Estimating Hyper-Parameters for Compressed Sensing

Dimitris Perdios

Signal Processing Laboratory (LTS5)
École Polytechnique Fédérale de Lausanne

NTDS – Final Presentation, February 27, 2017

Outline

Compressed sensing problem
Introduction

- Compressed sensing (CS) introduces a signal acquisition framework that goes beyond the traditional Nyquist sampling paradigm
- CS demonstrates that signals can be acquired using a small number of linear measurements and then recovered by solving a non-linear optimization problem

Problem

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n},$$

- $\boldsymbol{x} \in \mathbb{R}^N$ the vectorized (*i.e.* column-stacked) image (or patch) of $N = L \times L$ pixels to be recovered
- $\boldsymbol{y} \in \mathbb{R}^M$ the measurement (such that $M < N$)
- $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ the linear measurement matrix
- $\boldsymbol{n} \in \mathbb{R}^N$ the noise
- Ill-posed problem!

Optimization problem

$$\boldsymbol{x}_{opt} = \arg\min_{\boldsymbol{x}} \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda \|\boldsymbol{x}\|_1$$

- λ the sparsity parameter

Solving inverse problem
Iterative solvers

Solver choice

- Many iterative solvers exist: ISTA, AMP, ADMM, LASSO, primal-dual, etc..
- We focus on ISTA, let's start easy!

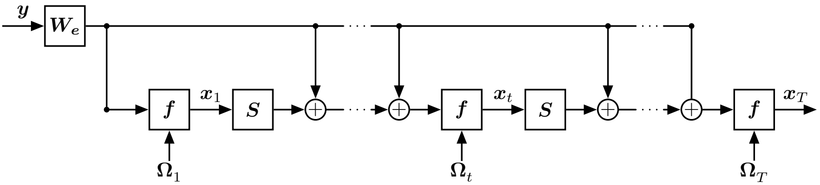
Main drawback of iterative solvers

- Might need a lot of iterations!
- How to determine a good λ (which is directly linked to the number of iterations...)
- By hand, can be quite painful (even horrible...)

The idea

- Use of neural networks
- And learn the (optimal?) λ
- Learned ISTA, *i.e.* LISTA[Gregor and LeCun, ICML2010]

Proposed approach
Classic LISTA



- Where $\boldsymbol{f}$ is the soft thresholding ($\boldsymbol{h}$) and each set $\boldsymbol{\Omega}_t$ is composed of the thresholding parameter θ
- Really bad for images (not directly sparse)

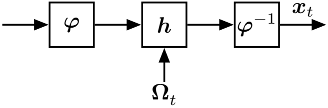
Ideas

- Decompose the image in a wavelet basis or wavelet frame
- Not only learn a single thresholding parameter θ, but a vector of thresholding parameters $\boldsymbol{\theta}$ (*i.e.* for each pixels/coefficients!)

Proposed approach
LISTA extended to a single wavelet decomposition

Threshold in a wavelet basis

- Decompose
- Threshold (soft thresholding)
- Reconstruct



Advantage

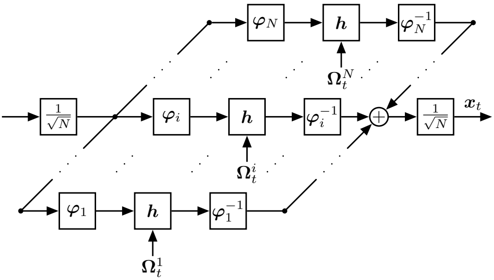
- Images have a sparse representation in the wavelets

Proposed approach

LISTA extended to a wavelet frame decomposition

Threshold in a wavelet frame

- Constructed by **multiple** wavelet bases



Advantage

- Learn the θ^i for each basis

Cooking recipe

- Measurement matrix: Gaussian random satisfying the Restricted Isometry Property[Rauhut, RSCAM2010]
- Compression: keeping only 40% of the data
- Layers: 20 (*i.e.* iterations of the algorithm)
- Epochs: 20
- Patch size: 32×32
- Batch size: 16
- Optimizer: Adam
- Learning rate: 10^{-2}
- Learned $\boldsymbol{\theta}$ are the same for each layers
- $\boldsymbol{\theta}$ initialized at: 10^{-3}

Results
Set-up and test cases

Training and validation sets

- Standard images: ‘cameraman’, ‘boat’, ‘baboon’, ‘airplane’, ‘arctichare’, ‘barbara’, ‘zelda’, ‘goldhill’, ‘cat’
- Train set (4/5): 1716 patches
- Validation set (1/5): 429 patches

Test set

- Standard image: ‘peppers’
- 256 patches

Decomposition test cases

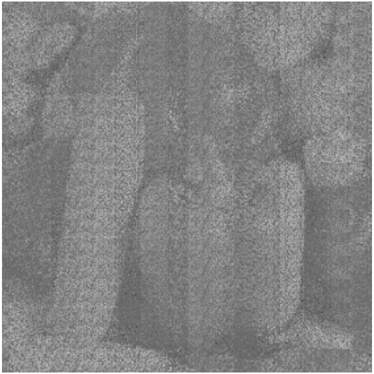
- ‘dirac’
- Single Daubechies wavelet: ‘db4’
- Daubechies wavelet frame: ‘db1’, ‘db4’, ‘db8’

Results

'dirac'

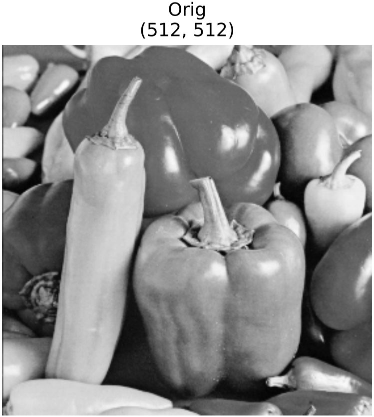
Orig
(512, 512)

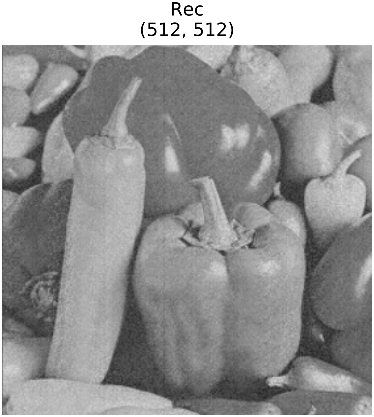
Rec
(512, 512)



Results

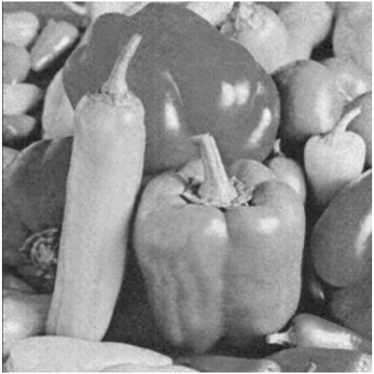
Single Daubechies wavelet: 'db4'





Results

Daubechies wavelet frame: ‘db1’, ‘db4’, ‘db8’



Next tasks and perspectives

- Add classical image distance metrics (*e.g.* PSNR, SSIM)
- Extend training and testing sets
- Implement the wavelet decomposition and reconstruction in TensorFlow
- Extend to more wavelets decompositions
- Compare against ‘best’ ISTA reconstructions
  - For the same number of iterations
  - Up to a specified reconstruction quality
- Quantify the benefit of learning the thresholding parameters
- Learn thresholding parameters for each layer
- Learn the non-linearity (*e.g.* piecewise-linear)
- Apply to other algorithms
- (long term) apply to ultrasound image reconstruction

THANK YOU FOR YOUR ATTENTION!

Dimitris Perdios

dimitris.perdios@epfl.ch

Signal Processing Laboratory (LTS5)

http://lts5www.epfl.ch

École Polytechnique Fédérale de Lausanne