Dynamic ℓ_1 Reconstruction

Justin Romberg
Georgia Tech, ECE

Collaborators:
M. Salman Asif
Aurèle Balavoine
Chris Rozell

ROKS Workshop
July 9, 2013
Leuven, Belgium

Underdetermined systems of equations

- Unknown N-point signal x_0
- Small number of measurements

$$y_k = \langle x_0, \phi_k \rangle, \quad k = 1, \ldots, M \qquad \text{or} \qquad y = \Phi x_0$$

- Fewer measurements than degrees of freedom, $M \ll N$

$$\begin{bmatrix} \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \qquad \boldsymbol{\Phi} \qquad \end{bmatrix} \begin{bmatrix} \boldsymbol{x_0} \\ \\ \\ \\ \end{bmatrix}$$

- Treat acquisition as a *linear inverse problem*
- Compressive Sampling: for *sparse* x_0, we can "invert" *incoherent* Φ

Sparse recovery

- Given M linear measurements of an S-sparse signal

$$y \;=\; \Phi x_0 \;+\; \text{noise}$$

when can we recover x_0 ?

Sparse recovery

- Given M linear measurements of an S-sparse signal

$$y \;=\; \Phi x_0 \;+\; \text{noise}$$

when can we recover x_0 ?

- Key condition: matrix Φ is a *restricted isometry*:

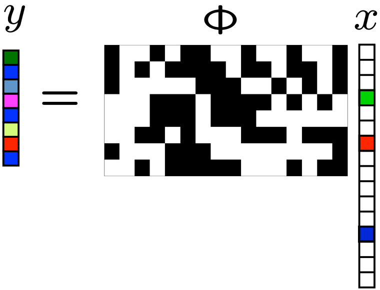
$$(1-\delta)\|x\|_2^2 \;\leq\; \|\Phi x\|_2^2 \;\leq\; (1+\delta)\|x\|_2^2$$

for all $2S$-sparse x

[Candes and Tao '06]

Random matrices

Example: $\Phi_{i,j} \sim \pm 1$ w/ prob $1/2$, iid



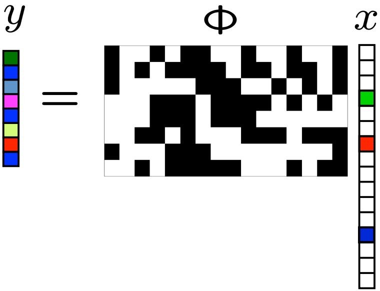
Can recover S-sparse x_0 from

$$M \gtrsim S \cdot \log(N/S)$$

measurements using convex programming, greedy algorithms, ...

Random matrices

Example: $\Phi_{i,j} \sim \pm 1$ w/ prob $1/2$, iid



Can recover S-sparse x_0 from

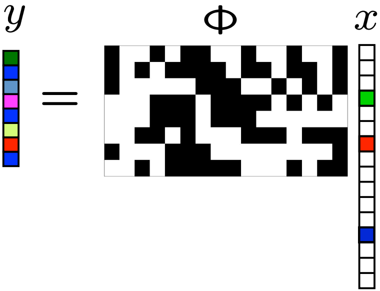
$$M \gtrsim S \cdot \log(N/S)$$

measurements using ℓ_1 minimization:

$$\min \ \|x\|_1 \quad \text{subject to} \quad \Phi x = y$$

Random matrices

Example: $\Phi_{i,j} \sim \pm 1$ w/ prob $1/2$, iid



Can recover S-sparse (in basis Ψ) $x_0 = \Psi\alpha_0$ from

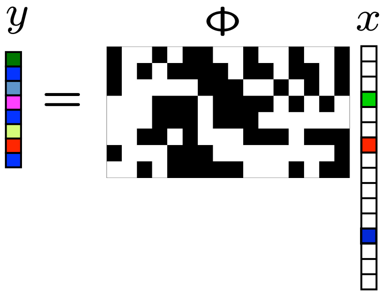
$$M \ \gtrsim \ S \cdot \log(N/S)$$

measurements using ℓ_1 minimization:

$$\min \ \|\alpha\|_1 \quad \text{subject to} \quad \Phi\Psi\alpha = y$$

Random matrices

Example: $\Phi_{i,j} \sim \pm 1$ w/ prob $1/2$, iid



Can stably recover $\approx S$-sparse (in basis Ψ) $x_0 = \Psi\alpha_0$ from

$$M \gtrsim S \cdot \log(N/S)$$

noisy measurements using ℓ_1 minimization:

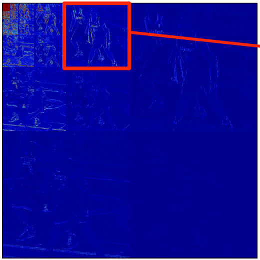
$$\min \ \lambda\|\alpha\|_1 + \frac{1}{2}\|\Phi\Psi\alpha - y\|_2^2$$

for appropriate λ.

Sparsity

Decompose signal/image $x(t)$ in orthobasis $\{\psi_i(t)\}_i$

$$x(t) = \sum_i \alpha_i \psi_i(t)$$



wavelet transform

zoom

x_0

$\{\alpha_i\}_i$

Wavelet approximation

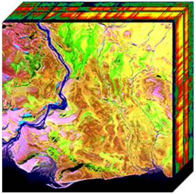
Take 1% of *largest* coefficients, set the rest to zero (adaptive)

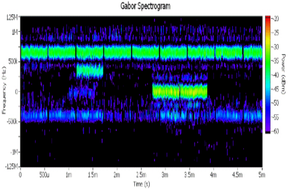
original

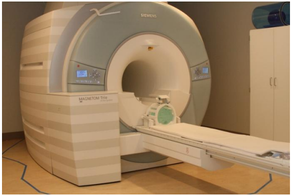
approximated

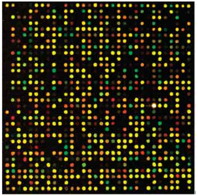
rel. error $= 0.031$

Integrating compression and sensing









Goal: a dynamical framework for sparse recovery

Given y and Φ, solve

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

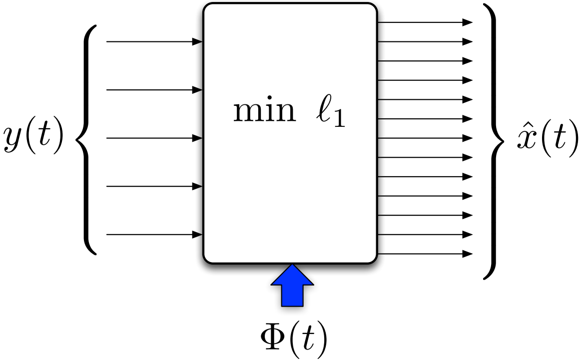
Goal: a dynamical framework for sparse recovery

We want to move from:

Given y and Φ, solve

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

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

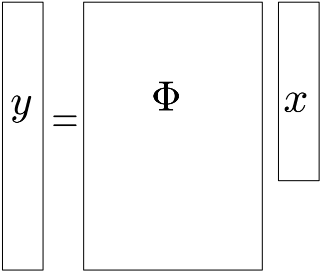
Chris Rozell

Classical: Recursive least-squares

- System model:

$$y = \Phi x$$

- Φ has full column rank
- x is arbitrary



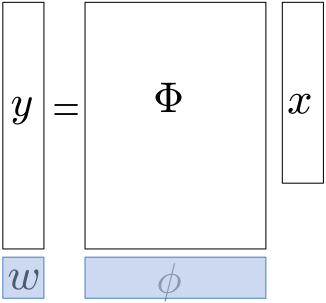
- Least-squares estimate:

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

Classical: Recursive least-squares

- Sequential measurement:

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



- Compute new estimate using rank-1 update:

$$\begin{aligned} \hat{x}_1 &= (\Phi^{\mathrm{T}}\Phi + \phi\phi^T)^{-1}(\Phi^T y + \phi \cdot w) \\ &= \hat{x}_0 + K_1(w - \phi^T x_0) \end{aligned}$$

  where

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

- 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:

$$
\begin{aligned}
y_t &= \Phi_t x_t + e_t \\
x_{t+1} &= F_t x_t + f_t
\end{aligned}
$$

$$
\begin{bmatrix}
I & 0 & 0 & 0 & \cdots \\
\Phi_1 & 0 & 0 & 0 & \cdots \\
-F_1 & I & 0 & 0 & \cdots \\
0 & \Phi_2 & 0 & 0 & \cdots \\
0 & -F_2 & I & 0 & \cdots \\
0 & 0 & \Phi_3 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ \vdots
\end{bmatrix}
=
\begin{bmatrix}
F_0 x_0 \\ y_1 \\ 0 \\ y_2 \\ 0 \\ y_3 \\ \vdots
\end{bmatrix}
$$

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

$$
\min_{x_1, x_2, \ldots} \sum_t \left(\sigma_t \|\Phi_t y_t - y_t\|_2^2 + \lambda_t \|x_t - F_{t-1} x_{t-1}\|_2^2 \right)
$$

Classical: The Kalman filter

- Linear dynamical system for state evolution and measurement:

$$\begin{aligned} y_t &= \Phi_t x_t + e_t \\ x_{t+1} &= F_t x_t + d_t \end{aligned}$$

- Least-squares problem:

$$\min_{x_1, x_2, \ldots} \sum_t \left(\sigma_t \|\Phi_t y_t - y_t\|_2^2 + \lambda_t \|x_t - F_{t-1} x_{t-1}\|_2^2 \right)$$

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

$$\begin{aligned} v_k &= F_k \hat{x} \\ K_{k+1} &= (F_k P_k F_k^T + I)\Phi_{k+1}^T (\Phi_{k+1}(F_k P_k F_k^T + I)\Phi_{k+1}^T + I)^{-1} \\ \hat{x}_{k+1|k+1} &= v_k + K_{k+1}(y_{k+1} - \Phi_{k+1} v_k) \\ P_{k+1} &= (I - K_{k+1}\Phi_{k+1})(F_k P_k F_k^T + I) \end{aligned}$$

Dynamic sparse recovery: ℓ_1 filtering

- Goal: efficient updating for optimization programs like

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

- We want to *dynamically update* the solution when
  - the underlying signal changes slightly,
  - we add/remove measurements,
  - the weights changes,
  - we have streaming measurements for an evolving signal (adding/removing columns from Φ)

Optimality conditions for BPDN

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

- Conditions for x^* (supported on Γ^*) to be a solution:

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

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

- Derived simply by computing the subgradient of the functional above

Example: time-varying sparse signal

- Initial measurements. Observe

$$y_1 = \Phi x_1 + e_1$$

- Initial reconstruction. Solve

$$\min_x \ \lambda\|x\|_1 + \frac{1}{2}\|\Phi x - y_1\|_2^2$$

Example: time-varying sparse signal

- Initial measurements. Observe

$$y_1 = \Phi x_1 + e_1$$

- Initial reconstruction. Solve

$$\min_x \ \lambda \|x\|_1 + \frac{1}{2}\|\Phi x - y_1\|_2^2$$

- A new set of measurements arrives:

$$y_2 = \Phi x_2 + e_2$$

- Reconstruct again using ℓ_1-min:

$$\min_x \ \lambda \|x\|_1 + \frac{1}{2}\|\Phi x - y_2\|_2^2$$

Example: time-varying sparse signal

- Initial measurements. Observe

$$y_1 = \Phi x_1 + e_1$$

- Initial reconstruction. Solve

$$\min_x \ \lambda \|x\|_1 + \frac{1}{2}\|\Phi x - y_1\|_2^2$$

- A new set of measurements arrives:

$$y_2 = \Phi x_2 + e_2$$

- Reconstruct again using ℓ_1-min:

$$\min_x \ \lambda \|x\|_1 + \frac{1}{2}\|\Phi x - y_2\|_2^2$$

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

$$\min \ \lambda \|x\|_1 + \frac{1}{2}\|\Phi x - (1-\epsilon) y_1 - \epsilon y_2\|_2^2$$

Take ϵ from $0 \to 1$

Example: time-varying sparse signal

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

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

$$\begin{aligned} \Phi_\Gamma^{\mathrm{T}}(\Phi x - (1-\epsilon)y_{\text{old}} - \epsilon y_{\text{new}}) &= -\lambda \operatorname{sign} x_\Gamma \\ \|\Phi_{\Gamma^c}^{\mathrm{T}}(\Phi x - (1-\epsilon)y_{\text{old}} - \epsilon y_{\text{new}})\|_\infty &< \lambda \end{aligned}$$

  Γ = active support

- Update direction:

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

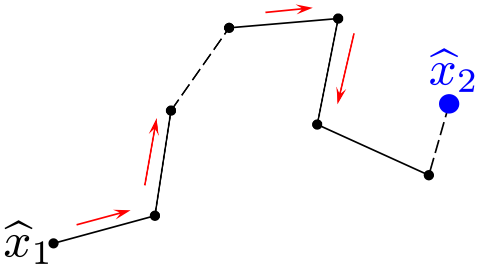
Path from old solution to new

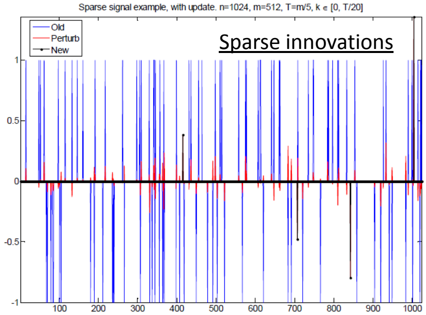
Γ = support of current solution.
Move in this direction

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

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

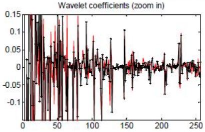
$$\left|\phi_\gamma^T(\Phi(x+\epsilon\partial x) - (1-\epsilon)y_{\mathrm{old}} - \epsilon y_{\mathrm{new}})\right| < \lambda \quad \text{for all } \gamma \in \Gamma^c$$

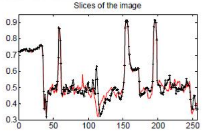




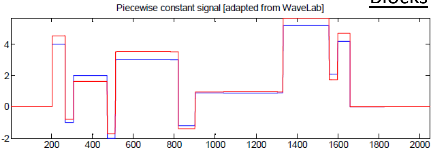
Sparse innovations

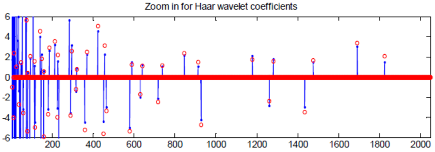
House



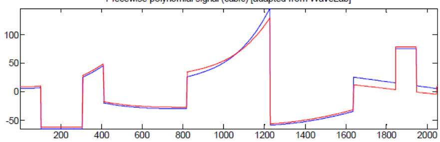


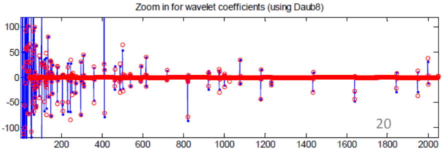
Blocks





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]

Adding a measurement

- Analog of recursive least squares for ℓ_1 min:

$$\begin{bmatrix} y \\ w \end{bmatrix} = \begin{bmatrix} \Phi \\ \phi \end{bmatrix} x + \begin{bmatrix} e \\ d \end{bmatrix} \quad \longrightarrow \quad \min_x \ \lambda\|x\|_1 + \frac{1}{2}\|\Phi x - y\|_2^2 + \frac{1}{2}|\langle \phi, x\rangle - w|^2$$

- Work in the new measurement slowly

$$\min \ \lambda\|x\|_1 + \frac{1}{2}\left(\|\Phi x - y\|_2^2 + \epsilon|\langle \phi, x\rangle - w|^2\right)$$

Again, the solution path is piecewise linear in ϵ

[Garrigues et al. 08, Asif and R 09]

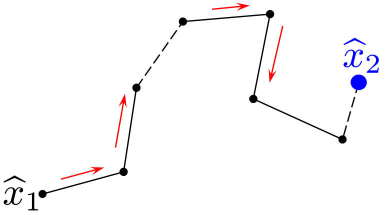
Adding a measurement: updating

- Optimality conditions

$$\Phi_\Gamma^{\mathrm{T}}(\Phi x - y) + \epsilon(\langle \phi, x\rangle - w)\phi_\Gamma = -\lambda \operatorname{sign} x_\Gamma$$
$$\|\Phi_{\Gamma^c}^T(\Phi x - y) + \epsilon(\langle \phi, x\rangle - w)\phi_{\Gamma^c}\|_\infty < \lambda$$

- From critical point x_{ϵ_0}, update direction is

$$\partial x = \begin{cases} (w - \langle \phi, x_{\epsilon_0}\rangle)\cdot(\Phi_\Gamma^{\mathrm{T}}\Phi_\Gamma + \epsilon_0\phi\phi^{\mathrm{T}})^{-1}\phi_\Gamma & \text{on } \Gamma \\ 0 & \text{off } \Gamma \end{cases}$$



Numerical experiments: adding a measurement

$N = 1024$, measurements $M = 512$, sparsity $S = 100$

Add P new measurements

Compare the average number of *matrix-vector* products per update

P	λ ($\tau = \lambda \|\mathbf{A}^T\mathbf{y}\|_\infty$)	DynamicSeq	LASSO	GPSR-BB	FPC_AS
1	0.5	2.3	41.86	11.86	15.98
	0.1	4.72	159.76	42.64	50.70
	0.05	4.5	162.34	38.80	97.73
	0.01	8.02	233.70	55.46	79.83
5	0.5	5.88	42.00	14.24	15.96
	0.1	9.58	152.54	46.42	47.48
	0.05	10.70	161.36	47.96	98.75
	0.01	20.32	227.82	66.64	78.58
10	0.5	7.6	44.72	14.96	16.12
	0.1	14.98	155.26	53.12	47.05
	0.05	16.40	162.72	52.12	98.51
	0.01	29.34	241.52	75.44	82.91

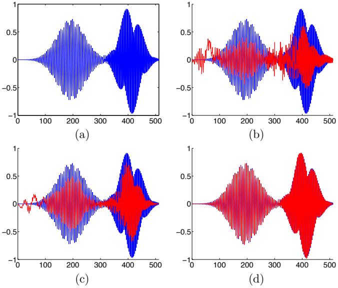
Reweighted ℓ_1

Weighted ℓ_1 reconstruction:

$$\min_x \sum_k w_k |x_k| + \frac{1}{2}\|\Phi x - y\|_2^2 \quad = \quad \min_x \|Wx\|_1 + \frac{1}{2}\|\Phi x - y\|_2^2$$

solve this iteratively, adapting the weights to the previous solution:

$$w_k = \frac{\lambda}{|x_k^{\mathrm{old}}| + c}$$



(a) (b) (c) (d)

(from Boyd, Candes, Wakin '08)

Changing the weights

Iterative reweighting: take $\{w_k\} \to \{\tilde{w}_k\}$

Optimality conditions:

$$\begin{aligned} \phi_k^*(y - \Phi x) &= (\epsilon w_k + (1-\epsilon)\tilde{w}_k) z_k && \text{on support, } k \in \Gamma \\ |\phi_k^*(y - \Phi x)| &< \epsilon w_k + (1-\epsilon)\tilde{w}_k && \text{off support, } k \in \Gamma^c \end{aligned}$$

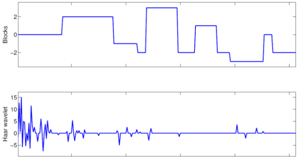
Update direction (increasing ϵ):

$$\partial x = \begin{cases} (\Phi_\Gamma^* \Phi_\Gamma)^{-1}(W - \tilde{W})z & \text{on } \Gamma \\ 0 & \text{on } \Gamma^c \end{cases}$$

[Asif and R 2012]

Numerical experiments: changing the weights

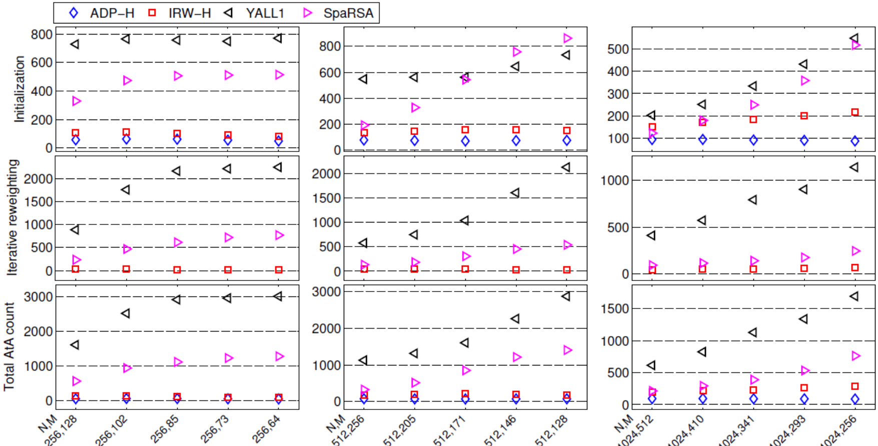
Numerical Experiments



Sparse signal of length N recovered from M Gaussian measurements

ADP-H (adaptive weighting via homotopy)
IRW-H (iterative reweighting via homotopy)

SpaRSA [Wright et al 2007]
YALL1 [Yang et al 2007]



A general, flexible homotopy framework

- Formulations above
  - depend critically on maintaining optimality
  - are very efficient when the solutions are close
- Streaming measurements for evolving signals require some type of *predict and update* framework

$$\text{Kalman filter:}\quad \begin{aligned} v_k &= F\hat{x}_k, && \text{(predict)} \\ \hat{x}_{k+1} &= v_k + K(y - \Phi_k v_k), && \text{(update)} \end{aligned}$$

- What program does the prediction v_k solve?
- Can we trace the path to the solution from a general "warm start"?

A general, flexible homotopy framework

We want to solve

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

- Initial guess/prediction: v
- Solve

  $$\min_x \ \|Wx\|_1 + \frac{1}{2}\|\Phi x - y\|_2^2 + (1-\epsilon)u^T x$$

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

  $$u = -Wz - \Phi^T(\Phi v - y)$$

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

Moving from the warm-start to the solution

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

The optimality conditions are

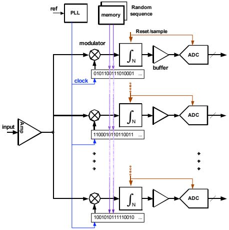
$$\begin{aligned} \Phi_\Gamma^T(\Phi x - y) + (1-\epsilon)u &= -W \operatorname{sign} x_\Gamma \\ \left|\phi_\gamma^T(\Phi x - y) + (1-\epsilon)u\right| &\leq W[\gamma,\gamma] \end{aligned}$$

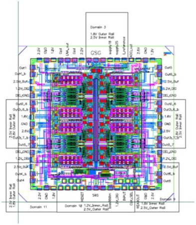
We move in direction

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

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

Streaming measurements: random modulation receiver

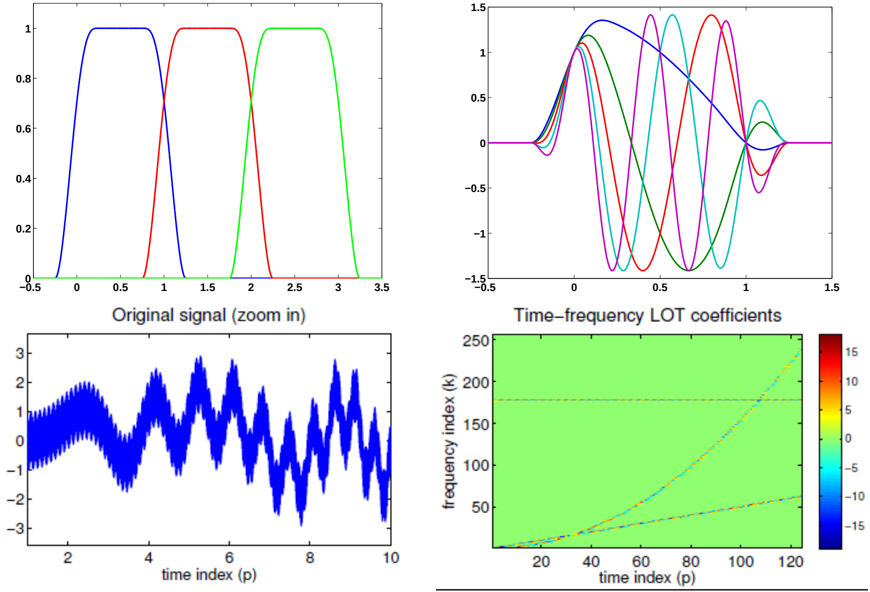




(arch. of Yoo and Emami; see also Mishali et al, Murray et al)

- Built as part of DARPA's A2I program
- Multiple (8) channels, operating with different mixing sequences
- Effective BW/chan = 2.5 GHz
  Sample rate/chan = 50 MHz
- Applications: radar pulse detection, communications surveillance, geolocation

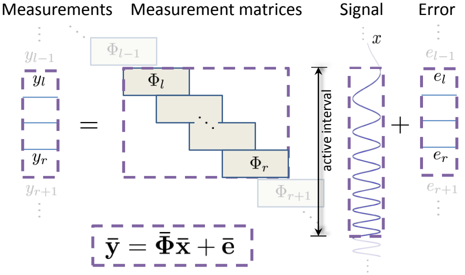
Streaming basis: Lapped orthogonal transform

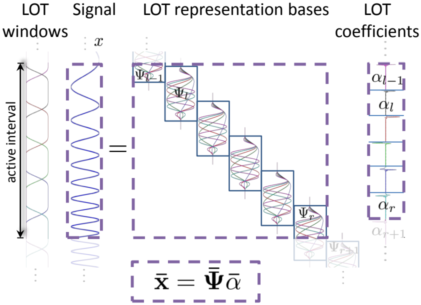


Streaming sparse recovery

Observations: $$y_t = \Phi_t x_t + e_t$$

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



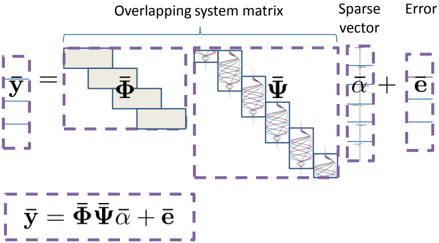


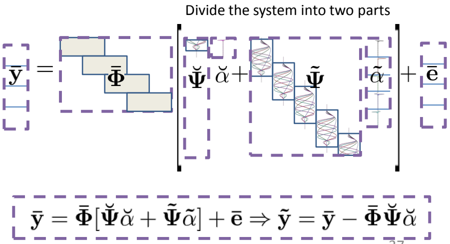
Streaming sparse recovery

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

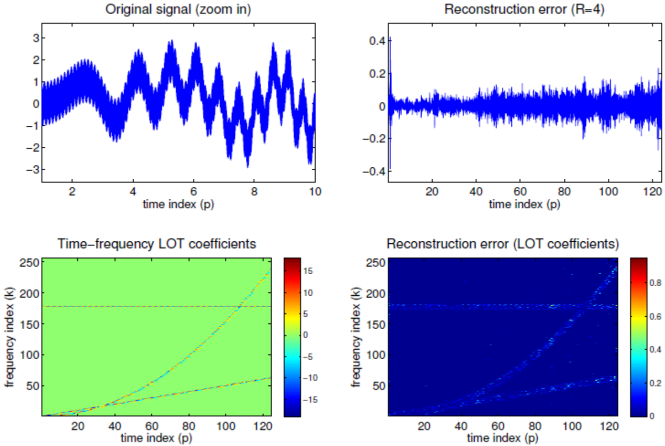
$$\min_{\alpha} \|W\alpha\|_1 + \frac{1}{2}\|\bar{\Phi}\tilde{\Psi}\alpha - \tilde{y}\|_2^2 + (1-\epsilon)u^T\alpha$$

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



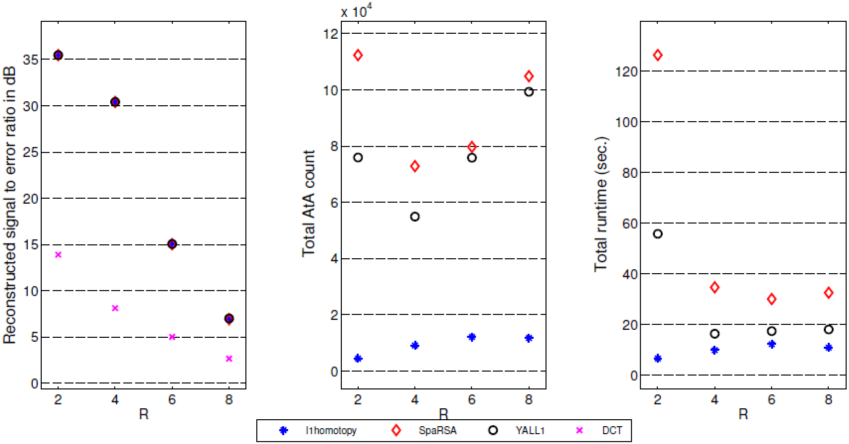


Streaming signal recovery: Simulation



(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:

$$\begin{aligned} y_t &= \Phi_t x_t + e_t \\ x_{t+1} &= F_t x_t + d_t \end{aligned}$$

We solve

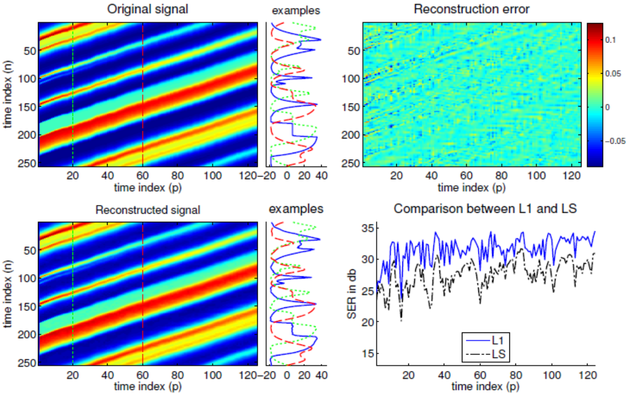
$$\min_{\alpha} \sum_t \|W_t \alpha_t\|_1 + \frac{1}{2}\|\Phi_t \Psi_t \alpha_t - y_t\|_2^2 + \frac{1}{2}\|F_{t-1}\Psi_{t-1}\alpha_{t-1} - \Psi_t \alpha_t\|_2^2$$

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

using

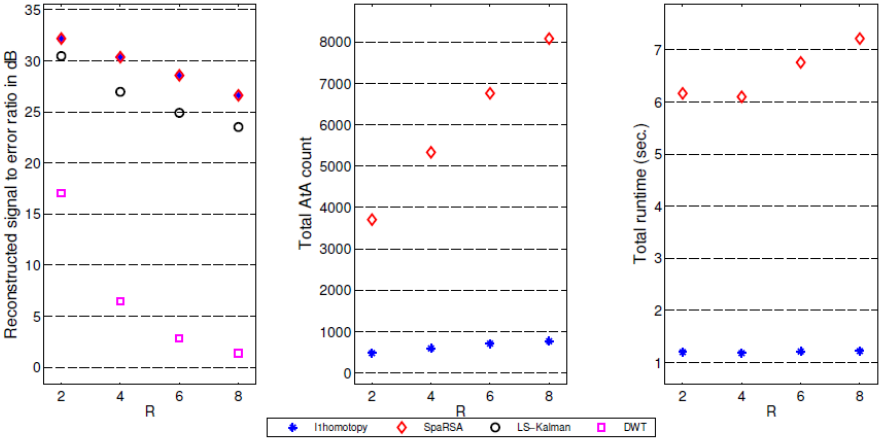
$$\min_{\alpha} \|W\alpha\|_1 + \frac{1}{2}\|\bar{\Phi}\tilde{\Psi}\alpha - \tilde{y}\|_2^2 + \frac{1}{2}\|\bar{F}\tilde{\Psi}\alpha - \tilde{q}\|_2^2 + (1-\epsilon)u^T\alpha$$

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

Approximate analog computing

- Radical re-think of how computer arithmetic is done — computations use the physics of the devices (transistors) more directly
- Use $< 1\%$ of the transistors, maybe $1/10,000$ of the power, possibly 100x faster than GPU
- Computations are *noisy*, overall precision $\approx 10^{-2}$

Approximate analog computing

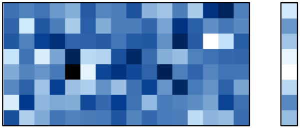
- Radical re-think of how computer arithmetic is done — computations use the physics of the devices (transistors) more directly
- Use $< 1\%$ of the transistors, maybe $1/10,000$ of the power, possibly 100x faster than GPU
- Computations are *noisy*, overall precision $\approx 10^{-2}$
- Small scale successes (embedded beamforming, adaptive filtering)

Approximate analog computing

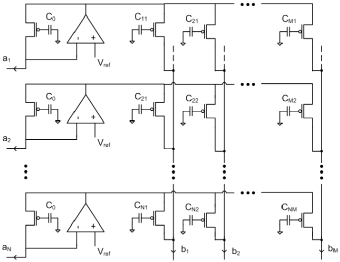
- Radical re-think of how computer arithmetic is done — computations use the physics of the devices (transistors) more directly
- Use $< 1\%$ of the transistors, maybe $1/10,000$ of the power, possibly 100x faster than GPU
- Computations are *noisy*, overall precision $\approx 10^{-2}$
- Small scale successes (embedded beamforming, adaptive filtering)
- Medium to large scale potential
  - FPAAs
  - specialized circuits for optimization
    (Hopfield networks, neural net implementations)
  - general (SIMD) computing architecture ?
- Much of this work is proprietary, start-ups swallowed up by AD, NI, ...
  Lyric semiconductor, GTronix, Singular Computing, ...

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

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

or more generally

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

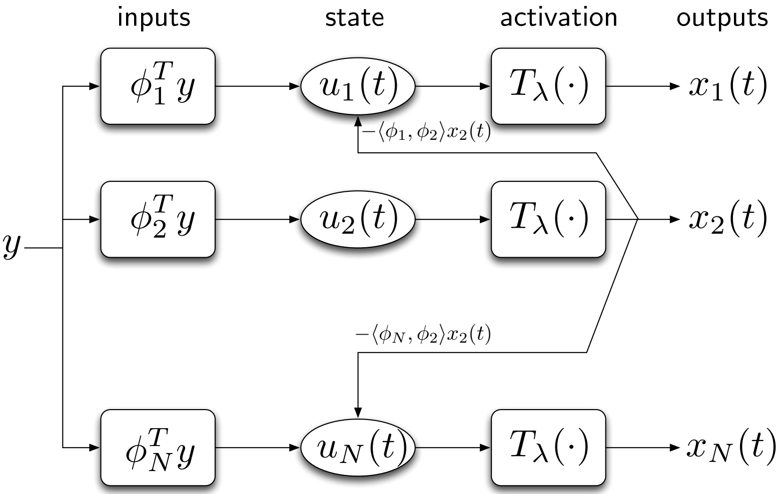
The Locally Competitive Algorithm (LCA):

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

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

Locally competitive algorithm

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

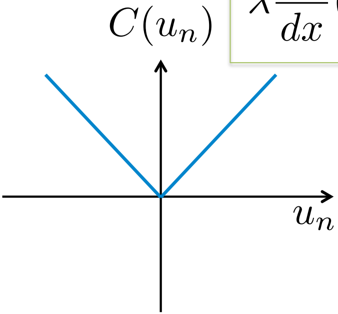


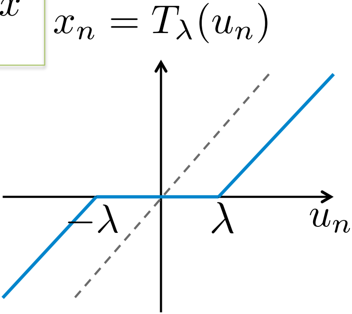
Locally competitive algorithm

Cost function

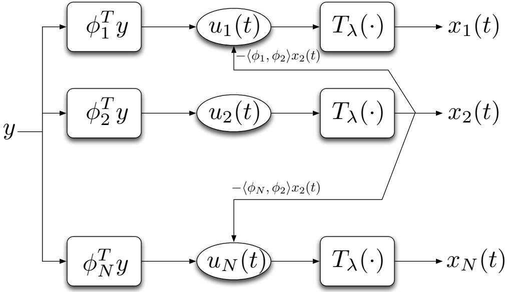
$$V(x) = \lambda \sum_n C(x_n) + \frac{1}{2}\|\Phi x - y\|_2^2 \qquad \begin{aligned} \tau \dot{u}(t) &= -u(t) - (\Phi^T\Phi - I)x(t) + \Phi^T y \\ x_n(t) &= T_\lambda(u_n(t)) \end{aligned}$$

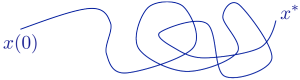
$$\boxed{\lambda \frac{dC}{dx}(u) = u - x}$$





Key questions

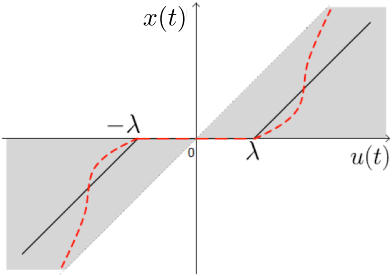




$$\min_x \lambda \sum_n C(x_n) + \frac{1}{2}\|\Phi x - y\|_2^2$$

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

LCA convergence



Assumptions

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

LCA convergence

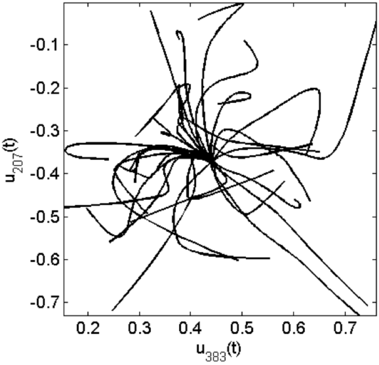
Global asymptotic convergence:

If 1–3 hold above, then the outputs stop moving eventually:

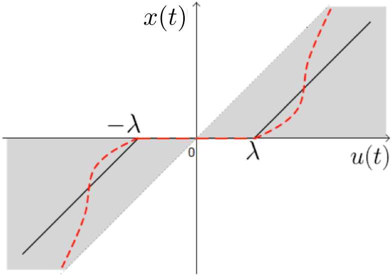
$$\dot{x}(t) \to 0 \quad \text{as} \quad t \to \infty$$

If the critical points of the cost function are isolated then

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



LCA convergence



Assumptions

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

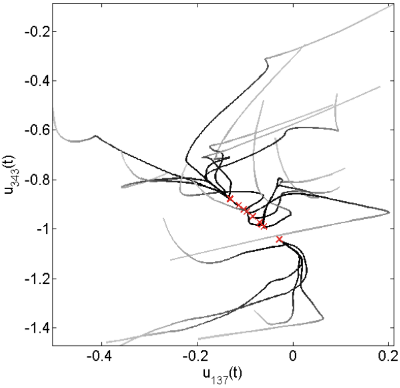
LCA convergence

Global asymptotic convergence:

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

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

where x^* is a critical point of the functional.

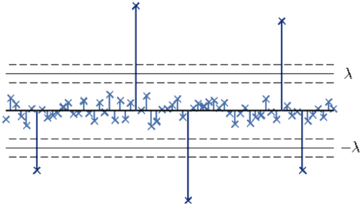


Convergence: support is recovered in finite time

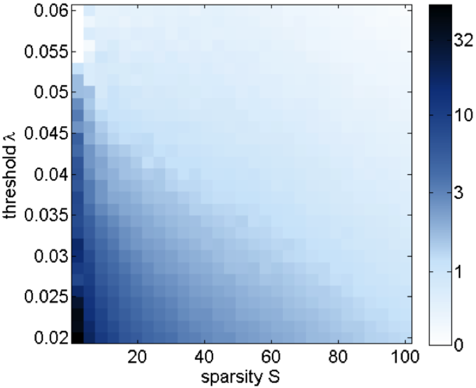
If the LCA converges to a fixed point u^* such that

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

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



# of switches/sparsity



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

Convergence: exponential (of a sort)

In addition to the conditions for global convergence, if there exists $0 \leq \delta < 1$ such that for all $t \geq 0$

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

where $\tilde{x}(t) = x(t) - x^*$, and $\alpha d < 1$ ($f'(u) \leq \alpha$), then the LCA exponentially converges to a unique fixed point:

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

Convergence: exponential (of a sort)

In addition to the conditions for global convergence, if there exists $0 \leq \delta < 1$ such that for all $t \geq 0$

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

where $\tilde{x}(t) = x(t) - x^*$, and $\alpha d < 1$ ($f'(u) \leq \alpha$), then the LCA exponentially converges to a unique fixed point:

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

Of course, this depends on not too many nodes being active at any one time ...

Activation in proper subsets for ℓ_1

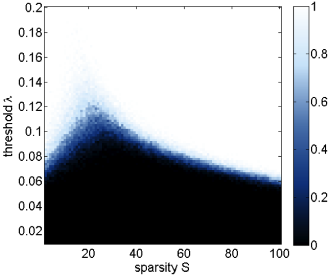
If Φ a "random compressed sensing matrix" and

$$M \;\geq\; \mathrm{Const} \cdot S^2 \log(N/S)$$

then for reasonably small values of λ and starting from rest

$$\Gamma(t) \subset \Gamma^*$$

That is, only subsets of the true support are ever active



Similar results for OMP and homotopy algorithms in CS literature

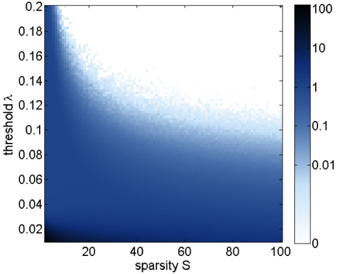
Efficient activation for ℓ_1

If Φ a "random compressed sensing matrix" and

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

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

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," submitted to IEEE Transactions on Signal Processing, July 2012.
- M. Asif and J. Romberg, "Sparse recovery of streaming signals using ℓ_1 homotopy," submitted to IEEE Transactions on Signal Processing, June 2013.
- 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, vol. 23, no. 9, pp. 13771389, 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, late 2013.
- A. Balavoine, J. Romberg, and C. Rozell, "Convergence of a neural network for sparse approximation using the nonsmooth ojasiewicz inequality," Proc. of Int. Joint. Conf. on Neural Networks, August 2013.

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