L0Soft: ℓ_0 Minimization via Soft Thresholding

Mostafa Sadeghi[1], Fateme Ghayem[1], Massoud Babaie-Zadeh[1]
Saikat Chatterjee[2], Mikael Skoglund[2]
Christian Jutten[3]

[1]Electrical Engineering Department, Sharif University of Technology, Tehran, Iran
[2]Communication Theory Lab, KTH, Royal Institute of Technology, Stockholm, 10044, Sweden
[2]GIPSA-Lab, Grenoble, and Institut Universitaire de France, France

September 2019

1. Introduction
   - Sparse representation
2. Proposed algorithm
   - Main idea
   - Problem formulation
   - Smooth approximation of sign
   - Final problem
   - Algorithm
3. Experimental results
4. Conclusions

1 Introduction
- Sparse representation

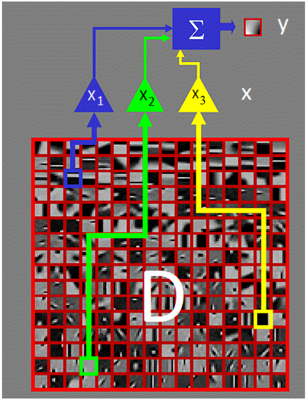
2 Proposed algorithm

3 Experimental results

4 Conclusions

Sparse representation

$\mathbf{y} \approx x_1\mathbf{d}_1 + x_2\mathbf{d}_2 + \ldots + x_n\mathbf{d}_m = \mathbf{D}\mathbf{x}$ most x_i's are zero



- Signal restoration:

$$\boxed{\mathbf{z} = \mathbf{H}\mathbf{y} + \mathbf{e}}$$

De-noising ($\mathbf{H} =$ identity), inpainting ($\mathbf{H} =$ random rows of identity), de-bluring ($\mathbf{H} =$ blurring matrix), super resolution ($\mathbf{H} =$ down sampling matrix), ...

$$\mathbf{y} \simeq \mathbf{D}\mathbf{x}, \quad \mathbf{x} : \text{sparse}$$

$$\boxed{\min_{\mathbf{y},\mathbf{x}} \ \|\mathbf{z} - \mathbf{H}\mathbf{y}\|_2^2 + \alpha\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2^2 + \beta\|\mathbf{x}\|_1}$$

Sparse recovery algorithms

- Greedy. Pick atoms sequentially.

Set $k=0$, $\mathbf{r}^0=\mathbf{y}$, and $\mathcal{I}_0=\varnothing$. Repeat:

$$\begin{cases} \mathcal{I}_{k+1} = \mathcal{I}_k \cup \left\{ i \mid |\mathbf{d}_i^T \mathbf{r}^k| \geq \tau_k \right\} & (\text{atom selection}) \\ \mathbf{x}^k_{\mathcal{I}_{k+1}} = \operatorname{argmin}_{\mathbf{x}} \|\mathbf{y} - \mathbf{D}_{\mathcal{I}_{k+1}} \mathbf{x}\|_2 & (\text{projection}) \\ \mathbf{r}^{k+1} = \mathbf{y} - \mathbf{D}_{\mathcal{I}_{k+1}} \mathbf{x}^k_{\mathcal{I}_{k+1}} & (\text{residual update}) \\ k \to k+1 \end{cases}$$

Examples:

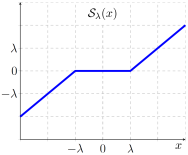
- OMP [Pati et al., 1993]: $\tau_k = \max_i \ |\mathbf{d}_i^T \mathbf{r}^k|$.
- GOMP [Wang et al., 2012]: $\tau_k = |\mathbf{d}_i^T \mathbf{r}^k|_N = N$th largest correlation.
- SP [Dai et al., 2009]: $\tau_k = |\mathbf{d}_i^T \mathbf{r}^k|_{2s} = 2s$th largest correlation ($s =$ sparsity level)+ pruning.

Sparse recovery algorithms

- Thresholding based algorithms

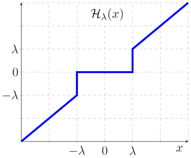
$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_1$$

$$\mathbf{x}^{k+1} = \mathcal{S}_{\mu_k\lambda}(\mathbf{x}^k - \mu_k(\mathbf{D}\mathbf{x}_k - \mathbf{y}))$$



$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2 + \lambda\|\mathbf{x}\|_0$$

$$\mathbf{x}^{k+1} = \mathcal{H}_{\mu_k\lambda}(\mathbf{x}^k - \mu_k(\mathbf{D}\mathbf{x}_k - \mathbf{y}))$$



$$\mu_k \in (0, 1/\sigma_{\max}(\mathbf{D}))$$

Examples: IST [Daubechies et al., 2004], GPSR [Figueiredo et al., 2007], IHT [Blumensath and Davies, 2009], ISP-Hard [Sadeghi and Babaie-Zadeh, 2016], TST [Maleki and Donoho, 2010], NESTA [Becker et al., 2009]

- ℓ_0 norm approximation. Approximate ℓ_0 norm with a smooth function.

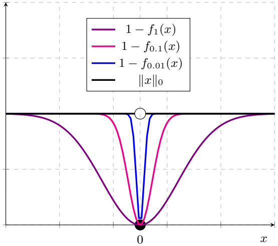
  Smoothed L0 (SL0) [Mohimani et al., 2009], SCSA [Malek-Mohammadi et al., 2016]

$$F_\sigma(\mathbf{x}) = n - \sum_{i=1}^{n} f_\sigma(x_i)$$

$$\min_{\mathbf{x}} F_\sigma(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{y} = \mathbf{D}\mathbf{x}$$

$$f_\sigma(x) = \exp(-\frac{x^2}{\sigma^2})$$

When $\sigma \to 0: \; F_\sigma(\mathbf{x}) \to \|\mathbf{x}\|_0$



$$\begin{cases} k = 0 \\ \mathbf{x}^0 = \mathbf{D}^\dagger \mathbf{y} \\ \text{For } i = 1, 2, \ldots \\ \begin{cases} \text{For } j = 1, 2, \ldots \\ \mathbf{x}^{k+1} = \mathbf{x}^k - \mu_{\sigma_i} \nabla \|\mathbf{x}_k\|_{\sigma_i} \\ \mathbf{x}^{k+1} = \mathbf{x}^{k+1} - \mathbf{D}^\dagger(\mathbf{D}\mathbf{x}_{k+1} - \mathbf{y}) \\ k \leftarrow k + 1 \\ \text{End} \end{cases} \\ \sigma_{i+1} = \sigma_i \cdot c \quad (0 < c < 1) \\ \text{End} \end{cases}$$

1 Introduction

2 Proposed algorithm
- Main idea
- Problem formulation
- Smooth approximation of sign
- Final problem
- Algorithm

3 Experimental results

4 Conclusions

ℓ_0 minimization via soft-thresholding

- Problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon$$

Main idea

Write ℓ_0 norm as sum of absolute values of entries' sign:

$$\|\mathbf{x}\|_0 = \sum_{i=1}^{n} |\mathsf{sgn}(x_i)|$$

or, equivalently:

$$\|\mathbf{x}\|_0 = \|\mathbf{z}\|_1, \quad \mathbf{z} = \mathsf{sgn}(\mathbf{x})$$

1 Introduction

2 Proposed algorithm
- Main idea
- Problem formulation
- Smooth approximation of sign
- Final problem
- Algorithm

3 Experimental results

4 Conclusions

ℓ_0 minimization via soft-thresholding

- Equivalent problem:

$$\min_{\mathbf{x},\mathbf{z}} \ \|\mathbf{z}\|_1 \ \text{ s.t. } \begin{cases} \mathbf{z} = \mathrm{sgn}(\mathbf{x}) \\ \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon \end{cases}$$

- Final problem to solve:

Using penalty method, we solve the following approximate problem:

$$\boxed{\min_{\mathbf{x},\mathbf{z}} \ \|\mathbf{z}\|_1 + \frac{1}{2\alpha}\|\mathbf{z} - \mathrm{sgn}(\mathbf{x})\|_2^2 + \delta_\epsilon(\mathbf{x})}$$

  - $\alpha > 0$ is a penalty parameter
  - $\delta_\epsilon(\mathbf{x}) = 0$ if $\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon$ and ∞ otherwise.

1 Introduction

2 Proposed algorithm
- Main idea
- Problem formulation
- Smooth approximation of sign
- Final problem
- Algorithm

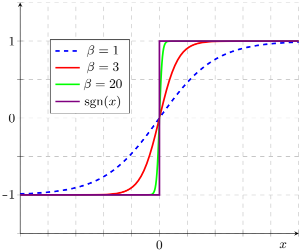
3 Experimental results

4 Conclusions

- Smoothed sign function:

Because the sign function is non-smooth, we approximate it by a smooth function:

$$f_\beta(x) \triangleq \tanh(\beta x) = \frac{\exp(2\beta x) - 1}{\exp(2\beta x) + 1}$$



☞ Larger values of β give tighter approximation

1. Introduction
2. Proposed algorithm
   - Main idea
   - Problem formulation
   - Smooth approximation of sign
   - Final problem
   - Algorithm
3. Experimental results
4. Conclusions

ℓ_0 minimization via soft-thresholding

- Final problem:

$$\boxed{\min_{\mathbf{x},\mathbf{z}} \ \alpha\|\mathbf{z}\|_1 + \frac{1}{2}\|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 + \delta_\epsilon(\mathbf{x})}$$

- Algorithm:

We adopt a proximal alternating linearized minimization (PALM) approach [Bolte et al., 2014]:

$$\min_{\mathbf{x},\mathbf{z}} \ \left\{ H(\mathbf{x},\mathbf{z}) \triangleq F(\mathbf{x},\mathbf{z}) + g(\mathbf{x}) + h(\mathbf{z}) \right\}$$

$$\begin{cases} F(\mathbf{x},\mathbf{z}) = \frac{1}{2}\|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 & \text{smooth, gradient Lipschitz part} \\ g(\mathbf{x}) = \delta_\epsilon(\mathbf{x}) & \text{non-smooth part} \\ h(\mathbf{z}) = \alpha\|\mathbf{z}\|_1 & \text{non-smooth part} \end{cases}$$

1 Introduction

2 Proposed algorithm
- Main idea
- Problem formulation
- Smooth approximation of sign
- Final problem
- Algorithm

3 Experimental results

4 Conclusions

ℓ_0 minimization via soft-thresholding

- $F(\mathbf{x}, \mathbf{z})$ is gradient Lipschitz. That is, there exist some $L_x, L_z > 0$ such that $\forall \mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{v}$:

$$\begin{cases} \|\nabla_x F(\mathbf{x}, \mathbf{z}) - \nabla_x F(\mathbf{u}, \mathbf{z})\|_2 \leq L_x \|\mathbf{x} - \mathbf{u}\|_2 \\ \|\nabla_z F(\mathbf{x}, \mathbf{z}) - \nabla_z F(\mathbf{x}, \mathbf{v})\|_2 \leq L_z \|\mathbf{z} - \mathbf{v}\|_2 \end{cases}$$

☞ It can be shown that $L_z = 1$ and $L_x = (3 + 2|z|) \cdot \beta^2$ satisfy the above conditions.

- Final algorithm: Solve the following problem, using PALM, and for a decreasing sequence of α:

$$\boxed{\min_{\mathbf{x}, \mathbf{z}} \ \alpha \|\mathbf{z}\|_1 + \frac{1}{2} \|\mathbf{z} - f_\beta(\mathbf{x})\|_2^2 + \delta_\epsilon(\mathbf{x})}$$

ℓ_0 minimization via soft-thresholding

Algorithm 2 L0Soft for

Inputs: $\mathbf{y}$, $\mathbf{A}$, $(\mathbf{x}_0, \mathbf{z}_0)$, ϵ, α_1, w, c
for $j = 1, 2, \cdots$ **do**
  $(\mathbf{x}_j, \mathbf{z}_j) = \text{PALM}(\mathbf{y}, \mathbf{A}, (\mathbf{x}_{j-1}, \mathbf{z}_{j-1}), \epsilon, \alpha_j, w)$
  $\alpha_{j+1} = c \cdot \alpha_j$
end for
Output: $\mathbf{x}_j$ =0

Algorithm 1 PALM (with inertial)

Inputs: $\mathbf{y}$, $\mathbf{A}$, $(\mathbf{x}_0, \mathbf{z}_0)$, ϵ, α, w
for $k = 0, 1, \cdots$ **do**
  $\mathbf{z}_{k+1} = S_{\mu_z \cdot \alpha}\big((1 - \mu_z) \cdot \mathbf{z}_k + \mu_z \cdot f_\beta(\mathbf{x}_k)\big)$
  $\hat{\mathbf{x}}_k = \mathbf{x}_k + w \cdot (\mathbf{x}_k - \mathbf{x}_{k-1})$
  $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{C}_\epsilon}\big(\hat{\mathbf{x}}_k - \mu_x \cdot \nabla_x F(\mathbf{x}_k, \mathbf{z}_{k+1})\big)$
end for
Output: $(\mathbf{x}_k, \mathbf{z}_k)$ =0

1 Introduction

2 Proposed algorithm

3 Experimental results

4 Conclusions

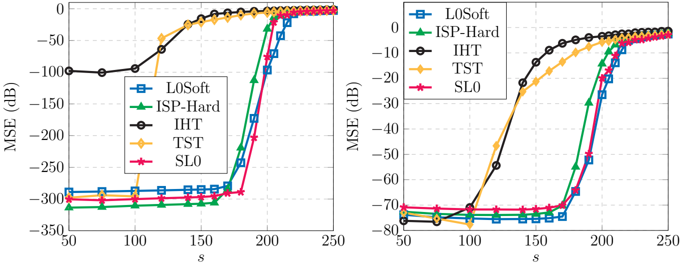
Experimental results

- Synthetic data:
  - Generate sparse signal, $\mathbf{x}$, of length $n = 1000$ from a Bernoulli-Gaussian distribution with s number of non-zero entries
  - Generate a random measurement matrix $\mathbf{A}$ with entries from normal distribution
  - Take $m = 400$ measurements from $\mathbf{x}$ as $\mathbf{y} = \mathbf{A}\mathbf{x}$ and add Gaussian noise
  - Apply different algorithms to estimate $\mathbf{x}$ from noisy $\mathbf{y}$
- Compressed image recovery:
  - Take some 32×32 image $\mathbf{X}$ and vectorized it to 1024×1 vector $\mathbf{x}$
  - Take random measurements $\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{a}$, $\mathbf{\Phi}$ = Gaussian, $\mathbf{\Psi} = 1024 \times 4096 =$ DCT matrix
  - Estimate original image by solving:

$$\hat{\mathbf{x}} = \mathbf{\Phi} \cdot \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{a}\|_0 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{a}$$

Experimental results

Synthetic data:



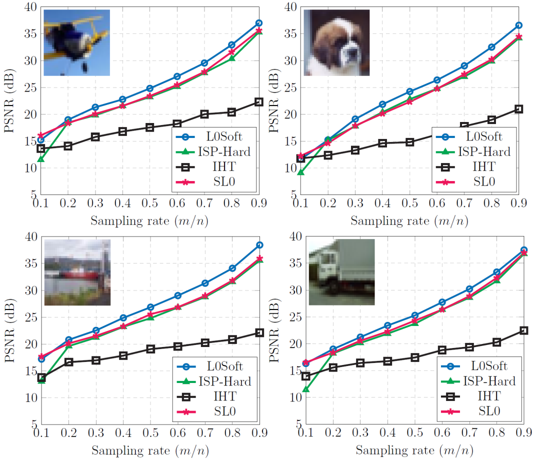
(a) (b)

Figure: Average MSEs (dB) obtained by different algorithms versus number of non-zeros (s), when recovering sparse signals of length $n = 1000$ from $m = 400$ (a) noiseless and (b) noisy ($\sigma = 0.001$) Gaussian measurements.

- L0Soft is better in the noiseless, less sparse as well as noisy, more sparse cases

Experimental results

Compressed image recovery:



1 Introduction

2 Proposed algorithm

3 Experimental results

4 Conclusions

Conclusions

- A new algorithm was introduced for ℓ_0 minimization
- The proposed algorithm relies on replacing ℓ_0 function with an equivalent definition based on absolute values of entries' sign
- Using penalty methods, ℓ_0 minimization was converted to an ℓ_1 minimization
- Proximal algorithms were used to solve the new problem
- Experimental results confirm the superiority of the proposed method over existing algorithms

Thank You For Your Attention!