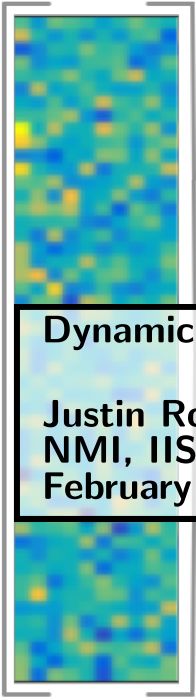
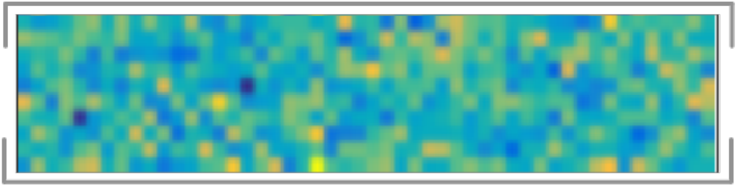




Dynamic ℓ_1 Reconstruction

Justin Romberg, Georgia Tech ECE
NMI, IISc, Bangalore, India
February 22, 2015

Goal: a dynamical framework for sparse recovery

Given $\boldsymbol{y}$ and $\boldsymbol{\Phi}$, solve

$$\min_{x} \; \lambda \|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi x} - \boldsymbol{y}\|_2^2$$

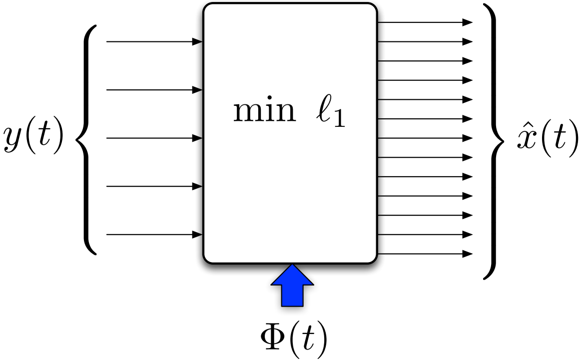
Goal: a dynamical framework for sparse recovery

We want to move from:

Given $\boldsymbol{y}$ and $\boldsymbol{\Phi}$, solve

$$\min_x \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi x} - \boldsymbol{y}\|_2^2$$

to



Agenda

We will look at dynamical reconstruction in two different contexts:

- Fast updating of solutions of ℓ_1 optimization programs

M. Salman Asif

- Systems of nonlinear differential equations that solve ℓ_1 (and related) optimization programs, implemented as continuous-time neural nets

Aurèle Balavoine

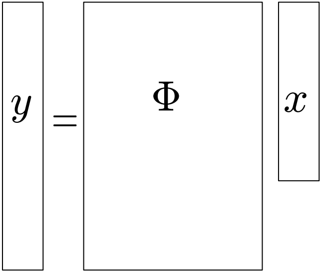
Chris Rozell

Classical: Recursive least-squares

- System model:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$$

- $\boldsymbol{\Phi}$ has full column rank
- $\boldsymbol{x}$ is arbitrary



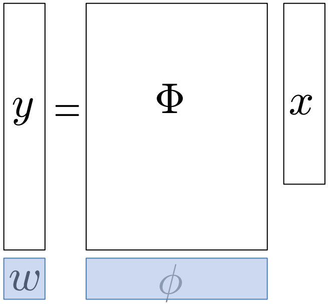
- Least-squares estimate:

$$\min \; \|\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}\|_2^2 \quad \Longrightarrow \quad \hat{\boldsymbol{x}} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{y}$$

Classical: Recursive least-squares

- Sequential measurement:

$$\begin{bmatrix} \boldsymbol{y} \\ w \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\phi}^{\mathrm{T}} \end{bmatrix} \boldsymbol{x}$$



- Compute new estimate using rank-1 update:

$$\begin{aligned} \hat{\boldsymbol{x}}_1 &= (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} + \boldsymbol{\phi}\boldsymbol{\phi}^{T})^{-1}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{\phi} \cdot w) \\ &= \hat{\boldsymbol{x}}_0 + K_1(w - \boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{x}_0) \end{aligned}$$

where

$$K_1 = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\phi}(1 + \boldsymbol{\phi}^{\mathrm{T}}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\phi})^{-1}$$

- With the previous inverse in hand, the update has the cost of a *few matrix-vector multiplies*

Classical: The Kalman filter

- Linear dynamical system for state evolution and measurement:

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t \boldsymbol{x}_t + \boldsymbol{e}_t$$
$$\boldsymbol{x}_{t+1} = \boldsymbol{F}_t \boldsymbol{x}_t + \boldsymbol{d}_t$$

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \boldsymbol{\Phi}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ -\boldsymbol{F}_1 & \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \boldsymbol{\Phi}_2 & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & -\boldsymbol{F}_2 & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi}_3 & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_0 \boldsymbol{x}_0 \\ \boldsymbol{y}_1 \\ \mathbf{0} \\ \boldsymbol{y}_2 \\ \mathbf{0} \\ \boldsymbol{y}_3 \\ \vdots \end{bmatrix}$$

- As time marches on, we add both rows and columns.
- Least-squares problem:

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots} \sum_t \left(\sigma_t \|\boldsymbol{\Phi}_t \boldsymbol{x}_t - \boldsymbol{y}_t\|_2^2 + \lambda_t \|\boldsymbol{x}_t - \boldsymbol{F}_{t-1} \boldsymbol{x}_{t-1}\|_2^2 \right)$$

Classical: The Kalman filter

- Linear dynamical system for state evolution and measurement:

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t x_t + \boldsymbol{e}_t$$
$$\boldsymbol{x}_{t+1} = \boldsymbol{F}_t \boldsymbol{x}_t + \boldsymbol{d}_t$$

- Least-squares problem:

$$\min_{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots} \sum_t \left(\sigma_t \|\boldsymbol{\Phi}_t \boldsymbol{x}_t - \boldsymbol{y}_t\|_2^2 + \lambda_t \|\boldsymbol{x}_t - \boldsymbol{F}_{t-1}\boldsymbol{x}_{t-1}\|_2^2 \right)$$

- Again, we can use low-rank updating to solve this recursively:

$$\begin{aligned}
\boldsymbol{v}_k &= \boldsymbol{F}_k \hat{\boldsymbol{x}} \\
\boldsymbol{K}_{k+1} &= (\boldsymbol{F}_k \boldsymbol{P}_k \boldsymbol{F}_k^{\mathrm{T}} + \mathbf{I})\boldsymbol{\Phi}_{k+1}^{\mathrm{T}}(\boldsymbol{\Phi}_{k+1}(\boldsymbol{F}_k \boldsymbol{P}_k \boldsymbol{F}_k^{\mathrm{T}} + \mathbf{I})\boldsymbol{\Phi}_{k+1}^{\mathrm{T}} + \mathbf{I})^{-1} \\
\hat{\boldsymbol{x}}_{k+1|k+1} &= \boldsymbol{v}_k + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \boldsymbol{\Phi}_{k+1}\boldsymbol{v}_k) \\
\boldsymbol{P}_{k+1} &= (\mathbf{I} - \boldsymbol{K}_{k+1}\boldsymbol{\Phi}_{k+1})(\boldsymbol{F}_k \boldsymbol{P}_k \boldsymbol{F}_k^{\mathrm{T}} + \mathbf{I})
\end{aligned}$$

Optimality conditions for BPDN

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{W}\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

- Conditions for $\boldsymbol{x}^*$ (supported on Γ^*) to be a solution:

$$\begin{aligned} \boldsymbol{\phi}_\gamma^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{x}^* - \boldsymbol{y}) &= -W[\gamma,\gamma]z[\gamma] \qquad && \gamma \in \Gamma^* \\ |\boldsymbol{\phi}_\gamma^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{x}^* - \boldsymbol{y})| &\le W[\gamma,\gamma] && \gamma \in \Gamma^{*c} \end{aligned}$$

  where $z[\gamma] = \mathrm{sign}(x[\gamma])$

- Derived simply by computing the subgradient of the functional above

Example: time-varying sparse signal

- Initial measurements. Observe

$$\boldsymbol{y}_1 = \boldsymbol{\Phi}\boldsymbol{x}_1 + \boldsymbol{e}_1$$

- Initial reconstruction. Solve

$$\min_{\boldsymbol{x}} \; \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}_1\|_2^2$$

Example: time-varying sparse signal

- Initial measurements. Observe

$$\boldsymbol{y}_1 = \boldsymbol{\Phi}\boldsymbol{x}_1 + \boldsymbol{e}_1$$

- Initial reconstruction. Solve

$$\min_{\boldsymbol{x}} \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}_1\|_2^2$$

- A new set of measurements arrives:

$$\boldsymbol{y}_2 = \boldsymbol{\Phi}\boldsymbol{x}_2 + \boldsymbol{e}_2$$

- Reconstruct again using ℓ_1-min:

$$\min_{\boldsymbol{x}} \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}_2\|_2^2$$

Example: time-varying sparse signal

- Initial measurements. Observe

$$\boldsymbol{y}_1 = \boldsymbol{\Phi}\boldsymbol{x}_1 + \boldsymbol{e}_1$$

- Initial reconstruction. Solve

$$\min_{\boldsymbol{x}} \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}_1\|_2^2$$

- A new set of measurements arrives:

$$\boldsymbol{y}_2 = \boldsymbol{\Phi}\boldsymbol{x}_2 + \boldsymbol{e}_2$$

- Reconstruct again using ℓ_1-min:

$$\min_{\boldsymbol{x}} \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}_2\|_2^2$$

- We can gradually move from the first solution to the second solution using *homotopy*

$$\min \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - (1-\epsilon)\boldsymbol{y}_1 - \epsilon\boldsymbol{y}_2\|_2^2$$

Take ϵ from $0 \to 1$

Example: time-varying sparse signal

$$\min\ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi x} - (1-\epsilon)\boldsymbol{y}_{\text{old}} - \epsilon\boldsymbol{y}_{\text{new}}\|_2^2, \quad \text{take } \epsilon \text{ from } 0 \to 1$$

- Path from old solution to new solution is *piecewise linear*
- Optimality conditions for fixed ϵ:

$$\boldsymbol{\Phi}_\Gamma^{\mathrm{T}}(\boldsymbol{\Phi x} - (1-\epsilon)\boldsymbol{y}_{\text{old}} - \epsilon\boldsymbol{y}_{\text{new}}) = -\lambda\,\text{sign}\,\boldsymbol{x}_\Gamma$$

$$\|\boldsymbol{\Phi}_{\Gamma^c}^{\mathrm{T}}(\boldsymbol{\Phi x} - (1-\epsilon)\boldsymbol{y}_{\text{old}} - \epsilon\boldsymbol{y}_{\text{new}})\|_\infty < \lambda$$

Γ = active support

- Update direction:

$$\partial\boldsymbol{x} = \begin{cases} -(\boldsymbol{\Phi}_\Gamma^{\mathrm{T}}\boldsymbol{\Phi}_\Gamma)^{-1}(\boldsymbol{y}_{\text{old}} - \boldsymbol{y}_{\text{new}}) & \text{on } \Gamma \\ \mathbf{0} & \text{off } \Gamma \end{cases}$$

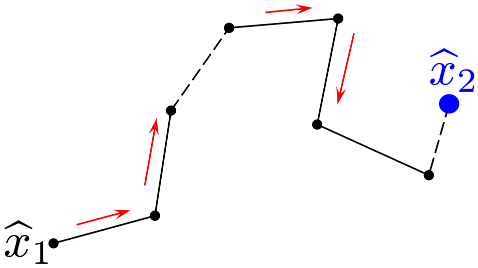
Path from old solution to new

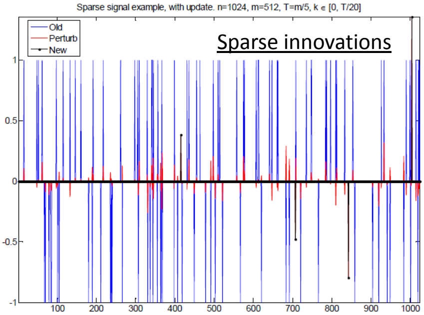
Γ = support of current solution.
Move in this direction

$$\partial \boldsymbol{x} = \begin{cases} -(\boldsymbol{\Phi}_\Gamma^{\mathrm{T}} \boldsymbol{\Phi}_\Gamma)^{-1}(\boldsymbol{y}_{\mathrm{old}} - \boldsymbol{y}_{\mathrm{new}}) & \text{on } \Gamma \\ \boldsymbol{0} & \text{off } \Gamma \end{cases}$$

until support changes, or one of these constraints is violated:

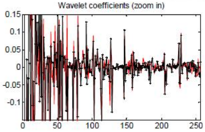
$$\left|\boldsymbol{\phi}_\gamma^{\mathrm{T}}(\boldsymbol{\Phi}(\boldsymbol{x} + \epsilon \partial \boldsymbol{x}) - (1-\epsilon)\boldsymbol{y}_{\mathrm{old}} - \epsilon \boldsymbol{y}_{\mathrm{new}})\right| < \lambda \quad \text{for all } \gamma \in \Gamma^c$$

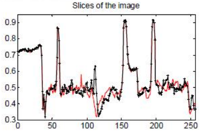




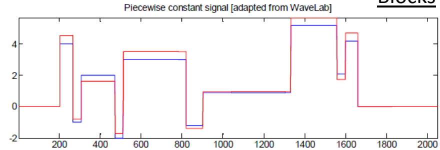
Sparse innovations

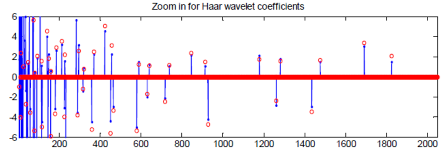
House



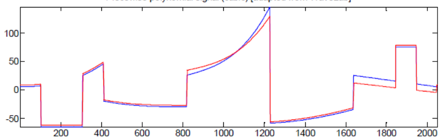


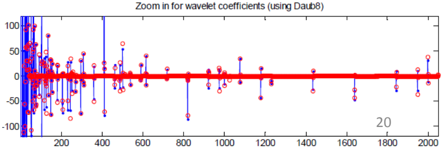
Blocks





Pcw. poly





Numerical experiments: time-varying sparse signals

Signal type	DynamicX* (nProdAtA, CPU)	LASSO homotopy (nProdAtA, CPU)	GPSR-BB (nProdAtA, CPU)	FPC_AS (nProdAtA, CPU)
N = 1024 M = 512 T = m/5, k ~ T/20 Values = +/- 1	(23.72, 0.132)	(235, 0.924)	(104.5, 0.18)	(148.65, 0.177)
Blocks	(2.7, 0.028)	(76.8, 0.490)	(17, 0.133)	(53.5, 0.196)
Pcw. Poly.	(13.83, 0.151)	(150.2, 1.096)	(26.05, 0.212)	(66.89, 0.25)
House slices	(26.2, 0.011)	(53.4, 0.019)	(92.24, 0.012)	(90.9, 0.036)

$$\tau = 0.01\|A^T y\|_\infty$$

nProdAtA: roughly the avg. no. of matrix vector products with A and A^T
CPU: average cputime to solve

[Asif and R. 2009]

Other updates

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{W}\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

$\boldsymbol{W}$ = weights (diagonal, positive)

Using similar ideas, we can *dynamically update* the solution when

- the underlying signal changes slightly,
- we add/remove measurements,
- the weights changes,

But none of these are really "predict and update" ...

A general, flexible homotopy framework

We want to solve

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{W}\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

- Initial guess/prediction: $\boldsymbol{v}$
- Solve

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{W}\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + (1-\epsilon)\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}$$

  for $\epsilon : 0 \to 1$.
- Taking

$$\boldsymbol{u} = -\boldsymbol{W}\boldsymbol{z} - \boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{v} - \boldsymbol{y})$$

  for some $\boldsymbol{z} \in \partial(\|\boldsymbol{v}\|_1)$ makes $\boldsymbol{v}$ optimal for $\epsilon = 0$

Moving from the warm-start to the solution

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{W}\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + (1-\epsilon)\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}$$

The optimality conditions are

$$\begin{aligned}\boldsymbol{\Phi}_\Gamma^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}) + (1-\epsilon)\boldsymbol{u} &= -W \operatorname{sign} \boldsymbol{x}_\Gamma \\ \left|\boldsymbol{\phi}_\gamma^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}) + (1-\epsilon)\boldsymbol{u}\right| &\leq W[\gamma,\gamma]\end{aligned}$$

We move in direction

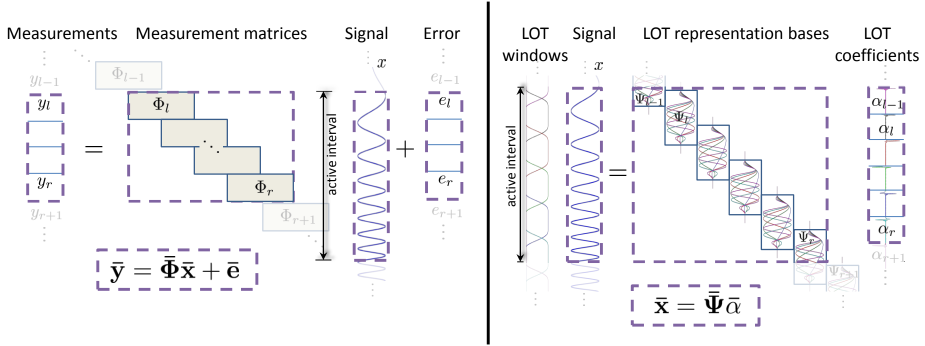
$$\partial \boldsymbol{x} = \begin{cases} \boldsymbol{u}_\Gamma & \text{on } \Gamma \\ \boldsymbol{0} & \text{on } \Gamma^c \end{cases}$$

until a component shrinks to zero or a constraint is violated, yielding new Γ

Streaming sparse recovery

Observations: $\boldsymbol{y}_t = \boldsymbol{\Phi}_t \boldsymbol{x}_t + \boldsymbol{e}_t$

Representation: $x[n] = \sum_{p,k} \alpha_{p,k} \psi_{p,k}[n]$

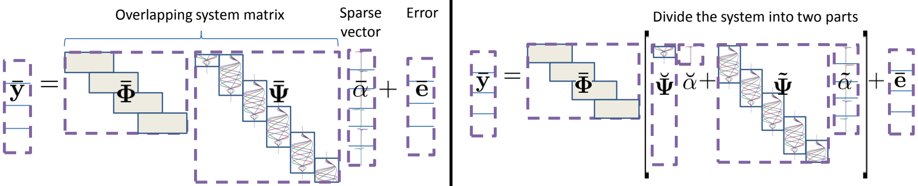


Streaming sparse recovery

Iteratively reconstruct the signal over a sliding (active) interval, form $\boldsymbol{u}$ from your prediction, then take $\epsilon : 0 \to 1$ in

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{W}\boldsymbol{\alpha}\|_1 + \frac{1}{2}\|\bar{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Psi}}\boldsymbol{\alpha} - \tilde{\boldsymbol{y}}\|_2^2 + (1-\epsilon)\boldsymbol{u}^{\mathrm{T}}\boldsymbol{\alpha}$$

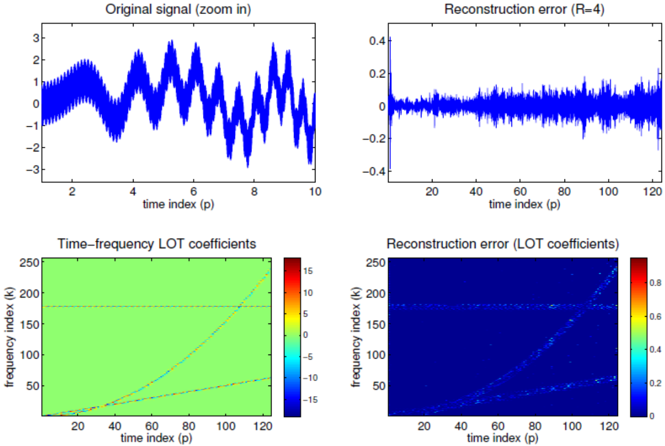
where $\tilde{\boldsymbol{\Psi}}, \tilde{\boldsymbol{y}}$ account for edge effects



$$\bar{\mathbf{y}} = \bar{\mathbf{\Phi}}\bar{\mathbf{\Psi}}\bar{\alpha} + \bar{\mathbf{e}}$$

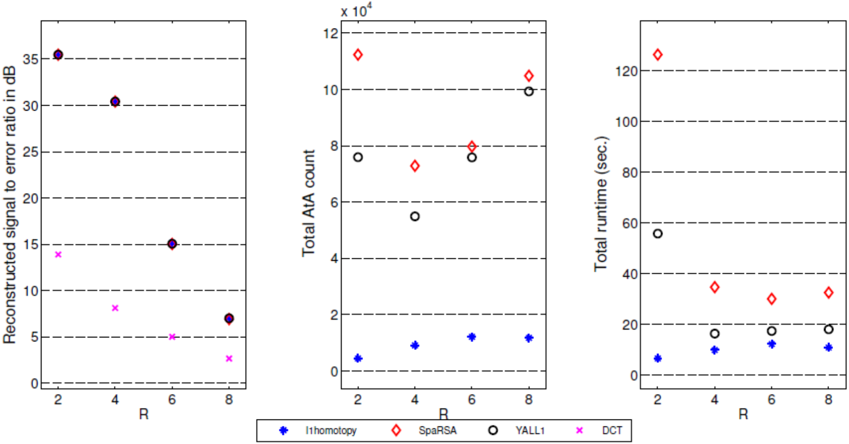
$$\bar{\mathbf{y}} = \bar{\mathbf{\Phi}}[\breve{\mathbf{\Psi}}\breve{\alpha} + \tilde{\mathbf{\Psi}}\tilde{\alpha}] + \bar{\mathbf{e}} \Rightarrow \tilde{\mathbf{y}} = \bar{\mathbf{y}} - \bar{\mathbf{\Phi}}\breve{\mathbf{\Psi}}\breve{\alpha}$$

Streaming signal recovery: Simulation



(Top-left) Mishmash signal (zoomed in for first 2560 samples.
(Top-right) Error in the reconstruction at R=N/M = 4.
(Bottom-left) LOT coefficients. (Bottom-right) Error in LOT coefficients

Streaming signal recovery: Simulation



(left) SER at different R from ±1 random measurements in 35 db noise
(middle) Count for matrix-vector multiplications
(right) Matlab execution time

Streaming signal recovery: Dynamic signal

Observation/evolution model:

$$\boldsymbol{y}_t = \boldsymbol{\Phi}_t \boldsymbol{x}_t + \boldsymbol{e}_t$$
$$\boldsymbol{x}_{t+1} = \boldsymbol{F}_t \boldsymbol{x}_t + \boldsymbol{d}_t$$

We solve

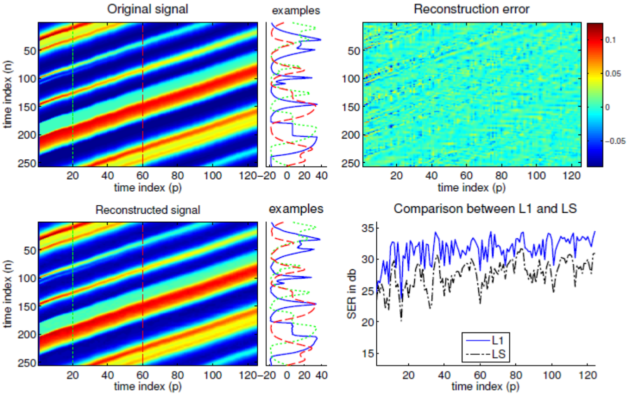
$$\min_{\boldsymbol{\alpha}} \sum_t \|\boldsymbol{W}_t \boldsymbol{\alpha}_t\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}_t \boldsymbol{\Psi}_t \boldsymbol{\alpha}_t - \boldsymbol{y}_t\|_2^2 + \frac{1}{2}\|\boldsymbol{F}_{t-1}\boldsymbol{\Psi}_{t-1}\boldsymbol{\alpha}_{t-1} - \boldsymbol{\Psi}_t \boldsymbol{\alpha}_t\|_2^2$$

(formulation similar to Vaswani 08, Carmi et al 09, Angelosante et al 09, Zainel at al 10, Charles et al 11)

using

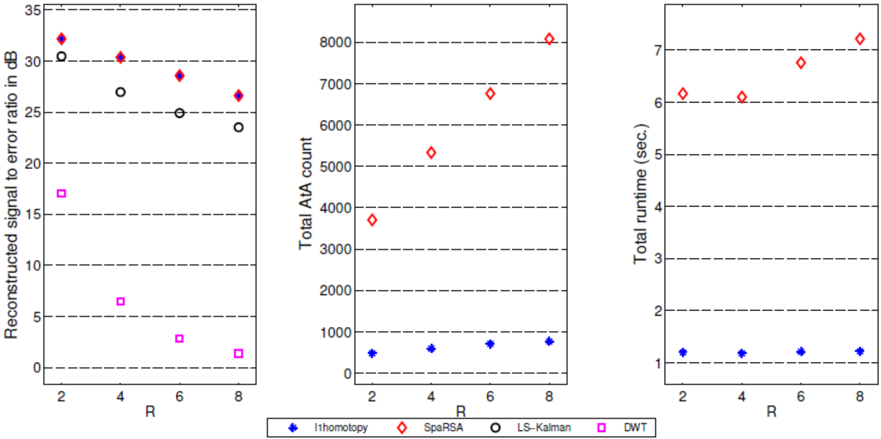
$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{W}\boldsymbol{\alpha}\|_1 + \frac{1}{2}\|\bar{\boldsymbol{\Phi}}\tilde{\boldsymbol{\Psi}}\boldsymbol{\alpha} - \tilde{\boldsymbol{y}}\|_2^2 + \frac{1}{2}\|\bar{\boldsymbol{F}}\tilde{\boldsymbol{\Psi}}\boldsymbol{\alpha} - \tilde{\boldsymbol{q}}\|_2^2 + (1-\epsilon)\boldsymbol{u}^{\mathrm{T}}\boldsymbol{\alpha}$$

Dynamic signal: Simulation



(Top-left) Piece-Regular signal (shifted copies) in image
(Top-right) Error in the reconstruction at R=N/M = 4.
(Bottom-left) Reconstructed signal at R=4.
(Bottom-right) Comparison of SER for the L1-regularized and the L2-regularized problems

Dynamic signal: Simulation

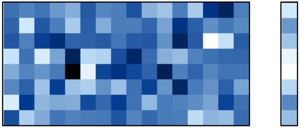


(left) SER at different R from ±1 random measurements in 35 db noise
(middle) Count for matrix-vector multiplications
(right) Matlab execution time

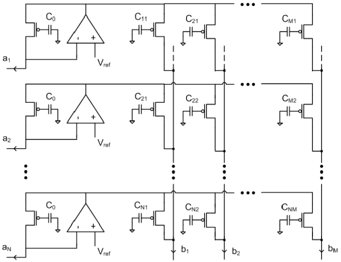
Dynamical systems for sparse recovery

Analog vector-matrix-multiply

- **Digital Multiply-and-Accumulate**



- **Analog Vector-Matrix Multiplier**



- Small time constant
- Low power consumption

- Limited accuracy
- Limited dynamic range

Dynamical systems for sparse recovery

There are simple systems of nonlinear differential equations that settle to the solution of

$$\min_{\boldsymbol{x}} \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

or more generally

$$\min_{\boldsymbol{x}} \ \lambda \sum_{n=1}^{N} C(x[n]) + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

The Locally Competitive Algorithm (LCA):

$$\begin{aligned} \tau\dot{\boldsymbol{u}}(t) &= -\boldsymbol{u}(t) - (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} - \mathbf{I})\boldsymbol{x}(t) + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y} \\ \boldsymbol{x}(t) &= T_\lambda(\boldsymbol{u}(t)) \end{aligned}$$

is a neurologically-inspired (Rozell et al 08) system which settles to the solutions of the above

Locally competitive algorithm

Cost function

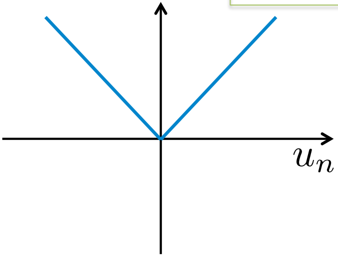
$$V(\boldsymbol{x}) = \lambda \sum_n C(x_n) + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

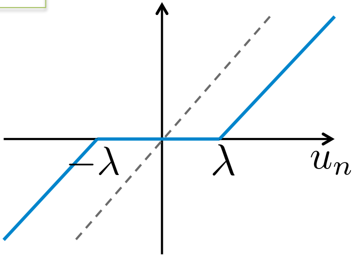
$$\tau \dot{\boldsymbol{u}}(t) = -\boldsymbol{u}(t) - (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} - \mathbf{I})\boldsymbol{x}(t) + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{y}$$
$$x_n(t) = T_\lambda(u_n(t))$$

$C(u_n)$

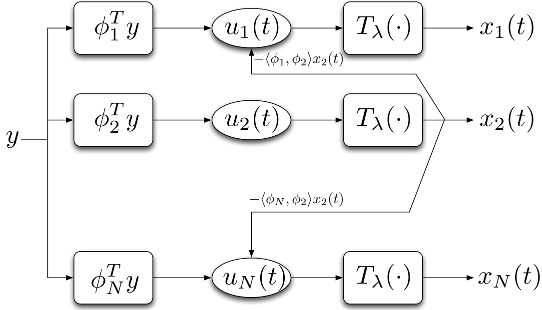
$$\lambda \frac{dC}{dx}(u) = u - x$$

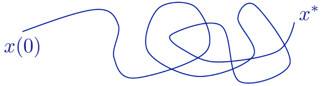
$x_n = T_\lambda(u_n)$





Key questions

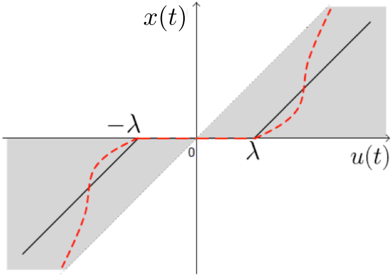




$$\min_x \lambda \sum_n C(x_n) + \frac{1}{2}\|\Phi x - y\|_2^2$$

- Uniform convergence (general)
- Convergence properties/speed (general)
- Convergence speed for sparse recovery via ℓ_1 minimization

LCA convergence



Assumptions

1. $\boldsymbol{u} - \boldsymbol{x} \in \lambda \partial C(\boldsymbol{x})$
2. $\boldsymbol{x} = T_\lambda(\boldsymbol{u}) = \begin{cases} 0 & |u| \leq \lambda \\ f(u) & |u| > \lambda \end{cases}$
3. $T_\lambda(\cdot)$ is odd and continuous, $f'(u) > 0$, $f(u) < u$
4. $f(\cdot)$ is subanalytic
5. $f'(u) \leq \alpha$

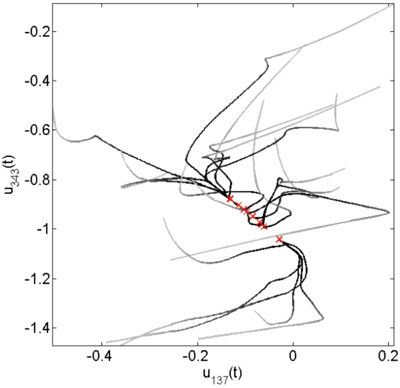
LCA convergence

Global asymptotic convergence:

If 1–5 hold above, then the LCA is globally asymptotically convergent:

$$\boldsymbol{x}(t) \to \boldsymbol{x}^*, \quad \boldsymbol{u}(t) \to \boldsymbol{u}^*, \quad \text{as } t \to \infty$$

where $\boldsymbol{x}^*$ is a critical point of the functional.

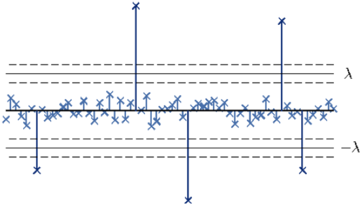


Convergence: support is recovered in finite time

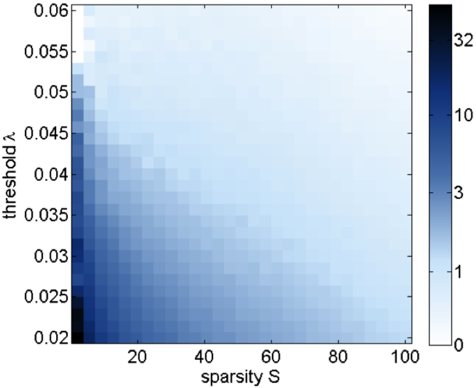
If the LCA converges to a fixed point $\boldsymbol{u}^*$ such that

$$|u_\gamma| \geq \lambda + r, \quad \text{and} \quad |u_\gamma| \leq \lambda - r$$

for all $\gamma \in \Gamma^{*c}$, then the *support* of $\boldsymbol{x}^*$ is recovered in *finite time*



\# of switches/sparsity



$$\boldsymbol{\Phi} = [DCT \quad I]$$
$$M = 256, \; N = 512$$

Convergence: exponential (of a sort)

Suppose we have

- the conditions for global convergence (with $f'(u) \leq \alpha$)
- energy preservation for every point we visit:

$$(1-\delta)\|\tilde{\boldsymbol{x}}(t)\|_2^2 \ \leq \ \|\boldsymbol{\Phi}\tilde{\boldsymbol{x}}(t)\|_2^2 \ \leq \ (1+\delta)\|\tilde{\boldsymbol{x}}(t)\|_2^2 \quad \forall t,$$

where $\tilde{\boldsymbol{x}}(t) = \boldsymbol{x}(t) - \boldsymbol{x}^*$, and $\alpha d < 1$

then the LCA converges exponentially to a unique fixed point:

$$\|\boldsymbol{u}(t) - \boldsymbol{u}^*\|_2 \ \leq \ \kappa_0 \, e^{-(1-\alpha\delta)t/\tau}$$

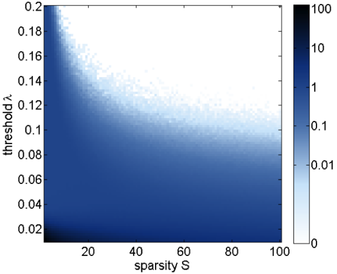
Efficient activation for ℓ_1

If $\mathbf{\Phi}$ a "random compressed sensing matrix" and

$$M \;\geq\; \text{Const} \cdot S \log(N/S)$$

then for reasonably small values of λ and starting from rest

$$|\Gamma(t)| \leq 2|\Gamma^*|$$



Similar results for OMP/ROMP, CoSAMP, etc. in CS literature

Iterative Soft Thresholding with a Dynamic Input

Iterative soft thresholding (ISTA)

Solve

$$\min_{\boldsymbol{x}} \ \lambda\|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

using the simple iteration:

$$\boldsymbol{x}(\ell+1) = \mathrm{T}_\lambda\left[\boldsymbol{x}(\ell) + \eta\left(\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}(\ell))\right)\right],$$

where $\mathrm{T}_\lambda =$ soft thresholding, $\eta =$ stepsize.

- One of the earliest sparse recovery algorithms (Daubechies et al '04)
- Basis for many competitive first-order methods for min-ℓ_1 (GPSR, SPARSA, Twist, etc.)

ISTA convergence

Observe

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}_* + \boldsymbol{\epsilon}, \quad \|\boldsymbol{\epsilon}\|_2 \leq \sigma, \quad \boldsymbol{x}_* \text{ is } k\text{-sparse.}$$

If $\boldsymbol{\Phi}$ satisfies the k-RIP and $\lambda\sqrt{k} \geq c_1\|\boldsymbol{x}_*\| + c_2\sigma$, then

$$\|\boldsymbol{x}(\ell) - \boldsymbol{x}^*\|_2 \leq C_0\kappa^\ell + E_*, \quad E_* \leq C_1\left(\lambda\sqrt{k} + \sigma\right)$$

(Bredies et al '08, Zhang '09)

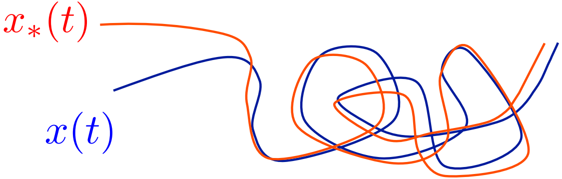
ISTA with a dynamic input

We observe

$$\boldsymbol{y}(t) = \boldsymbol{\Phi}\boldsymbol{x}_*(t) + \boldsymbol{\epsilon}(t)$$

$\boldsymbol{x}_*$ is k sparse, $\|\boldsymbol{\epsilon}(t)\|_2 \leq \sigma$, $\|\dot{\boldsymbol{x}}_*(t)\|_2 \leq \mu$

The input changes at each step, so we are constantly "chasing" the solution



ISTA tracking results

Static: (Bredies et al '08, Zhang '09)

$$\|\boldsymbol{x}(\ell) - \boldsymbol{x}^*\|_2 \leq C_0 \kappa^\ell + E_*,$$

where $E_* \lesssim \lambda\sqrt{k} + \sigma$

Dynamic: (Balavoine, R, Rozell, '14)

$$\|\boldsymbol{x}(t) - \boldsymbol{x}^*\|_2 \leq \tilde{C}_0 \tilde{\kappa}^t + E_* + \tilde{C}_1\, \mu$$

where $\|\dot{\boldsymbol{x}}_*(t)\|_2 \leq \mu$.

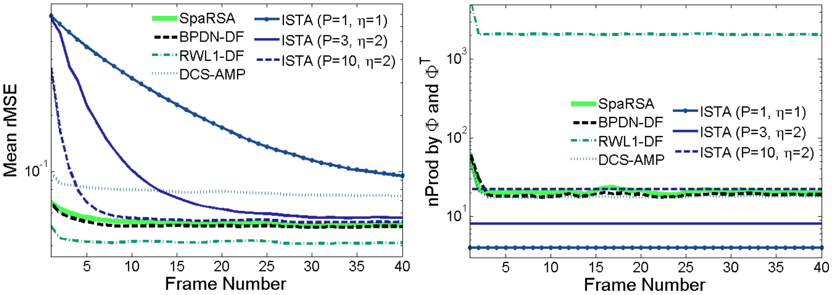
Dynamic ISTA: Simulations

videos

Signal:
$X(t)$ is 128×128
$X(t) = \Psi a(t)$
Ψ is a wavelet basis.

Measurements:
$\mathrm{y(t)} = \Phi X(t) + \epsilon(t)$
Φ is a subsampled noiselet transform.

Dynamic ISTA: Simulations



DCS-AMP: Ziniel et al '10

RWL1-DF/BPDN-DF: Charles et al '13

Other techniques: Sejdinovic et al '10, Angelosante et al '10, Vaswani '08

References

- M. Asif and J. Romberg, "Dynamic updating for l1 minimization," IEEE Journal on Special Topics in Signal Processing, April 2010.
- M. Asif and J. Romberg, "Fast and accurate algorithms for re-weighted ℓ_1-norm minimization," IEEE Transactions on Signal Processing, 2013.
- M. Asif and J. Romberg, "Sparse recovery of streaming signals using ℓ_1 homotopy," IEEE Transactions on Signal Processing, 2014.
- A. Balavoine, J. Romberg, and C. Rozell, "Convergence and Rate Analysis of Neural Networks for Sparse Approximation," IEEE Transactions on Neural Networks and Learning Systems, September 2012.
- A. Balavoine, J. Romberg, and C. Rozell, "Convergence Speed of a Dynamical System for Sparse Recovery," to appear in IEEE Transactions on Signal Processing, 2013.
- A. Balavoine, C. Rozell, and J. Romberg, "Iterative and continuous soft-thresholding with a dynamic input," submitted to IEEE Transactions on Signal Processing, May 2014.

`http://users.ece.gatech.edu/~justin/Publications.html`