Recursive Compressive Sampling

Nick Freris, Orhan Öçal

EPFL
LCAV

April 25, 2013

Outline

- Background
- Recursive Compressive Sampling
  - Problem Formulation
  - Complexity
  - Robustness
- Simulations

Definitions

k-sparse:
For $\mathbf{x} \in \mathbb{R}^n$, define the support of $\mathbf{x}$ as $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0\}$

$$\mathbf{x} \in \mathcal{R}^{10}$$

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & \boxed{3} & 0 & \boxed{-1} & 0 & 0 & \boxed{5} & 0 \end{bmatrix}^T$$

Definitions

k-sparse:
For $\mathbf{x} \in \mathbb{R}^n$, define the support of $\mathbf{x}$ as $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0\}$
Define $\|\mathbf{x}\|_0$ as the cardinality of the support, i.e., $\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})|$

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & \boxed{3} & 0 & \boxed{-1} & 0 & 0 & \boxed{5} & 0 \end{bmatrix}^T$$

$\mathbf{x} \in \mathcal{R}^{10}$

$\|\mathbf{x}\|_0 = 3$

Definitions

k-sparse:

For $\mathbf{x} \in \mathbb{R}^n$, define the support of $\mathbf{x}$ as $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0\}$

Define $\|\mathbf{x}\|_0$ as the cardinality of the support, i.e., $\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})|$

$\mathbf{x}$ is called k-sparse if $\|\mathbf{x}\|_0 \leq k$

Typically $k << n$

$$\mathbf{x} = \begin{bmatrix} 0 & 0 & 0 & \boxed{3} & 0 & \boxed{-1} & 0 & 0 & \boxed{5} & 0 \end{bmatrix}^T$$

$\mathbf{x} \in \mathcal{R}^{10}$

$\|\mathbf{x}\|_0 = 3$

3-sparse

Matrix Properties

Restricted Isometry Property (RIP):
Matrix $\mathbf{A} \in \mathcal{R}^{m \times n}$ satisfies RIP if
$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2, \quad \forall \mathbf{x}$ k-sparse.

Matrices satisfying RIP with high probability:

- iid Gaussian entries $\sim \mathcal{N}(0, \frac{1}{m})$
- iid Bernoulli entries $\sim \mathcal{U}(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}})$
- randomly selected rows from DFT(n) matrix

Restricted Isometry Property (RIP):
Matrix $\mathbf{A} \in \mathcal{R}^{m \times n}$ satisfies RIP if
$(1-\delta_k)\|\mathbf{x}\|_2^2 \le \|\mathbf{A}\mathbf{x}\|_2^2 \le (1+\delta_k)\|\mathbf{x}\|_2^2, \quad \forall \mathbf{x}$ k-sparse.

Matrices satisfying RIP with high probability:

- iid Gaussian entries $\sim \mathcal{N}(0, \frac{1}{m})$
- iid Bernoulli entries $\sim \mathcal{U}(-\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}})$
- randomly selected rows from DFT(n) matrix

Mutual Coherence:
For $\mathbf{A} \in \mathcal{R}^{m \times n}$

$$\mu(\mathbf{A}) = \max_{1 \le i,j \le n,\ i \ne j} \frac{|\mathbf{a}_i^\top \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \cdot \|\mathbf{a}_j\|_2}$$

CS - Noiseless Case

Setting:

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$
$$\mathbf{y} \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{m \times n},\ m << n$$
$$\mathbf{x} \text{ sparse}$$

Optimization Problem:

$$\min \|\mathbf{x}\|_0$$
$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}$$

(Combinatorial - Intractable)

CS - Noiseless Case

Setting:

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$
$$\mathbf{y} \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{m \times n},\ m << n$$
$$\mathbf{x} \text{ sparse}$$

Optimization Problem:

$$\min \|\mathbf{x}\|_0$$
$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}$$

(Combinatorial - Intractable)

Basis Pursuit:
If $\mathbf{A} \in \mathbb{R}^{m \times n}$ satisfies RIP then $\mathbf{x}$ can be recovered by [CT05]

$$\min \|\mathbf{x}\|_1$$
$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}$$

(Linear Program)

CS - Noiseless Case

Setting:

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$
$$\mathbf{y} \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{m \times n},\ m << n$$
$$\mathbf{x} \text{ sparse}$$

Optimization Problem:

$$\min \|\mathbf{x}\|_0$$
$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}$$
(Combinatorial - Intractable)

Basis Pursuit:

If $\mathbf{A} \in \mathbb{R}^{m \times n}$ satisfies RIP then $\mathbf{x}$ can be recovered by [CT05]

$$\min \|\mathbf{x}\|_1$$
$$\text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}$$
(Linear Program)

for $m > Ck\log(n),\ C(\mathbf{A}) \sim \mu^2$

CS - Noisy Case

Setting:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$$
$$\mathbf{y} \in \mathbb{R}^m,\ \mathbf{A} \in \mathbb{R}^{m \times n},\ m << n$$
$$\mathbf{x} \text{ sparse}$$

Goal:

$$\text{recover } \mathbf{x} \in \mathbb{R}^n$$

LASSO:

$$\begin{aligned} &\text{minimize} && \|\mathbf{x}\|_1 \\ &\text{subject to} && \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon. \end{aligned}$$

Theorem ([CW08])

Assuming the isometry constant of $\mathbf{A}$ *for* $2k$*-sparse signals is* $\delta_{2k} \leq \sqrt{2} - 1$, *the solution* $\mathbf{x}^*$ *to* (6) *obeys:*

$$\|\mathbf{x}^* - \mathbf{x}\|_2 \leq \; C_0 \cdot \|\mathbf{x} - \mathbf{x}_k\|_1 / \sqrt{k} \; + \; C_1 \cdot \epsilon$$

where $\mathbf{x}_k$ *is the vector obtained by preserving* k*-many elements of* $\mathbf{x}$ *having highest absolute magnitude, for constants* C_0 *and* C_1 *that depend on the matrix* $\mathbf{A}$.

Equivalent form of LASSO:

$$\text{minimize} \quad \|\mathbf{Ax} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1$$

$\lambda > 0$: ℓ_1 regularization $\rightarrow$ sparse solutions

Least Absolute Shrinkage and Selection Oprerator

Theorem ([CW08])

Assuming the isometry constant of $\mathbf{A}$ *for* $2k$*-sparse signals is* $\delta_{2k} \leq \sqrt{2} - 1$*, the solution* $\mathbf{x}^*$ *to* (6) *obeys:*

$$\|\mathbf{x}^* - \mathbf{x}\|_2 \leq \underbrace{C_0 \cdot \|\mathbf{x} - \mathbf{x}_k\|_1/\sqrt{k}}_{\text{Model Mismatch}} + \underbrace{C_1 \cdot \epsilon}_{\text{Noise}}$$

where $\mathbf{x}_k$ *is the vector obtained by preserving* k*-many elements of* $\mathbf{x}$ *having highest absolute magnitude, for constants* C_0 *and* C_1 *that depend on the matrix* $\mathbf{A}$*.*

Equivalent form of LASSO:

$$\text{minimize} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

$\lambda > 0$: ℓ_1 regularization $\rightarrow$ sparse solutions

Iterative Algorithm for LASSO

Iterative Soft Thresholding Algorithm [DDDM04]:

LASSO Objective Function

$G(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1$

Define:

$$J_k(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1 + (\mathbf{x} - \mathbf{x}_k)^{\mathsf{T}} \left(\alpha\mathbf{I} - \mathbf{A}^{\mathsf{T}}\mathbf{A}\right) (\mathbf{x} - \mathbf{x}_k)$$

Iterative Algorithm for LASSO

Iterative Soft Thresholding Algorithm [DDDM04]:

LASSO Objective Function

$G(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1$

Define:

$$J_k(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 + (\mathbf{x} - \mathbf{x}_k)^{\mathsf{T}} \underbrace{\left(\alpha \mathbf{I} - \mathbf{A}^{\mathsf{T}}\mathbf{A}\right)}_{\alpha \geq \mathrm{maxeig}(\mathbf{A}^{\mathsf{T}}\mathbf{A})} (\mathbf{x} - \mathbf{x}_k)$$

$$J_k(\mathbf{x}) \geq G(\mathbf{x})$$

Iterative Algorithm for LASSO

Iterative Soft Thresholding Algorithm [DDDM04]:

LASSO Objective Function

$G(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1$

Define:

$$J_k(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1 + (\mathbf{x} - \mathbf{x}_k)^T \underbrace{\left(\alpha\mathbf{I} - \mathbf{A}^T\mathbf{A}\right)}_{\alpha \geq \text{maxeig}(\mathbf{A}^T\mathbf{A})} (\mathbf{x} - \mathbf{x}_k)$$

$$J_k(\mathbf{x}) \geq G(\mathbf{x})$$

Equivalently:

$$J_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_k - \frac{1}{\alpha}\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_k)\|_2^2 + \frac{\lambda}{\alpha}\|\mathbf{x}\|_1 + const.$$

Iterative Algorithm for LASSO

Iterative Soft Thresholding Algorithm [DDDM04]:

LASSO Objective Function

$G(\mathbf{x}) := \|\mathbf{Ax} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1$

Define:

$$J_k(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 + (\mathbf{x} - \mathbf{x}_k)^T \underbrace{\left(\alpha \mathbf{I} - \mathbf{A}^T \mathbf{A}\right)}_{\alpha \geq \text{maxeig}\left(\mathbf{A}^T\mathbf{A}\right)} (\mathbf{x} - \mathbf{x}_k)$$

$$J_k(\mathbf{x}) \geq G(\mathbf{x})$$

Equivalently:

$$J_k(\mathbf{x}) = \|\mathbf{x} \overbrace{- \mathbf{x}_k - \frac{1}{\alpha}\mathbf{A}^T(\mathbf{y} - \mathbf{Ax}_k)}^{-\mathbf{z}}\|_2^2 + \frac{\lambda}{\alpha}\|\mathbf{x}\|_1 + const.$$

Iterative Soft Thresholding Algorithm Cont.

With $\mathbf{z} = \mathbf{x}_k + \frac{1}{\alpha}\mathbf{A}^\mathsf{T}\left(\mathbf{y} - \mathbf{A}\mathbf{x}_k\right)$ we get

$$J_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{z}\|_2^2 + \frac{\lambda}{\alpha}\|\mathbf{x}\|_1 + const.$$

This function is separable:

$$J_k(\mathbf{x}) = \sum_{i=1}^{n}\left[(x_i - z_i)^2 + \frac{\lambda}{\alpha}|x_i|\right] + const.$$

Iterative Soft Thresholding Algorithm Cont.

With $\mathbf{z} = \mathbf{x}_k + \frac{1}{\alpha}\mathbf{A}^\mathsf{T}(\mathbf{y} - \mathbf{A}\mathbf{x}_k)$ we get

$$J_k(\mathbf{x}) = \|\mathbf{x} - \mathbf{z}\|_2^2 + \frac{\lambda}{\alpha}\|\mathbf{x}\|_1 + const.$$

This function is separable:

$$J_k(\mathbf{x}) = \sum_{i=1}^{n}\left[(x_i - z_i)^2 + \frac{\lambda}{\alpha}|x_i|\right] + const.$$

Independently minimize each term

Iterative Soft Thresholding Algorithm Cont.

For each term we have the problem

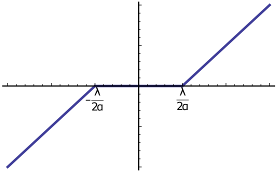
$$\min\ (x_i - z_i)^2 + \frac{\lambda}{\alpha}|x_i|$$

Solution:

$$x_i = \text{SoftThreshold}\left(z_i, \frac{\lambda}{2\alpha}\right)$$

where

$$\text{soft}\left(z_i, \frac{\lambda}{2\alpha}\right) := \begin{cases} z_i + \frac{\lambda}{2\alpha} & \text{if } z_i \leq -\frac{\lambda}{2\alpha} \\ 0 & \text{if } |z_i| < \frac{\lambda}{2\alpha} \\ z_i - \frac{\lambda}{2\alpha} & \text{if } z_i \geq \frac{\lambda}{2\alpha} \end{cases}$$

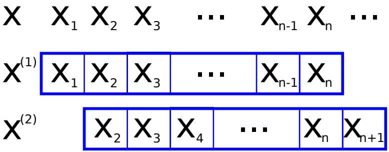


Algorithms for Solving LASSO

Algorithm	Convergence	Method	Reference
ISTA	$\frac{1}{k}$	IST	[DDDM04]
FISTA	$\frac{1}{k^2}$	IST	[BT09]
TwIST	-	IST	[BDF07]
SALSA	-	Augmented Lagrangian	[ABDF11]

Recursive Compressive Sampling

Problem Formulation



Setup:

- Streaming data: $\{x_i\}_{i=1,2,\ldots}$
- Measurements: $\mathbf{y}^{(i)} = \mathbf{A}^{(i)}\mathbf{x}^{(i)}$
- $\mathbf{x}^{(i)} := (x_i \; x_{i+1} \; \ldots \; x_{i+n-1})^\mathsf{T}$, window of size n,

Design:

- **Recursive Sampling:** $\mathbf{y}^{(i+1)} \leftarrow f(\mathbf{y}^{(i)}, \mathbf{x}_{i+n}, \mathbf{x}_i)$
- **Recursive Estimation:** $\hat{\mathbf{x}}^{(i)} \leftarrow g(\hat{\mathbf{x}}^{(i-1)}, \mathbf{y}^{(i)})$

Goal:

- Low complexity filters f, g for online implementation

Recursive Sampling

For obtaining: $\mathbf{y}^{(0)} = \mathbf{A}^{(0)}\mathbf{x}^{(0)}$

- Take $\mathbf{A}^{(i+1)} = \mathbf{A}^{(i)}\mathbf{\Pi}$, where $\mathbf{\Pi} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix}$

$$\mathbf{A}^{(0)} = \begin{bmatrix} | & | & & | & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_{n-1} & \mathbf{a}_n \\ | & | & & | & | \end{bmatrix} \rightarrow \mathbf{A}^{(1)} = \begin{bmatrix} | & | & & | & | \\ \mathbf{a}_2 & \mathbf{a}_3 & \cdots & \mathbf{a}_n & \mathbf{a}_1 \\ | & | & & | & | \end{bmatrix}$$

- Then $\mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} + \mathbf{a}_1^{(i)}(x_{i+n} - x_i)$, rank-1 update

Lemma

If $\mathbf{A}^{(0)}$ satisfies RIP then $\mathbf{A}^{(i)}$ satisfies RIP $\forall i$

Recursive Estimation

Abstraction: $\hat{\mathbf{x}}^{(i)} \leftarrow g(\hat{\mathbf{x}}^{(i-1)}, \mathbf{y}^{(i)})$

Given an iterative solver for LASSO

- Utilization of previous estimate for warm start

$\hat{\mathbf{x}}^{(i)}_{[k]}$:= the estimate of $\mathbf{x}^{(i)}$ at k^{th} iteration

- Starting point:

$$\hat{\mathbf{x}}^{(i+1)}_{[0]} \leftarrow \begin{pmatrix} \hat{x}^{(i)}_2 & \hat{x}^{(i)}_3 & \cdots & \hat{x}^{(i)}_n & 0 \end{pmatrix}$$

Recursive Estimation

Abstraction: $\hat{\mathbf{x}}^{(i)} \leftarrow g(\hat{\mathbf{x}}^{(i-1)}, \mathbf{y}^{(i)})$

Given an iterative solver for LASSO

- Utilization of previous estimate for warm start

$\hat{\mathbf{x}}^{(i)}_{[k]} :=$ the estimate of $\mathbf{x}^{(i)}$ at k^{th} iteration

- Starting point:

$$\hat{\mathbf{x}}^{(i+1)}_{[0]} \leftarrow \begin{pmatrix} \hat{x}^{(i)}_2 & \hat{x}^{(i)}_3 & \cdots & \hat{x}^{(i)}_n & 0 \end{pmatrix}$$

accelerated convergence

Sparse Signals in a Given Orthonormal Basis

It is often the case that the signal itself is not sparse but it is representable sparsely in a particular basis:

$$\begin{aligned} \mathbf{y}^{(i)} &= \mathbf{A}^{(i)}\mathbf{x}^{(i)} \\ &= \mathbf{A}^{(i)}\mathbf{\Phi}\boldsymbol{\alpha}^{(i)} \end{aligned}$$

where $\boldsymbol{\alpha}^{(i)}$ is sparse.
Consider $\mathbf{\Phi}$ is the inverse DFT matrix:

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & \omega & \dots & \omega^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \dots & \omega^{(n-1)^2} \end{bmatrix}$$

Goal: Recursive calculation of $\boldsymbol{\alpha}^{(i)}$ for warm start

$$\text{minimize} \quad \|\mathbf{A}^{(i)}\mathbf{\Phi}\boldsymbol{\alpha}^{(i)} - \mathbf{y}^{(i)}\|_2^2 + \lambda\|\boldsymbol{\alpha}^{(i)}\|_1$$

Sparse Signals in a Given Orthonormal Basis

We follow the Fourier domain operations:

$$\textcolor{blue}{circularshift}: \begin{bmatrix} x_i \\ x_{i+1} \\ \vdots \\ x_{i+n-1} \end{bmatrix} \rightarrow \begin{bmatrix} x_{i+1} \\ \vdots \\ x_{i+n-1} \\ x_i \end{bmatrix} \Leftrightarrow \boldsymbol{\alpha}^{(i)} e^{-\frac{j2\pi}{n}}$$

$$\mathbf{x}^{(i+1)} = \begin{bmatrix} x_{i+1} \\ \vdots \\ x_{i+n-1} \\ x_{i+n} \end{bmatrix} = \begin{bmatrix} x_{i+1} \\ \vdots \\ x_{i+n-1} \\ x_i \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_{i+n} \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ x_i \end{bmatrix} \Leftrightarrow e^{-\frac{j2\pi}{n}} \boldsymbol{\alpha}^{(i)} + (x_{i+n} - x_i)\, \boldsymbol{\psi}_n,$$

where $\boldsymbol{\psi}_n$ is the last column of DFT matrix of size n.
Update rule for $\boldsymbol{\alpha}^{(i+1)}$ becomes

$$\boldsymbol{\alpha}^{(i+1)} = e^{-\frac{j2\pi}{n}} \boldsymbol{\alpha}^{(i)} + (x_{i+n} - x_i)\, \boldsymbol{\psi}_n.$$

Compression Efficiency

Extension: Windowing with variable overlap.
τ: number of slots slid between two estimation periods.

Sampling Efficiency: $\eta_i := \frac{\text{number of samples taken}}{\text{number of entries recovered}}$ up to i^{th} window

Naïve approach: keep m samples for each window

$$\eta := \lim_{i\to\infty} \eta_i = \lim_{i\to\infty} \frac{im}{n+(i-1)\tau}$$
$$= \frac{m}{\tau}.$$

Rank τ update: Sample by $\mathbf{A}^{(i+1)} = \mathbf{A}^{(i)}\mathbf{\Pi}^{\tau}$

$$\eta = \lim_{i\to\infty} \eta_i = \lim_{i\to\infty} \frac{m+(i-1)\tau}{n+(i-1)\tau}$$
$$= 1.$$

Compression Efficiency

Extension: Windowing with variable overlap.

τ: number of slots slid between two estimation periods.

Sampling Efficiency: $\eta_i := \frac{\text{number of samples taken}}{\text{number of entries recovered}}$ up to i^{th} window

Naïve approach: keep m samples for each window

$$\eta = \frac{m}{\tau}.$$

Rank τ update: Sample by $\mathbf{A}^{(i+1)} = \mathbf{A}^{(i)}\mathbf{\Pi}^{\tau}$

$$\eta = 1.$$

$$\eta = \min\left(\frac{m}{\tau}, 1\right)$$

Complexity:

- Sampling: $O(\tau)$
- Estimation:
  - Computations in single iteration $O(n^2)$
  - Expected number of iterations

Signal modal:

$$p_{x_i}(x) = \begin{cases} 0 & \text{w.p.} \quad (1-p) \\ \mathcal{U}([-1,1]) & \text{w.p.} \quad p \end{cases}$$

Expected Number of Iterations

Number of iterations to reach δ optimal point is bounded by [BT09]

$$\left\lceil K \frac{\|\mathbf{x}_{[0]} - \mathbf{x}_*\|}{\sqrt{\delta}} - 1 \right\rceil$$

Noiseless measurements:

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_{[0]}^{(i)} - \mathbf{x}_*^{(i)}\|\right]^2 \leq \tau \mathbb{E}\left[(x_{*j}^{(i)})^2\right] = O(\tau)$$

Noisy measurements:

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_{[0]}^{(i)} - \mathbf{x}_*^{(i)}\|\right]^2 \leq 2\left(C_0 \cdot \mathbb{E}\left[\|\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)}\|_1\right] / \sqrt{k} + \; C_1 \epsilon \right) + \tau \mathbb{E}\left[(x_{*j}^{(i)})^2\right]$$

Number of iterations to reach δ optimal point is bounded by [BT09]

$$\left\lceil K \frac{\|\mathbf{x}_{[0]} - \mathbf{x}_*\|}{\sqrt{\delta}} - 1 \right\rceil$$

Noiseless measurements:

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_{[0]}^{(i)} - \mathbf{x}_*^{(i)}\|\right]^2 \leq \tau \mathbb{E}\left[(x_{*j}^{(i)})^2\right] = O(\tau)$$

Noisy measurements:

$$\mathbb{E}\left[\|\hat{\mathbf{x}}_{[0]}^{(i)} - \mathbf{x}_*^{(i)}\|\right]^2 \leq 2\left(C_0 \cdot \underbrace{\mathbb{E}\left[\|\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)}\|_1\right]/\sqrt{k}}_{\text{Decreasing in n}} + \underbrace{C_1\epsilon}_{O(n)}\right) + \underbrace{\tau\mathbb{E}\left[(x_{*j}^{(i)})^2\right]}_{O(\tau)}$$

Deviation from Expected Signal Sparsity

From signal model it follow:

$$\mathbb{E}\left[\|\mathbf{X} - \mathbf{X}_k\|_1\right] = \sum_{\kappa=k+1}^{n} \binom{n}{\kappa} p^{\kappa}(1-p)^{n-\kappa} \sum_{i=k+1}^{\kappa} \left(1 - \frac{i}{\kappa+1}\right)$$

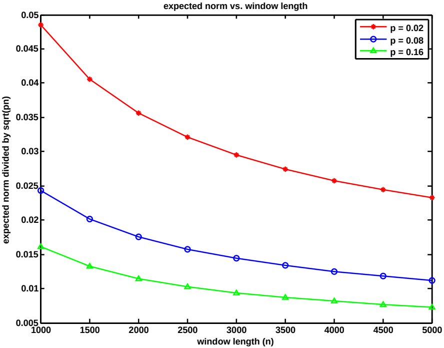


Figure 1: $\mathbb{E}\left[\|\mathbf{x}^{(i)} - \mathbf{x}_k^{(i)}\|_1\right] / \sqrt{k}$ for $k = np$ vs. the window length n

Simulations

Simulation Setting

Measurement model:

- Measurement: $\mathbf{y}^{(i)} = \mathbf{A}^{(i)}\mathbf{x}^{(i)} + \mathbf{w}^{(i)}$
- Signal: Fixed sparsity model (5%), non-zeros $\sim \mathcal{U}([-1, 1])$
- Measurement Matrix: $\mathbf{A}^{(0)}$ randomly select rows from DFT(n) matrix, $m = 5 \times$ window sparsity
- Noise: $\mathbf{w} \sim \mathcal{N}\left(0, \sigma^2\mathbf{I}\right)$

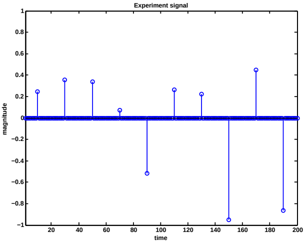


Figure 2: First window of length 200 from the signal used for simulations of warm start and benefit of averaging estimates

Benefit of Warm Start

Simulation:

- Run FISTA on different n and σ^2
- Stop Criterion: normalized update is less than threshold γ

$$\frac{\|\hat{\mathbf{x}}^{(i)}_{[k]} - \hat{\mathbf{x}}^{(i)}_{[k-1]}\|_2}{\|\hat{\mathbf{x}}^{(i)}_{[k-1]}\|_2} < \gamma$$

- Averaged Estimate: $\tilde{x}_i = \frac{1}{n} \sum_{j=0}^{n-1} \hat{x}^{(i-j)}_{j+1}$, for $i \geq n$

Compare: Number of iterations for converge by:

1. using previous estimate for starting point (warm start)
2. using zero vector for starting point (zero start)

Benefit of Warm Start

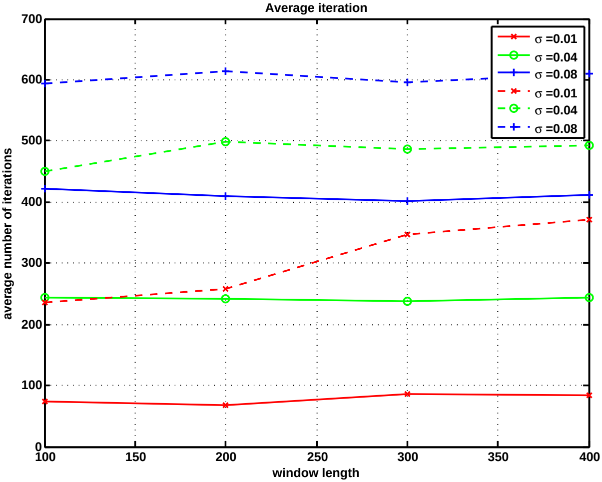


Figure 3: Average number of iterations for convergence. Solid lines depict warm start while dashed lines are used for zero start

Runtime

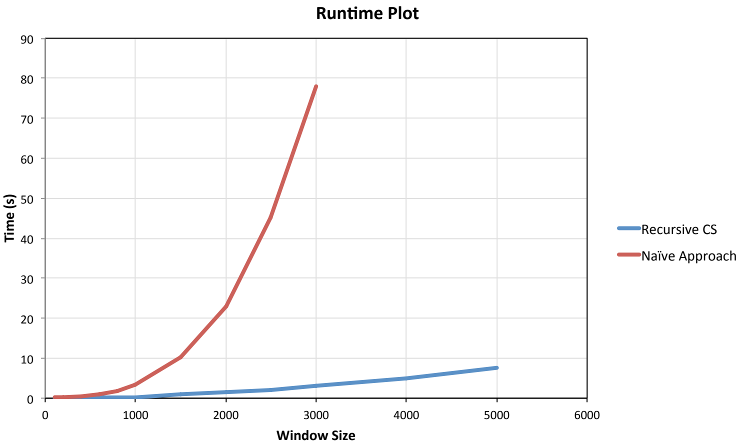


Figure 4: Average time required to solve one window

Performance Simulation

Simulation:

- Averaged Estimate: $\tilde{x}_i = \frac{1}{n}\sum_{j=0}^{n-1} \hat{x}_{j+1}^{(i-j)}$, for $i \geq n$

Define:

1. error of non-overlapping estimates: $e_i = |\hat{x}_1^{(i)} - x_i|^2$
2. error of averaged estimate: $\tilde{e}_i = |\tilde{x}_i - x_i|^2$

Compare:

Average error

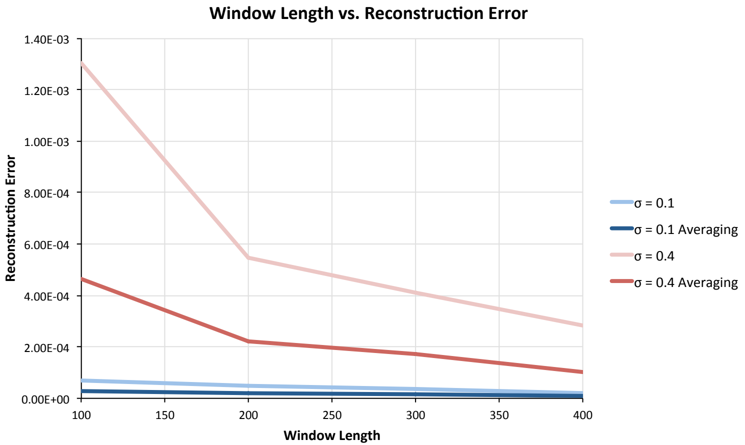


Figure 5: Error plots for averaging and single estimation for different noise variance

Averaging vs Single Estimation

Simulation:

- $\mathbf{y}^{(i)} = \mathbf{Ax} + \mathbf{w}^{(i)}$, $\mathbf{w}^{(i)} \sim \mathcal{N}(0, 0.04^2\mathbf{I})$
- $\mathbf{A} \in \mathcal{R}^{25\times 100}$, random rows from DFT(100).
- Obtain estimate $\hat{\mathbf{x}}^{(i)}$ by LASSO
- Averaged estimate: $\tilde{\mathbf{x}}^{(i)} = \sum_{j=1}^{i} \hat{x}^{(i)}$

Define:

1. Average error: $\frac{1}{i}\sum_{j=1}^{i} \|\mathbf{x}^{(i)} - \mathbf{x}\|_2$
2. Error of averaged estimate: $\|\tilde{\mathbf{x}}^{(i)} - \mathbf{x}\|_2$

Averaging vs Single Estimation

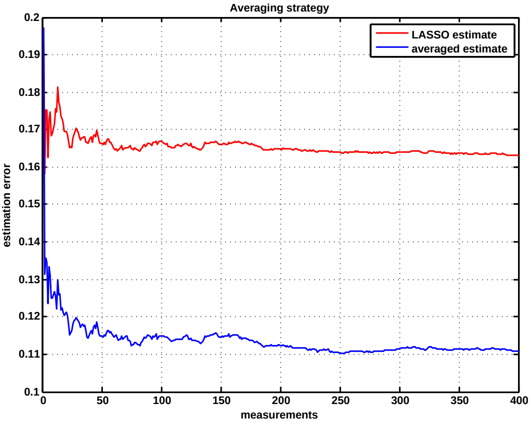


Figure 6: Error plots for single LASSO estimate and averaged estimate.

Voting Strategy

Error does not go to zero $\leftarrow$ LASSO is a biased estimator.

Voting Strategy

Error does not go to zero $\leftarrow$ LASSO is a biased estimator.
But, LASSO is an unbiased estimator for nonzero positions [ZH08]

Voting Strategy

Error does not go to zero $\leftarrow$ LASSO is a biased estimator.

But, LASSO is an unbiased estimator for nonzero positions [ZH08]

Voting:

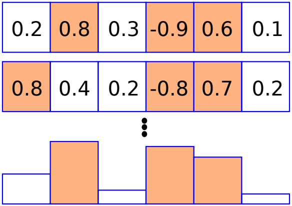


Figure 7: Error plots for single LASSO estimate and averaged estimate.

Voting strategy:

- Solve LASSO to get estimage
- Add votes on indexes of highes absolute magnitude
- Solve least squares on highest cumulative vote indexes
- Average the least squares estimates

Voting Strategy

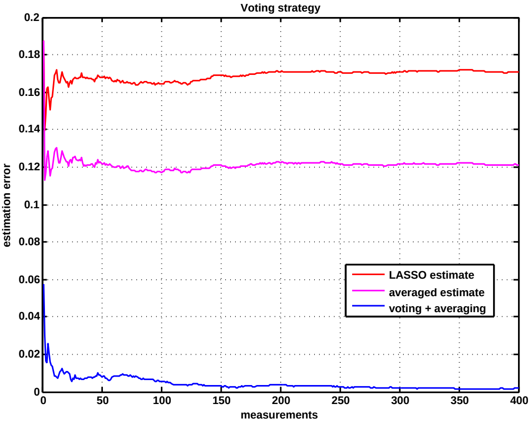


Figure 8: Error plots for single LASSO estimate, averaged estimate and voting + averaging.

Conclusion

- Compressive Sampling on streaming data
- Minimal computational overhead in sampling (rank-1 update)
- Iterative optimization
  - Warm start for leveraging previous estimates for faster convergence
  - Averaged estimates for leveraging window overlap for estimation error reduction

Ingredients for Online Implementation

Speed Up: recursive sampling, recursive estimation
Robustness: averaging estimates

References

[ABDF11] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo.
An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems.
Image Processing, IEEE Transactions on, 20(3):681–695, March 2011.

[BDF07] J. M. Bioucas-Dias and M. A. T. Figueiredo.
A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration.
Image Processing, IEEE Transactions on, 16(12):2992–3004, December 2007.

[BT09] Amir Beck and Marc Teboulle.
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM Journal on Imaging Sciences, 2(1):183–202, 2009.

[CT05] E.J. Candès and T. Tao.
Decoding by linear programming.
IEEE Transactions on Information Theory, 51(12):4203–4215, dec. 2005.

[CW08] E.J. Candès and M.B. Wakin.
An introduction to compressive sampling.
IEEE Signal Processing Magazine, 25(2):21–30, march 2008.

[DDDM04] I. Daubechies, M. Defrise, and C. De Mol.
An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.
Comm. Pure Appl. Math., 57:1413–1457, 2004.

[ZH08] Cun-Hui Zhang and Jian Huang.
The sparsity and bias of the Lasso selection in high-dimensional linear regression.
arXiv.org, August 2008.

Iterative Soft Thresholding Algorithm

Iterative Soft Thresholding Algorithm:

ISTA

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^{m}$, λ, $\epsilon \geq 0$

```
x_0 ← 0, t ← 1, α ← maxeig(AᵀA), T ← λ/(2α)
for t = 1, 2, 3, ... do
    x_t ← soft(x_{t-1} + (Aᵀ(y − A x_{t-1}))/α, T)
    Check stopping criterion and terminate if it holds:
        |G(x_t) − G(x_{t-1})| ≤ ε
    t ← t + 1
end for
```

$$\mathbf{x}_0 \leftarrow 0,\ t \leftarrow 1,\ \alpha \leftarrow \text{maxeig}\left(\mathbf{A}^{\mathsf{T}}\mathbf{A}\right),\ T \leftarrow \frac{\lambda}{2\alpha}$$

$$\mathbf{x}_t \leftarrow \text{soft}\left(\mathbf{x}_{t-1} + \left(\mathbf{A}^{\mathsf{T}}\left(\mathbf{y} - \mathbf{A}\mathbf{x}_{t-1}\right)\right)/\alpha,\ T\right)$$

$$\left|G(\mathbf{x}_t) - G(\mathbf{x}_{t-1})\right| \leq \epsilon$$