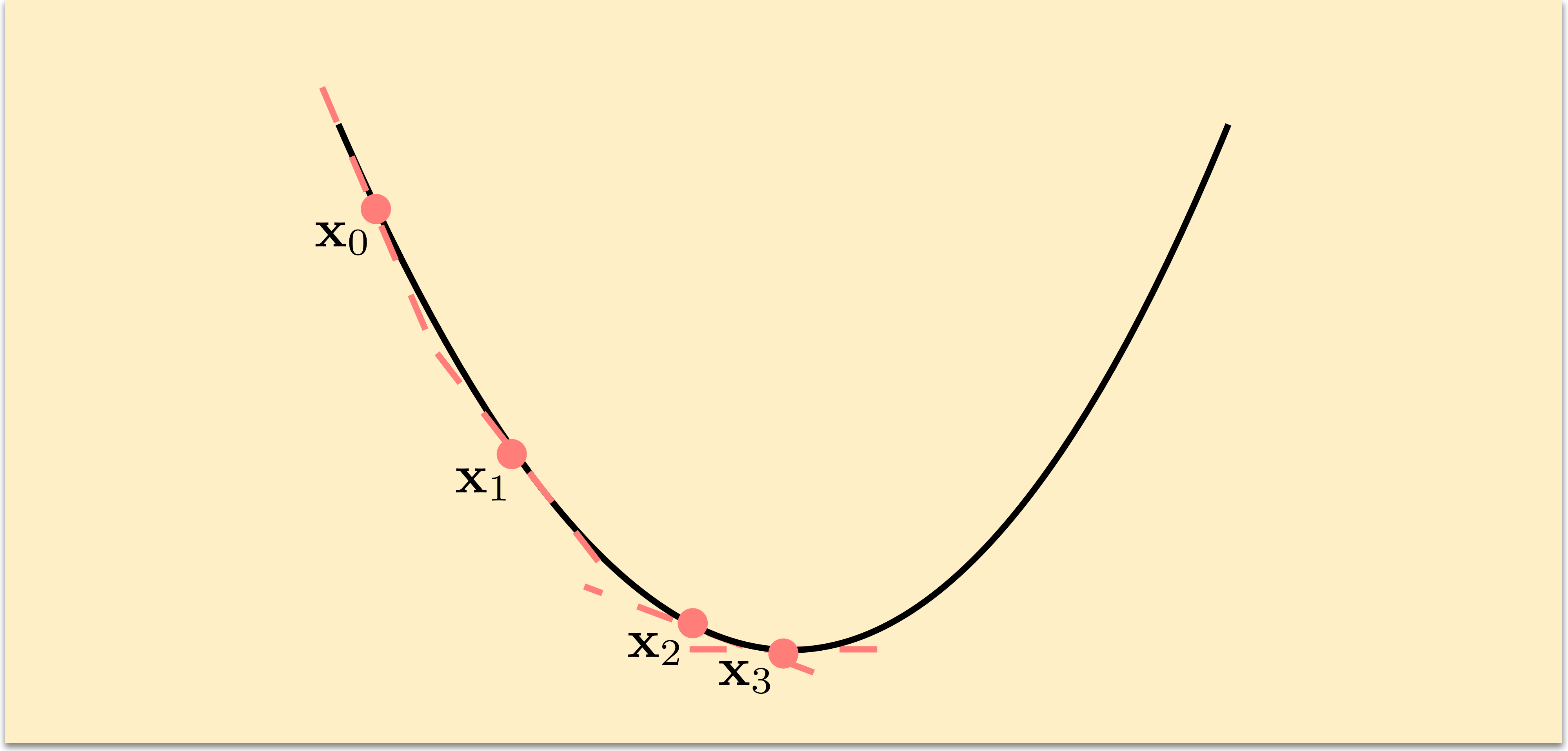
$$\nabla F(\mathbf{x}) = \mathbf{A}^T\left(\mathbf{y} - \mathbf{A}\mathbf{x}\right)$$



Inverse Problems

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$$

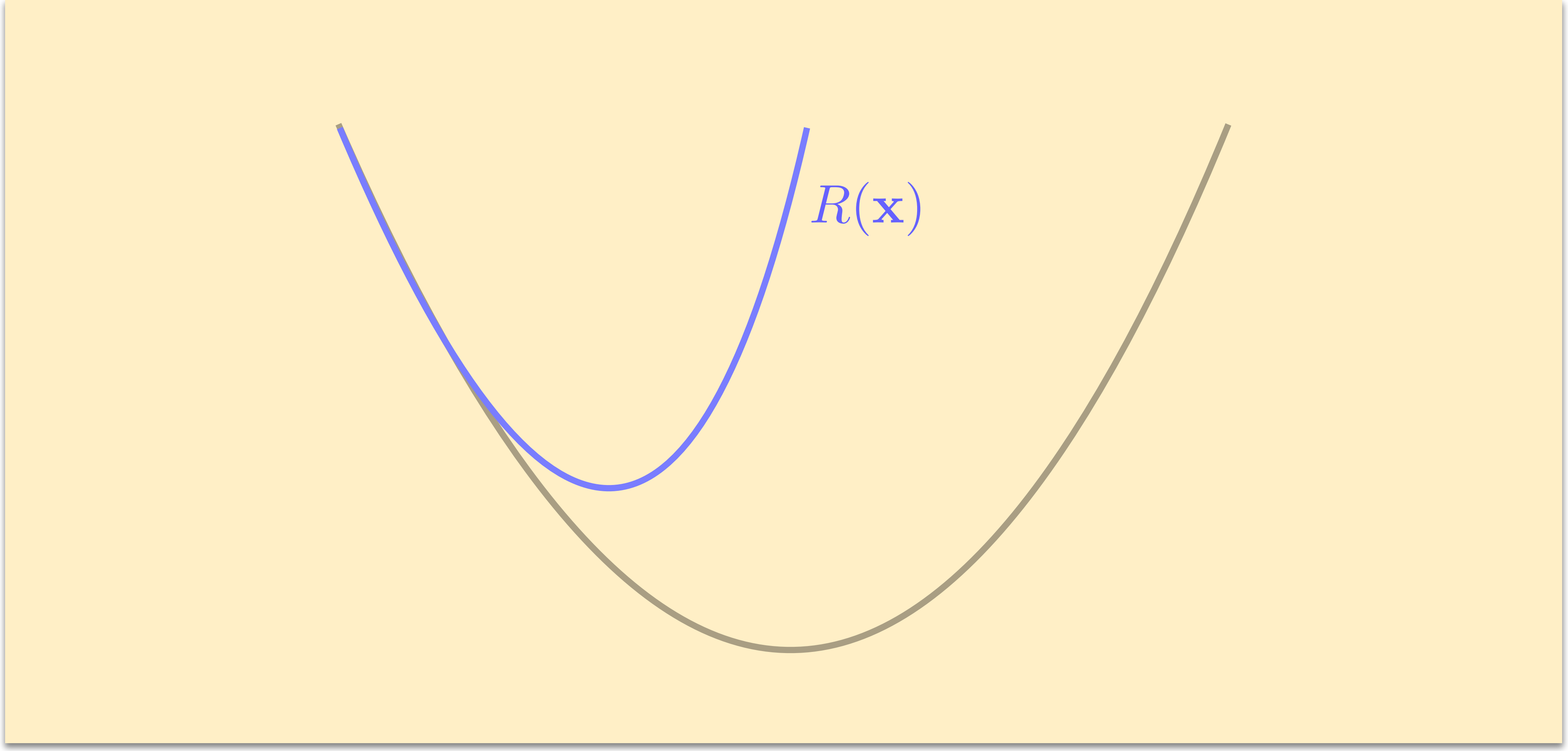
$$\nabla F(\mathbf{x}) = \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x})$$



Inverse Problems

$$F(\mathbf{x}) = \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$$

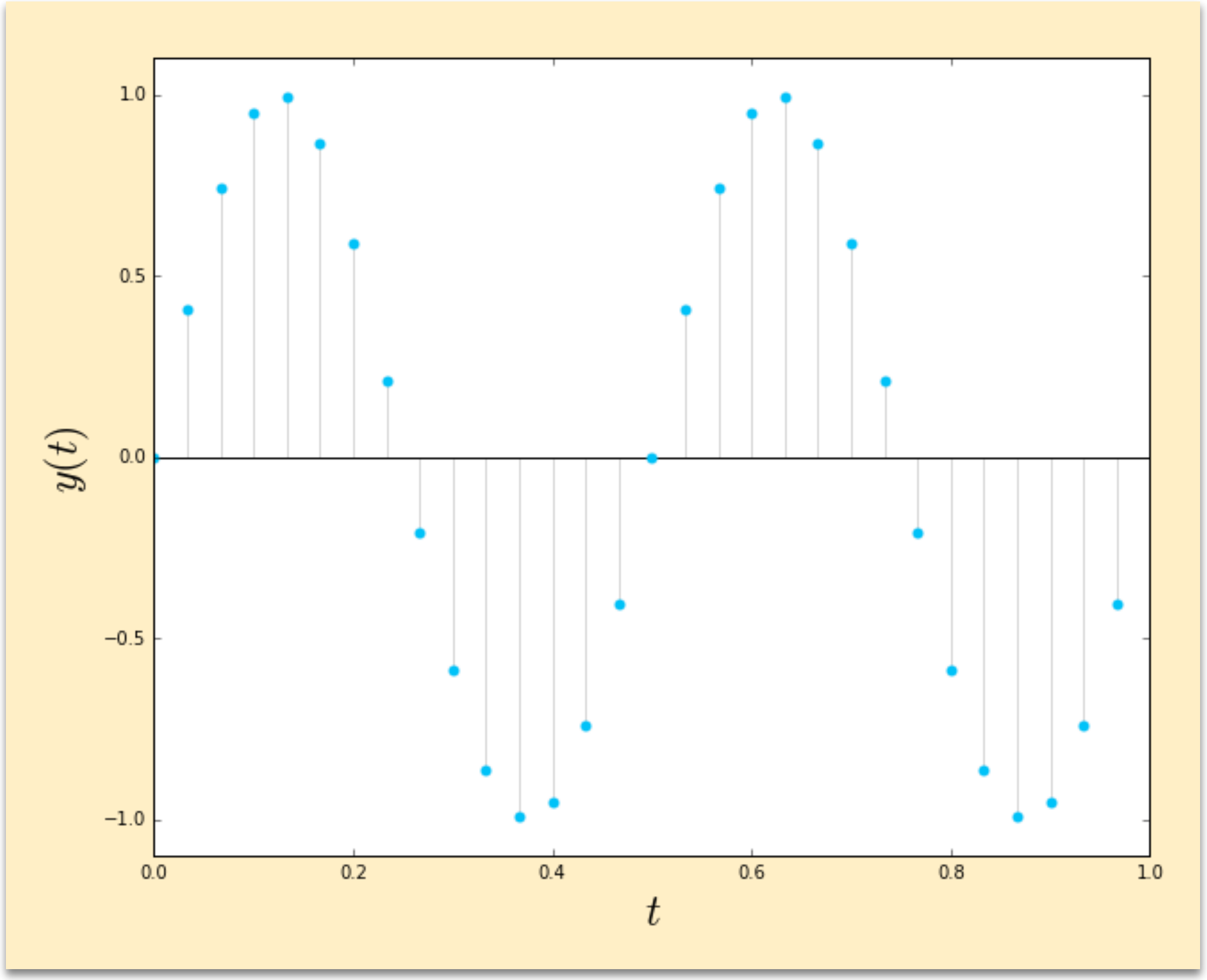
$$\nabla F(\mathbf{x}) = \mathbf{A}^T\left(\mathbf{y} - \mathbf{A}\mathbf{x}\right)$$



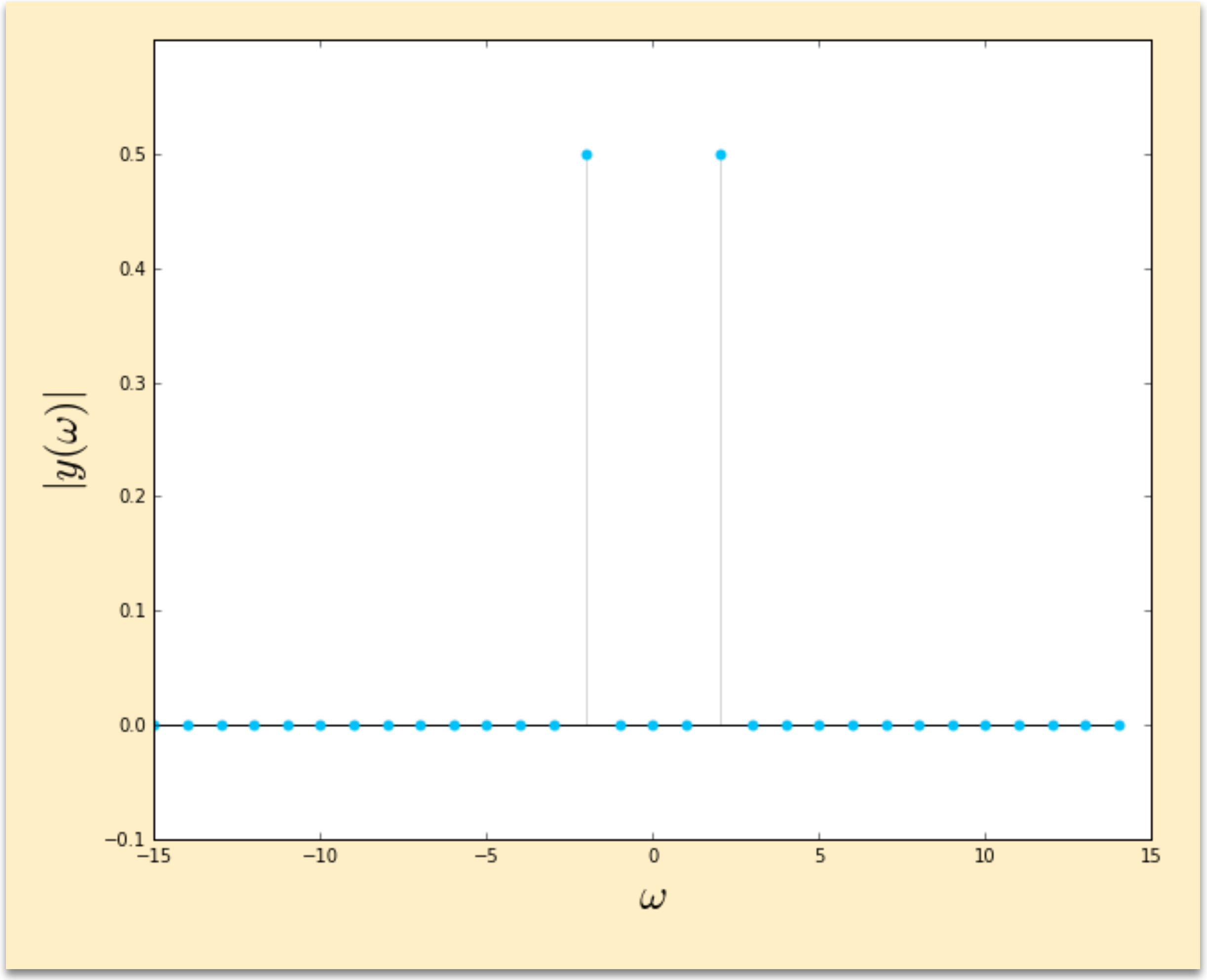
Inverse Problems

A sparse signal is one that is comprised mostly of zeros when expressed in the appropriate basis.

Direct Space



Sparse Space



Inverse Problems

$$\mathbf{x} = \phi\alpha = \sum_{i=1}^{n} \phi_i \alpha_i$$

ϕ is the dictionary that converts the signal to a sparse representation. (*e.g.* Fourier transform, wavelet transform, *etc.*)

Measuring Sparsity

$$\|\alpha\|_0 \rightarrow \|\alpha\|_1 = \sum_{i=1}^{n} |\alpha_i|$$

Not convex

Sparse Minimisation

$$\underset{\alpha}{\operatorname{argmin}} \quad \frac{1}{2}\|\mathbf{y} - A\phi\alpha\|_2^2 + \lambda\|\alpha\|_1$$

Applications

- Denoising
- Deconvolution
- Component Separation
- Inpainting
- Blind Source Separation
- Minimisation algorithms
- **Compressed Sensing**

Outline

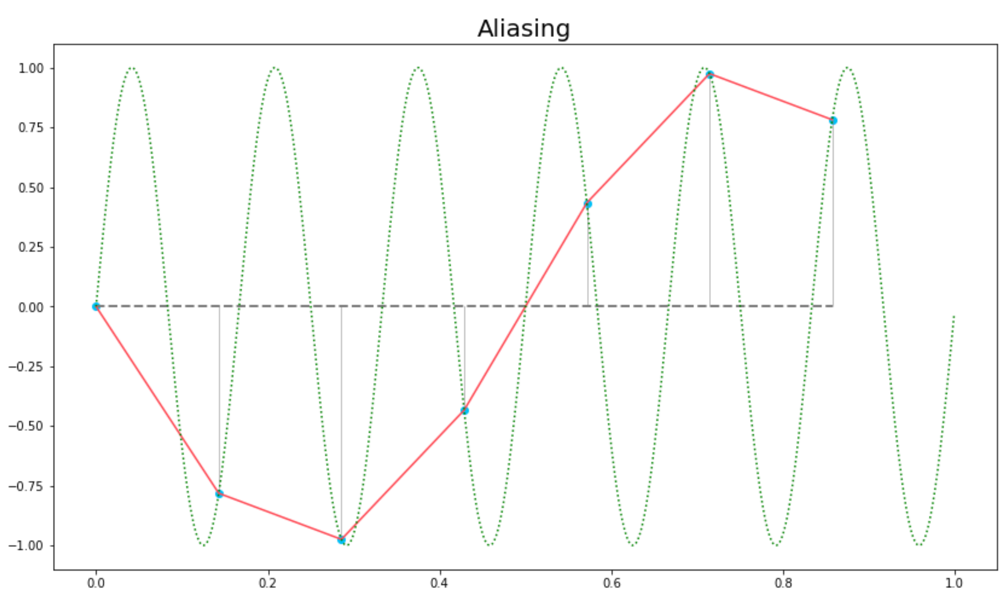
❖ Compressed Sensing

Compressed Sensing

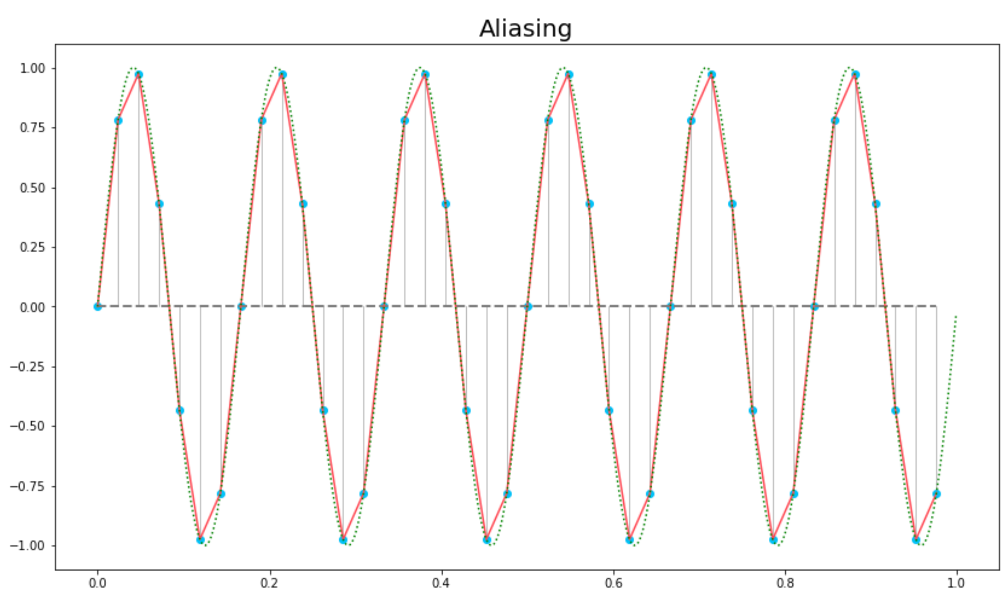
Nyquist-Shannon Sampling Theorem

A bandlimited signal can perfectly be recovered if the sample frequency (*i.e.* number of sample taken per unit time/space) is at least twice the highest frequency contained in the signal.

$$f_s \geq 2f_c$$



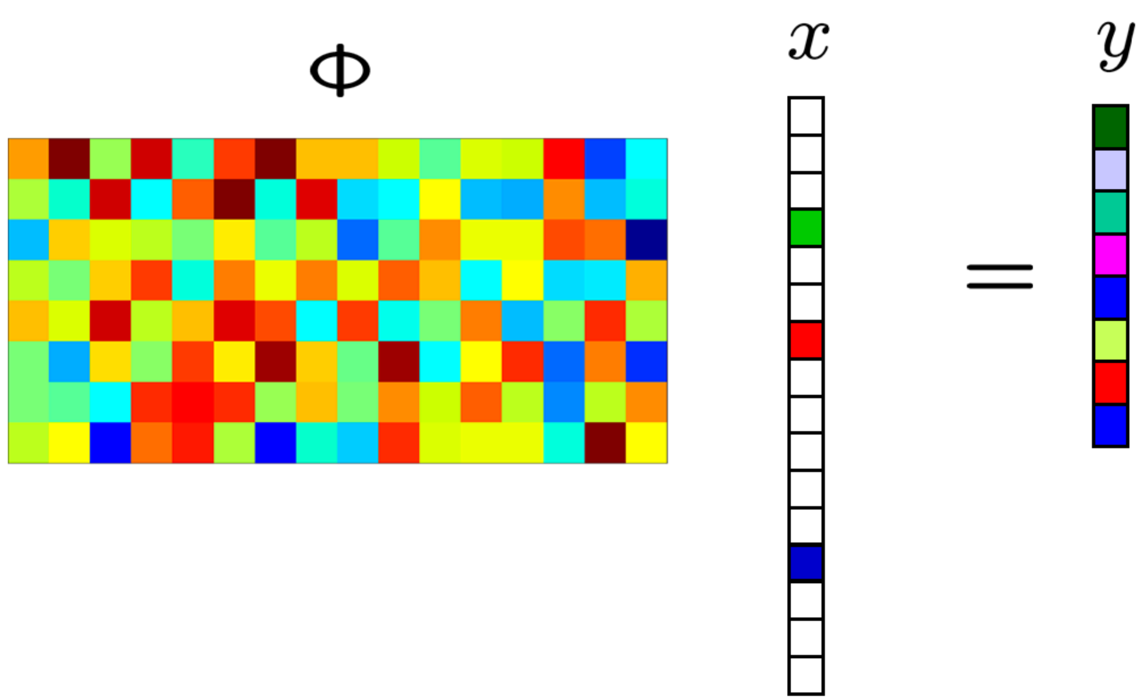
freq = 6, samples = 7



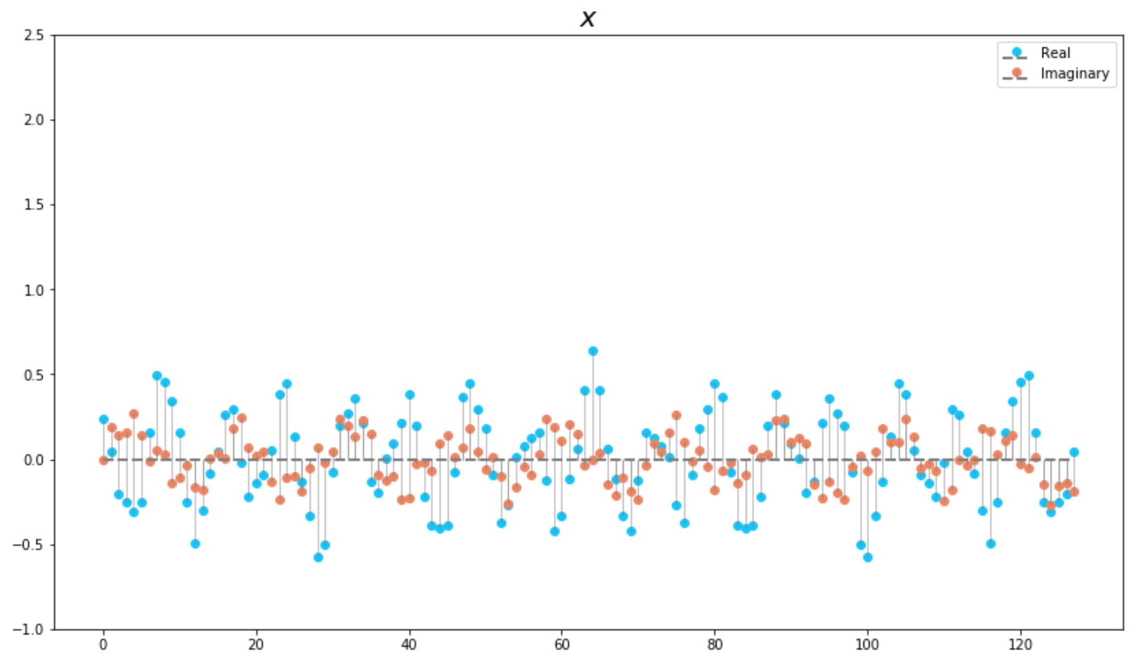
freq = 6, samples = 42

What is Compressed Sensing?

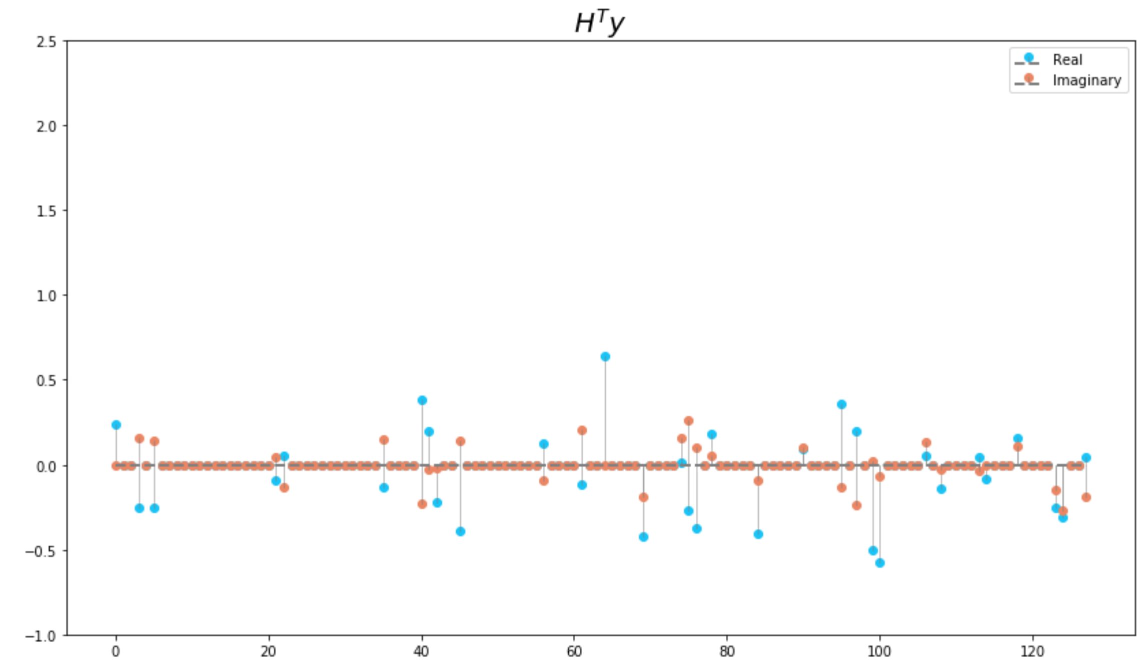
Compressed (or compressive) sensing is a paradigm that allows one to sample certain singals at a rate lower than the Nyquist rate by exploiting the sparsity of the signal in a given domain.



Compressed Sensing

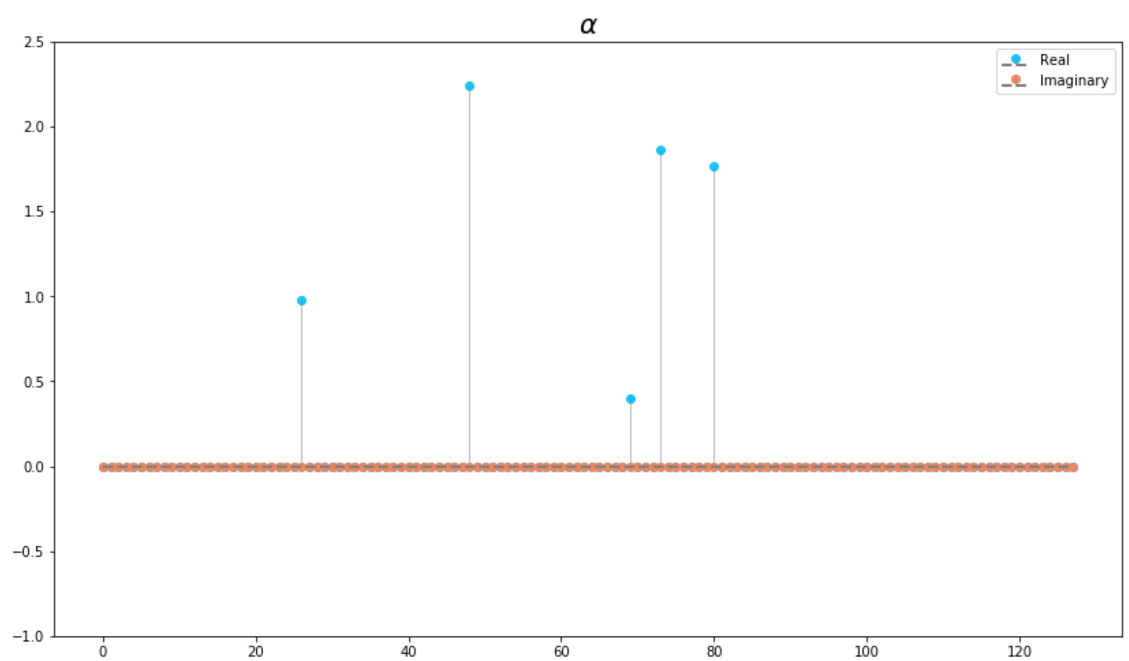


True Signal



Masked Observation

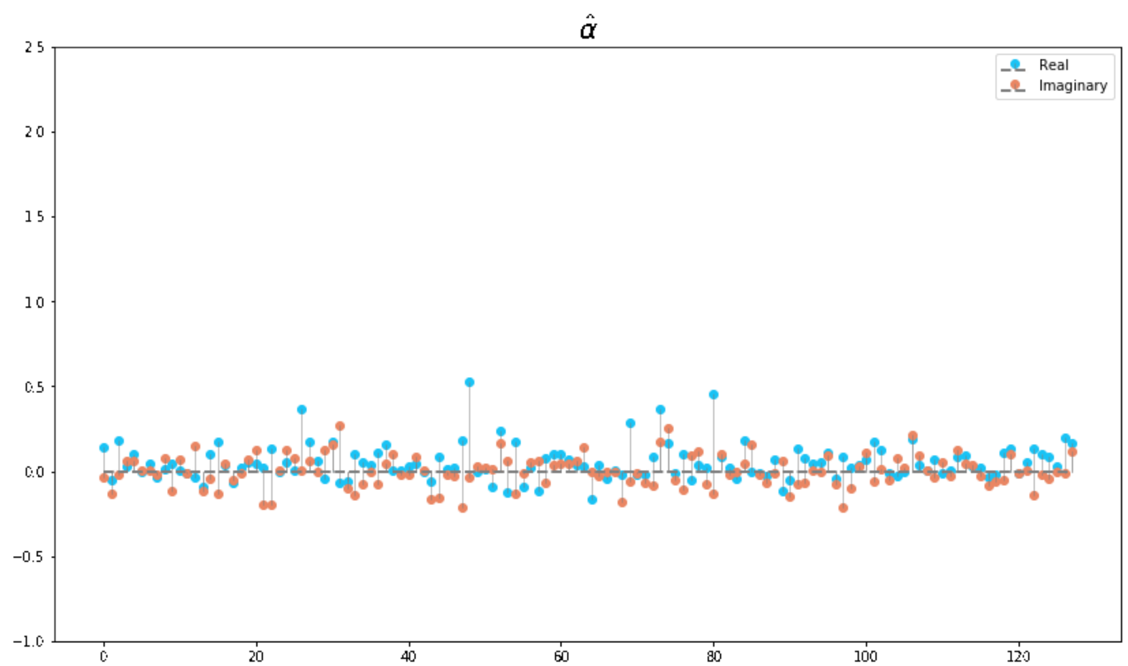
75% of coefficients masked!

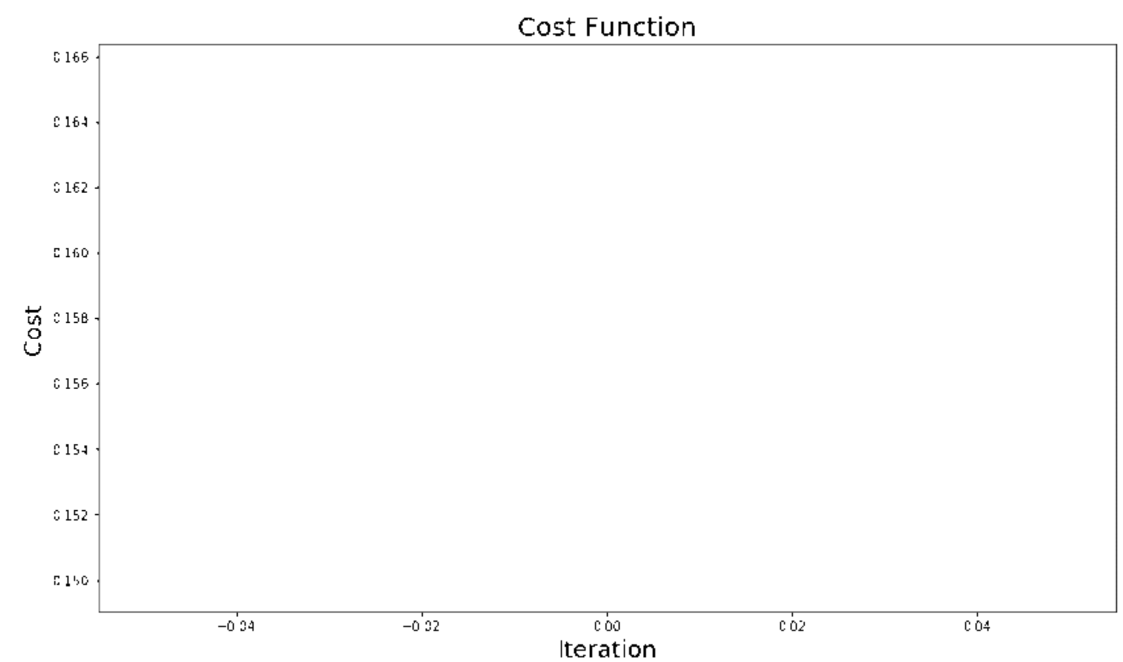


Sparse Representation via FFT

Compressed Sensing

$$\underset{\alpha}{\textbf{argmin}} \quad \frac{1}{2}\|y - H\phi\alpha\|_2^2 + \lambda\|\alpha\|_1$$

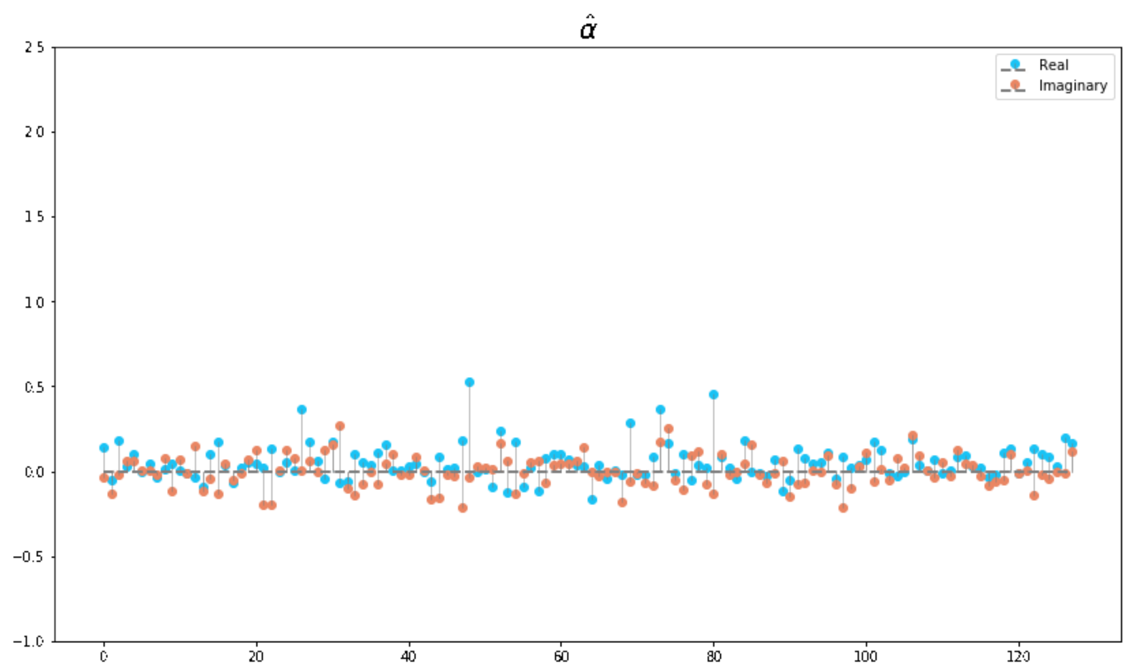


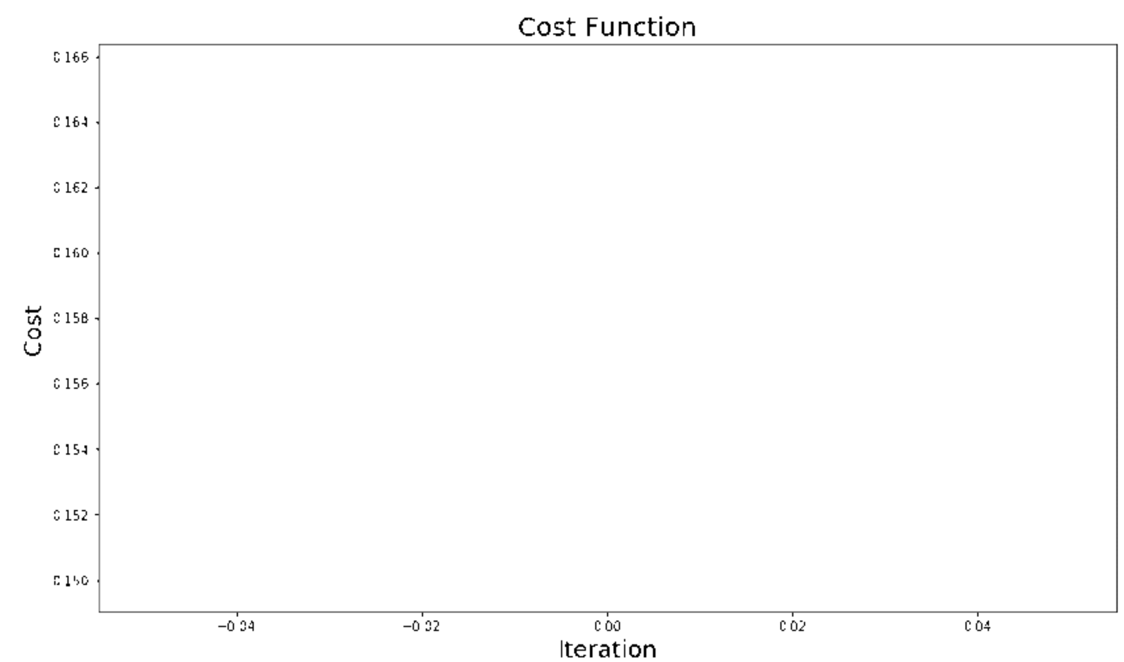


Near perfect signal reconstruction!

Compressed Sensing

$$\underset{\alpha}{\textbf{argmin}} \quad \frac{1}{2}\|y - H\phi\alpha\|_2^2 + \lambda\|\alpha\|_1$$





Near perfect signal reconstruction!

Outline

❖ PySAP

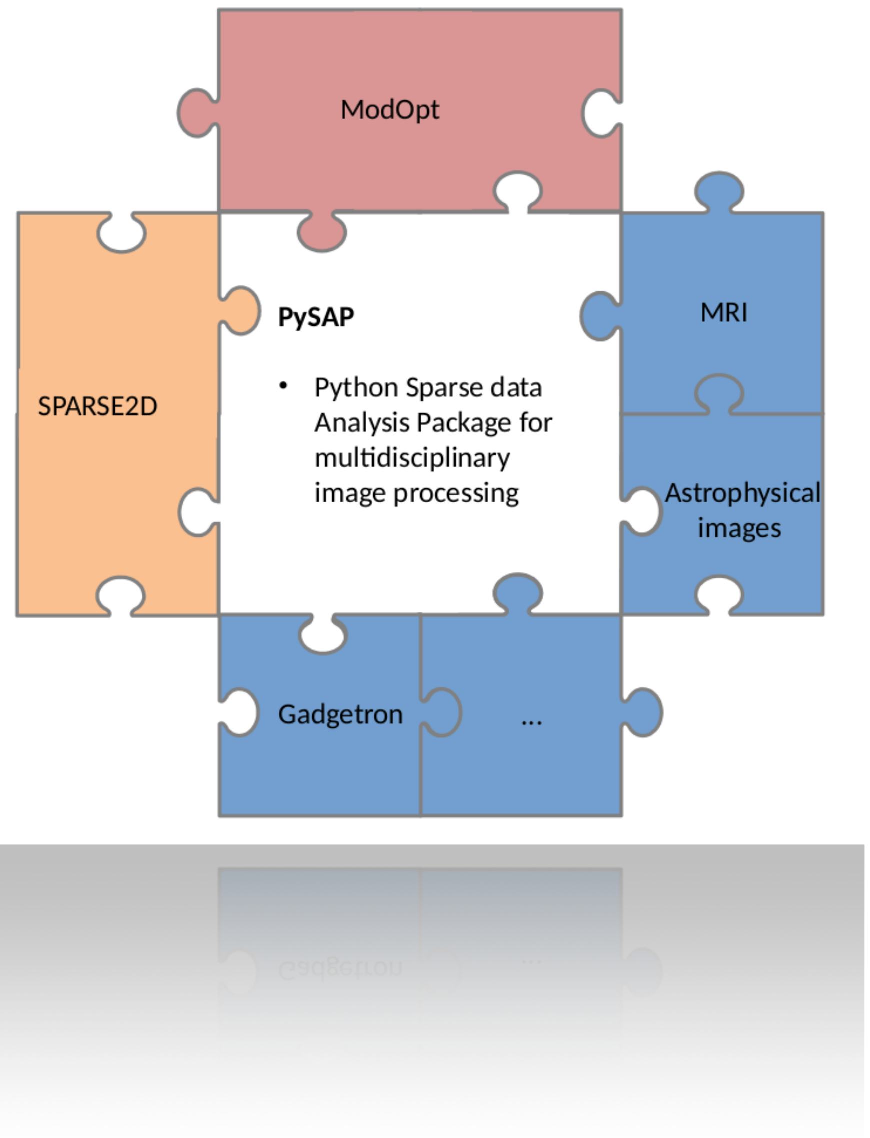
PySAP

Python Sparse Data Analysis Package

ModOpt

Sparse2D

Plug-Ins



https://github.com/cea-cosmic/pysap

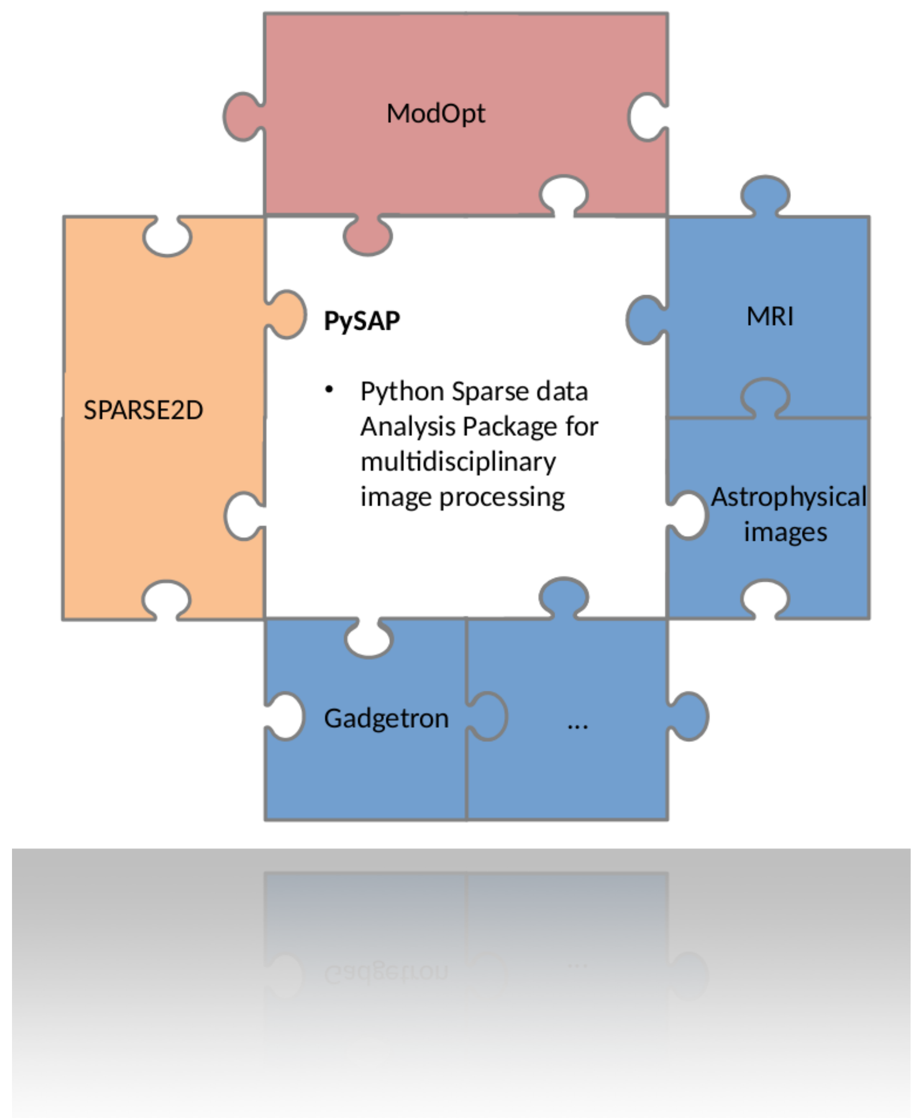
PySAP

Python Sparse Data Analysis Package

ModOpt

Sparse2D

Plug-Ins



https://github.com/cea-cosmic/pysap

Modular **Opt**imisation

- Optimisation Algorithms
  - Forward-Backward
  - FISTA
  - Generalized Forward-Backward
  - Condat-Vu
- Proximity Operators
  - Positivity
  - Hard Thresholding
  - Soft Thresholding
  - Low-Rank Approximation
- Linear Operators
  - Sparse2D
- Cost Function

→

class `modopt.opt.algorithms.GenForwardBackward`(*x, grad, prox_list, cost='auto', gamma_param=1.0, lambda_param=1.0, gamma_update=None, lambda_update=None, weights=None, auto_iterate=True*) [source]

Bases: `modopt.opt.algorithms.SetUp`

Generalized Forward-Backward Algorithm

This class implements algorithm 1 from [R2012]

Parameters:
- **x** (*list, tuple or np.ndarray*) – Initial guess for the primal variable
- **grad** (*class instance*) – Gradient operator class
- **prox_list** (*list*) – List of proximity operator class instances
- **cost** (*class or str, optional*) – Cost function class (default is 'auto'); Use 'auto' to automatically generate a costObj instance
- **gamma_param** (*float, optional*) – Initial value of the gamma parameter (default is 1.0)
- **lambda_param** (*float, optional*) – Initial value of the lambda parameter (default is 1.0)
- **gamma_update** (*function, optional*) – Gamma parameter update method (default is None)
- **lambda_update** (*function, optional*) – Lambda parameter parameter update method (default is None)
- **weights** (*list, tuple or np.ndarray, optional*) – Proximity operator weights (default is None)
- **auto_iterate** (*bool, optional*) – Option to automatically begin iterations upon initialisation (default is 'True')

https://github.com/cea-cosmic/ModOpt

Primal-Dual Splitting

$$1: \tilde{\mathbf{X}}_{k+1} = \mathrm{prox}_{\tau G}(\mathbf{X}_k - \tau\nabla F(\mathbf{X}_k) - \tau\mathcal{L}^*(\mathbf{Y}_k))$$

$$2: \tilde{\mathbf{Y}}_{k+1} = \mathbf{Y}_k + \varsigma\mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k) - \varsigma\mathrm{prox}_{K/\varsigma}\left(\frac{\mathbf{Y}_k}{\varsigma} + \mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k)\right)$$

$$3: (\mathbf{X}_{k+1}, \mathbf{Y}_{k+1}) := \xi(\tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{Y}}_{k+1}) + (1-\xi)(\mathbf{X}_k, \mathbf{Y}_k)$$

(Condat 2013, Vu 2013)

https://github.com/cea-cosmic/ModOpt

Primal-Dual Splitting

Gradient

$$1:\ \tilde{\mathbf{X}}_{k+1} = \mathrm{prox}_{\tau G}(\mathbf{X}_k - \tau\nabla F(\mathbf{X}_k) - \tau\mathcal{L}^*(\mathbf{Y}_k))$$

$$2:\ \tilde{\mathbf{Y}}_{k+1} = \mathbf{Y}_k + \varsigma\mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k) - \varsigma\mathrm{prox}_{K/\varsigma}\Big(\frac{\mathbf{Y}_k}{\varsigma} + \mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k)\Big)$$

$$3:\ (\mathbf{X}_{k+1}, \mathbf{Y}_{k+1}) := \xi(\tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{Y}}_{k+1}) + (1-\xi)(\mathbf{X}_k, \mathbf{Y}_k)$$

(Condat 2013, Vu 2013)

https://github.com/cea-cosmic/ModOpt

Primal-Dual Splitting

Proximity Operators

$$1: \tilde{\mathbf{X}}_{k+1} = \mathrm{prox}_{\tau G}(\mathbf{X}_k - \tau\nabla F(\mathbf{X}_k) - \tau\mathcal{L}^*(\mathbf{Y}_k))$$

$$2: \tilde{\mathbf{Y}}_{k+1} = \mathbf{Y}_k + \varsigma\mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k) - \varsigma\mathrm{prox}_{K/\varsigma}\left(\frac{\mathbf{Y}_k}{\varsigma} + \mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k)\right)$$

$$3: (\mathbf{X}_{k+1}, \mathbf{Y}_{k+1}) := \xi(\tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{Y}}_{k+1}) + (1-\xi)(\mathbf{X}_k, \mathbf{Y}_k)$$

(Condat 2013, Vu 2013)

https://github.com/cea-cosmic/ModOpt

Primal-Dual Splitting

Linear Operators

$$1:\ \tilde{\mathbf{X}}_{k+1} = \mathrm{prox}_{\tau G}(\mathbf{X}_k - \tau\nabla F(\mathbf{X}_k) - \tau\mathcal{L}^*(\mathbf{Y}_k))$$

$$2:\ \tilde{\mathbf{Y}}_{k+1} = \mathbf{Y}_k + \varsigma\mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k) - \varsigma\mathrm{prox}_{K/\varsigma}\left(\frac{\mathbf{Y}_k}{\varsigma} + \mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k)\right)$$

$$3:\ (\mathbf{X}_{k+1}, \mathbf{Y}_{k+1}) := \xi(\tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{Y}}_{k+1}) + (1-\xi)(\mathbf{X}_k, \mathbf{Y}_k)$$

(Condat 2013, Vu 2013)

https://github.com/cea-cosmic/ModOpt

Primal-Dual Splitting

Linear Operators

$$1:\ \tilde{\mathbf{X}}_{k+1} = \mathrm{prox}_{\tau G}(\mathbf{X}_k - \tau\nabla F(\mathbf{X}_k) - \tau\mathcal{L}^*(\mathbf{Y}_k))$$

$$2:\ \tilde{\mathbf{Y}}_{k+1} = \mathbf{Y}_k + \varsigma\mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k) - \varsigma\mathrm{prox}_{K/\varsigma}\left(\frac{\mathbf{Y}_k}{\varsigma} + \mathcal{L}(2\tilde{\mathbf{X}}_{k+1} - \mathbf{X}_k)\right)$$

$$3:\ (\mathbf{X}_{k+1}, \mathbf{Y}_{k+1}) := \xi(\tilde{\mathbf{X}}_{k+1}, \tilde{\mathbf{Y}}_{k+1}) + (1-\xi)(\mathbf{X}_k, \mathbf{Y}_k)$$

(Condat 2013, Vu 2013)

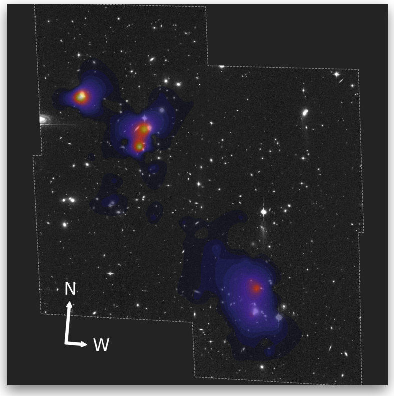
Sparse2D

https://github.com/cea-cosmic/ModOpt

Sparse Dictionaries in 2D

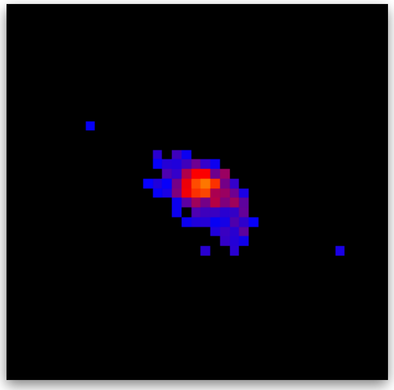
- PySAP provides Python bindings and wrappers for Sparse2D (C++) wavelet and curvet transforms.

Mass Mapping



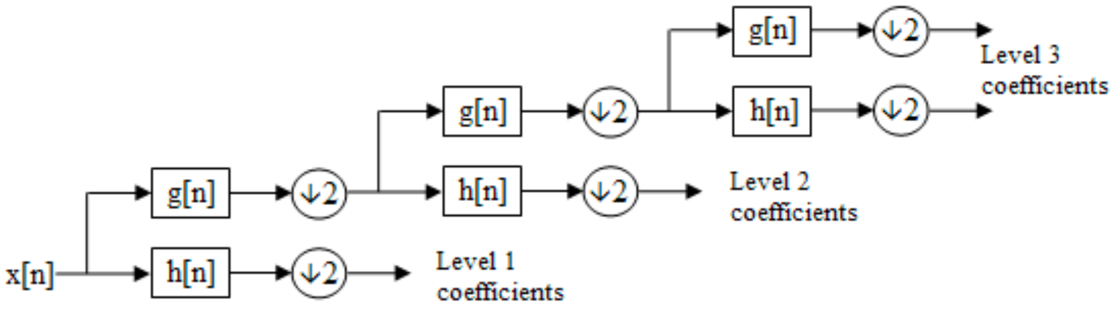
Peel et al. (2017)

Deconvolution

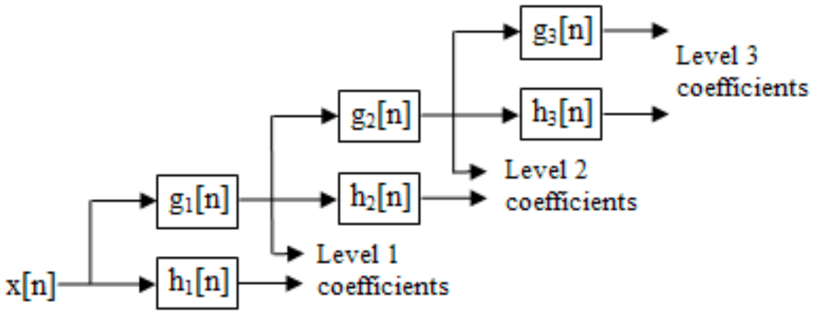


Farrens et al. (2017)

Discrete Wavelet transforms



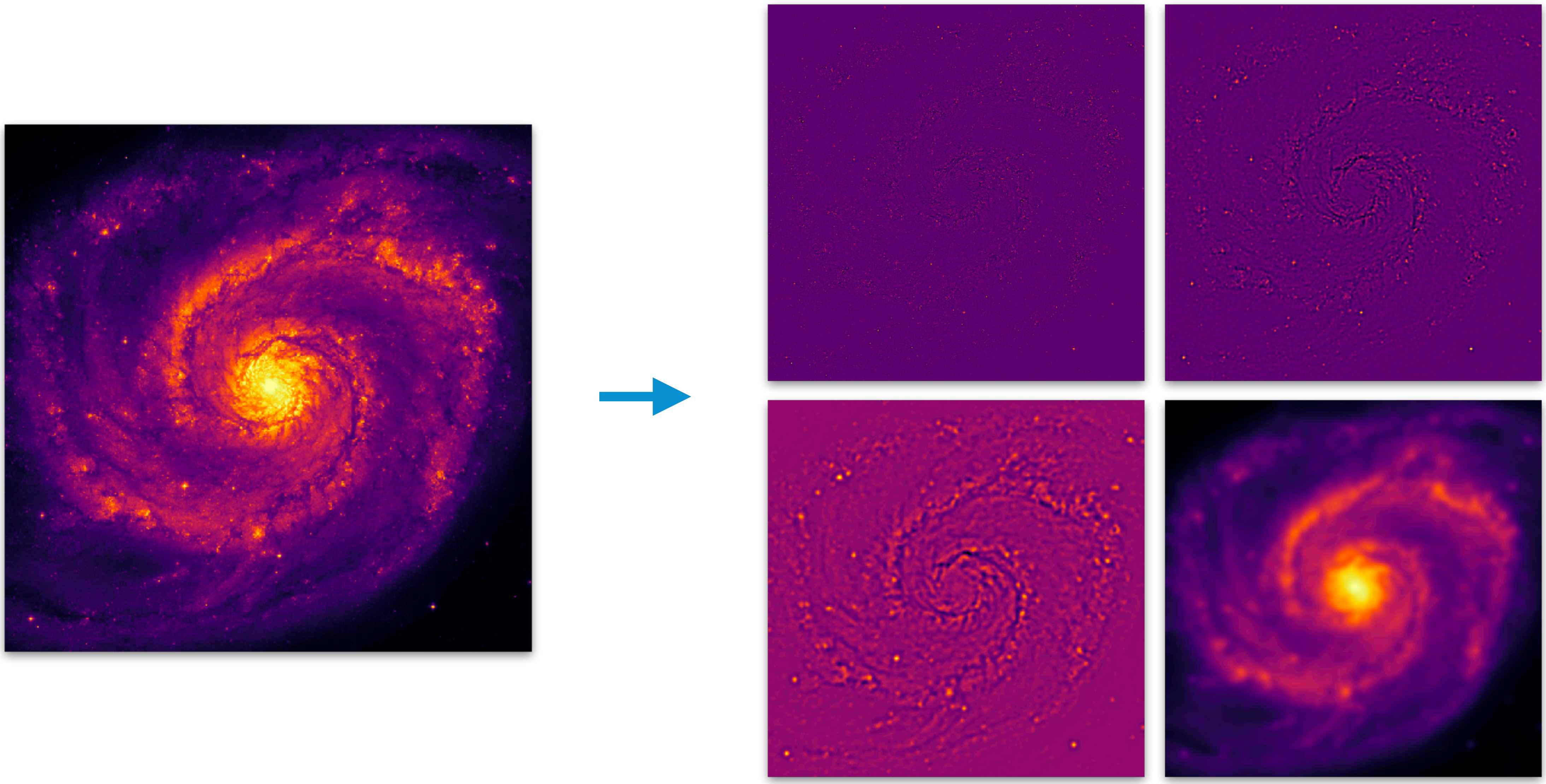
Undecimated Wavelet transforms



https://github.com/CosmoStat/Sparse2D

Sparse2D

Sparse Transforms in 2D



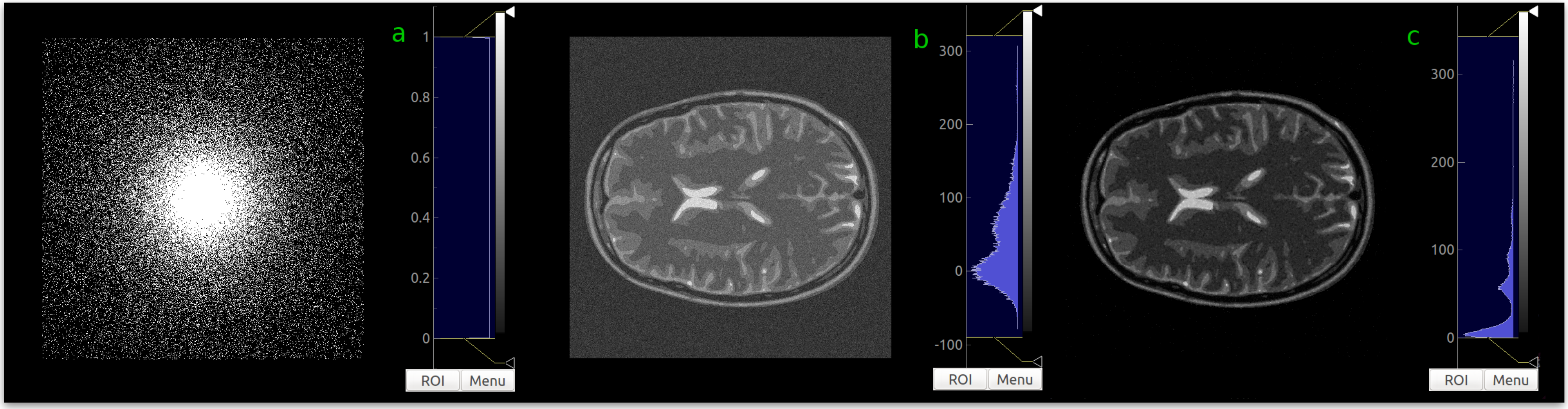
Starlet Transform (Starck et al. 2015a)

https://github.com/CosmoStat/Sparse2D

Plug-Ins

- Space for PySAP tools to be combined for specific applications.
- Gives visibility to contributors.
- Currently includes example applications to astrophysical and MRI data.

k-Space **Observed Image** **PySAP Denoising**

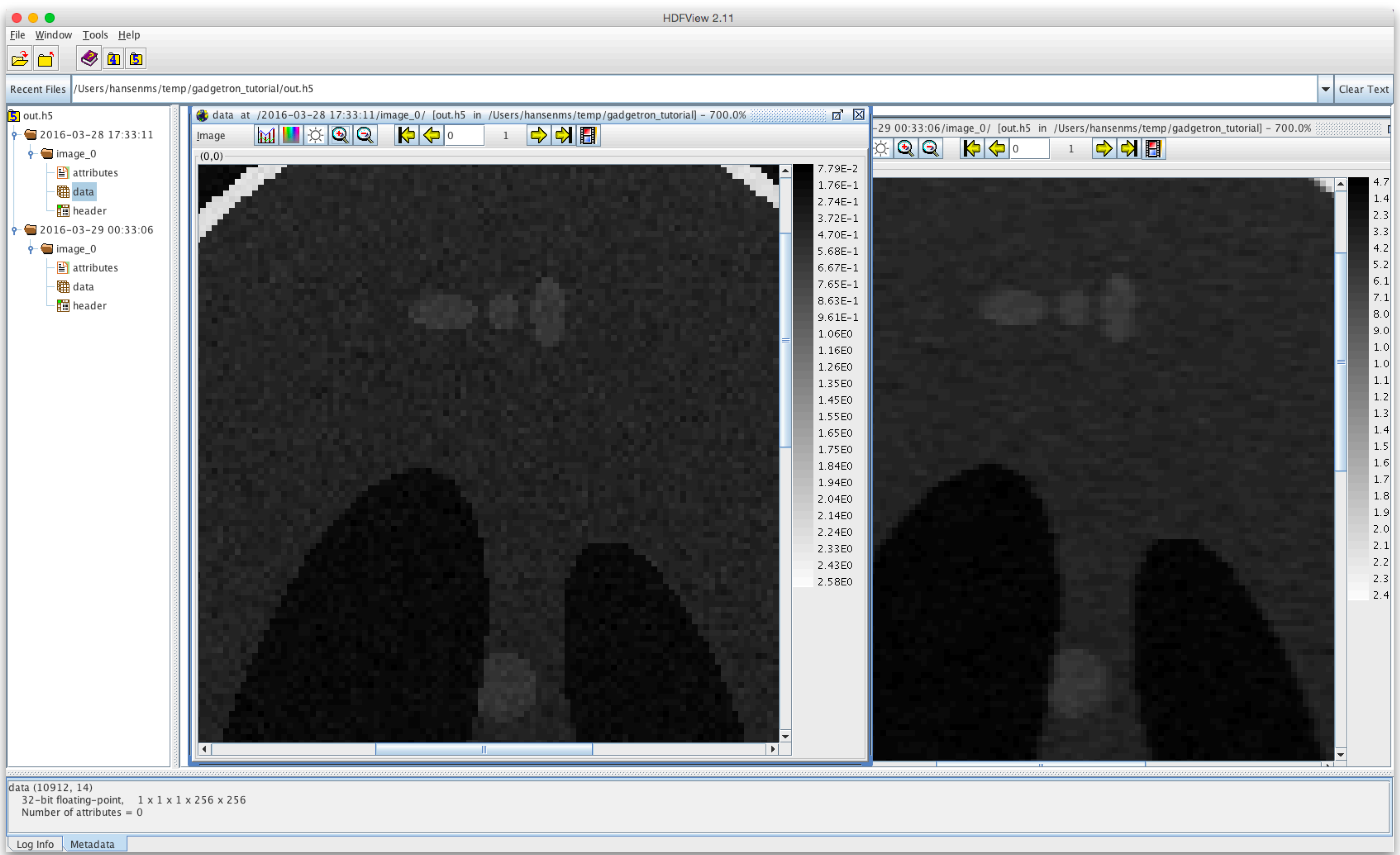


https://github.com/cea-cosmic/pysap

Plug-Ins

Gadgetron

An open source framework for medical image reconstruction



http://gadgetron.github.io/

Outline

❖ The Future

The Future

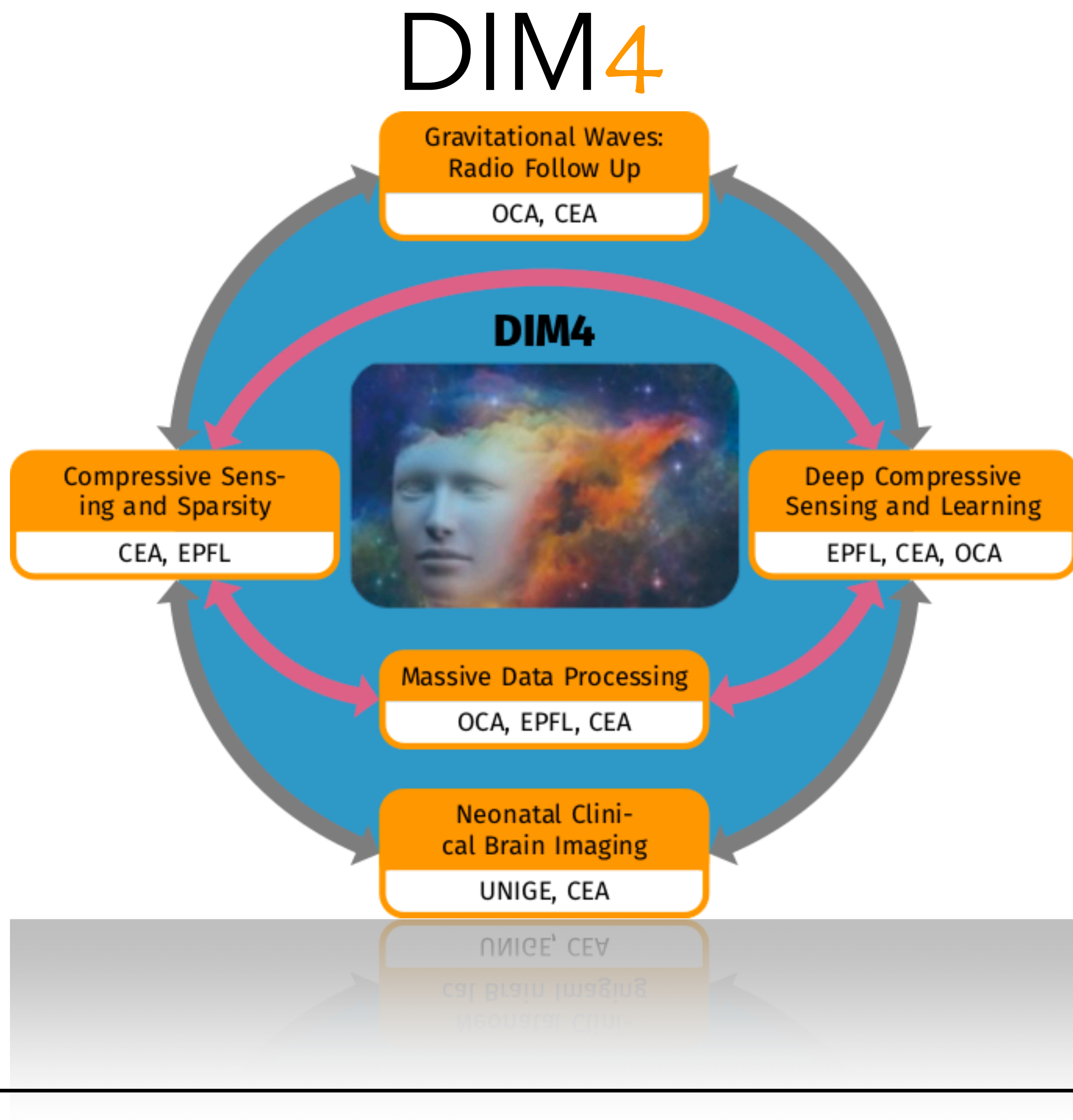
BabyBrainScope

Fast High Resolution Brain Imaging in Preterm Infants

ANR Projet PRCI

CES 45 — Mathematical and signal processing tools for life sciences

The Future



DIM4
Gravitational Waves: Radio Follow Up
OCA, CEA
DIM4
Compressive Sensing and Sparsity
CEA, EPFL
Deep Compressive Sensing and Learning
EPFL, CEA, OCA
Massive Data Processing
OCA, EPFL, CEA
Neonatal Clinical Brain Imaging
UNIGE, CEA

Summary

- COSMIC constitutes an exciting collaboration between NeuroSpin and CosmoStat for developing new software and new ideas for image processing.
- PySAP pre-release is available for beta testing.
- Through the Gadgetron system PySAP could be made readily available on virtually any MRI scanner around the world.
- The plug-in system opens up the software to various other image analysis application such as electron tomography.

https://github.com/cea-cosmic/pysap