Streaming sparse recovery: ℓ_1 filtering

- Solving an optimization program like

$$\min_x \ \tau\|x\|_{\ell_1} + \frac{1}{2}\|\Phi x - y\|_2^2$$

  can be costly

- We want to *update* the solution when
  1. the underlying signal changes slightly, or
  2. we add measurements

Duality and optimality conditions

Most of the work is done by deriving *optimality conditions* (a version of KKT) for the solution.

We can show that a vector $x^\star$ supported on Γ is the unique solution to

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

if

$$\begin{aligned} \Phi_\Gamma^T(y - \Phi x^\star) &= \tau\,\mathrm{sgn}(x_\Gamma^\star) \quad \text{on } \Gamma \\ \|\Phi_{\Gamma^c}^T(y - \Phi x^\star)\|_\infty &\le \tau \quad \text{on } \Gamma^c \end{aligned}$$

(Show this on the board ...)

Variable τ

Given the support Γ, the non-zero components of the solution $x^\star$ obey

$$x^\star_\Gamma = (\Phi_\Gamma^T \Phi_\Gamma)^{-1} \Phi_\Gamma^T y - \tau (\Phi_\Gamma^T \Phi_\Gamma)^{-1} \operatorname{sgn}(x^\star_\Gamma)$$

If we were to nudge τ just a little, the solution would move like

$$\partial x = \begin{cases} (\Phi_\Gamma^T \Phi_\Gamma)^{-1} \operatorname{sgn}(x^\star_\Gamma) & \text{on } \Gamma \\ 0 & \text{on } \Gamma^c \end{cases}$$

This direction holds until a component disappears, or a new dual constraint becomes active.

$\Rightarrow$ as we change τ, the path of solutions is piecewise linear

Time-varying sparse signals

- Initial measurements. Observe

$$y_0 = \Phi x_0 + e_0$$

- Initial reconstruction. Solve

$$\min_x \ \tau\|x\|_{\ell_1} + \frac{1}{2}\|\Phi x - y_0\|_2^2$$

- A new set of measurements arrives:

$$y_1 = \Phi x_1 + e_1$$

- Reconstruct again using ℓ_1-min:

$$\min_x \ \tau\|x\|_{\ell_1} + \frac{1}{2}\|\Phi x - y_1\|_2^2$$

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

$$\min \ \tau\|x\|_{\ell_1} + \frac{1}{2}\|\Phi x - (1-\epsilon)y_0 - \epsilon y_1\|_2^2$$

Take ϵ from $0 \to 1$

Update direction

$$\min \ \tau\|x\|_{\ell_1} + \frac{1}{2}\|\Phi x - (1-\epsilon)y_{\text{old}} - \epsilon y_{\text{new}}\|_2^2$$

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

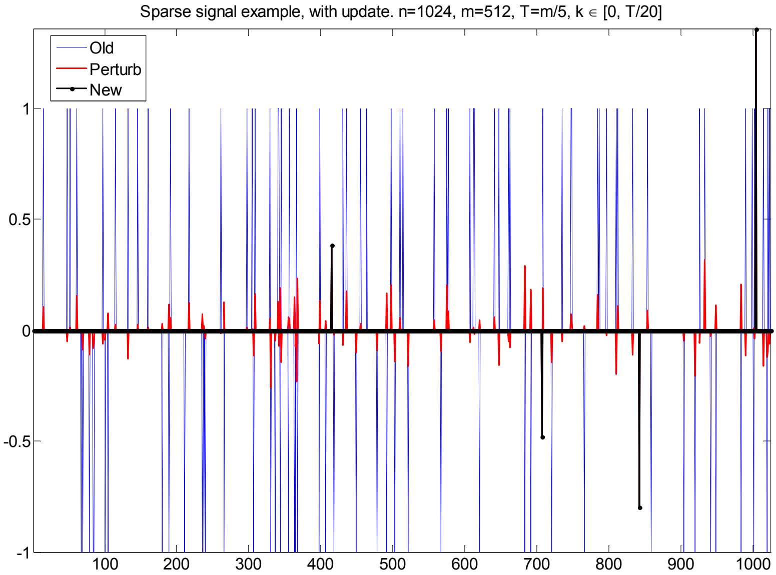
$$\begin{aligned} \Phi_\Gamma^T(\Phi x - (1-\epsilon)y_{\text{old}} - \epsilon y_{\text{new}}) &= -\tau \,\text{sign}\, x_\Gamma \\ \|\Phi_{\Gamma^c}^T(\Phi x - (1-\epsilon)y_{\text{old}} - \epsilon y_{\text{new}})\|_\infty &< \tau \end{aligned}$$

  Γ = active support

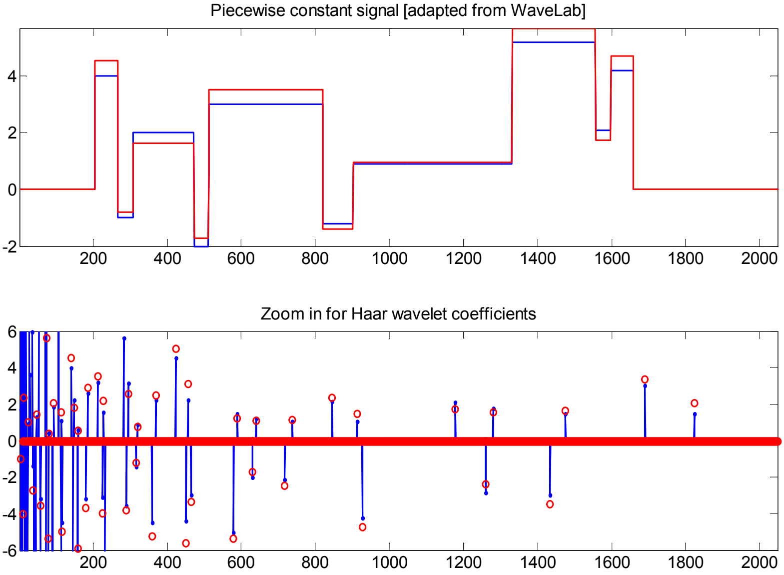
- Update direction:

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

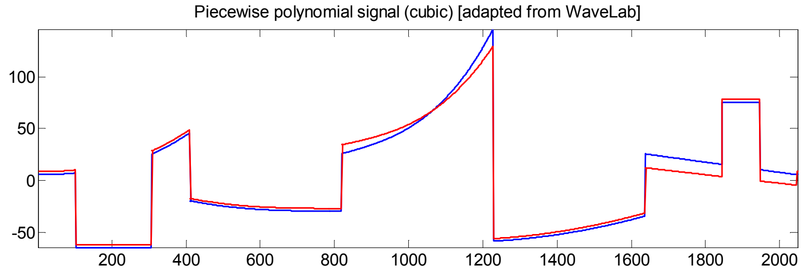
Experiments

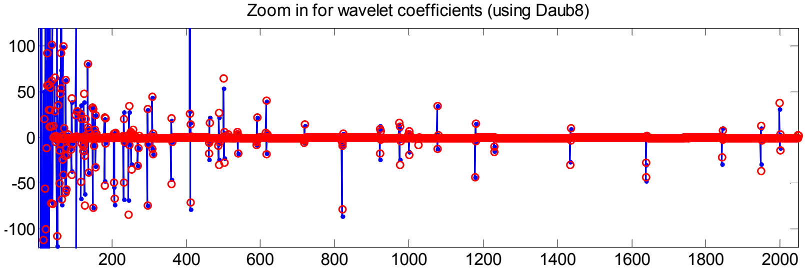


Experiments

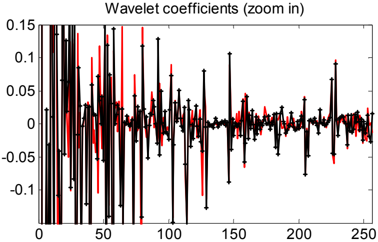


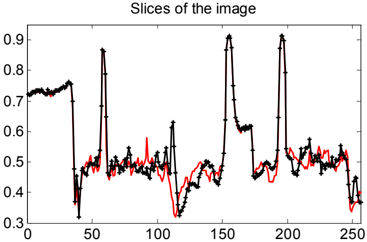
Experiments





Experiments





Experiments

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)

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

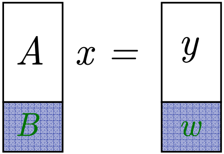
nProdAtA: roughly the avg. no. of matrix vector products with A and A^T
CPU: average cputime to solve

Adding a measurement: Recursive least-squares

- Classical least-squares:

solve a system of linear eqns	$y = Ax + e$
min energy solution	$\min_x \; \|Ax - y\|_2^2$
analytical solution	$\hat{x} = (A^T A)^{-1} A^T y$

- Suppose we add new measurements $w = B^T x$



$$\hat{x}_0 = (A^T A + B^T B)^{-1}(A^T y + B^T w)$$

- Recursive Least-Squares (RLS): easy low-rank update

$$\hat{x}_1 = \hat{x}_0 + (I + B(A^T A)^{-1} B^T)^{-1} (A^T A)^{-1} B^T (w - B\hat{x}_0)$$

Adding a measurement: Dynamic ℓ_1

- We want the analog of RLS for the LASSO. Adding one measurement

$$\begin{bmatrix} y \\ w \end{bmatrix} = \begin{bmatrix} \Phi \\ b \end{bmatrix} x + \begin{bmatrix} e \\ d \end{bmatrix} \quad \longrightarrow \quad \min_x \; \tau \|x\|_{\ell_1} + \frac{1}{2}\|\Phi x - y\|_2^2 + \frac{1}{2}\|bx - w\|_2^2$$

- Challenges:
  - not as smooth as least-squares update
  - solution can change drastically with just one new measurement
  - need to move slowly, use a *homotopy* method

(see also work by Garrigues et al. 08)

Dynamic ℓ_1 update

- Work in the new measurement slowly

$$\min \ \tau\|x\|_{\ell_1} + \frac{1}{2}\left(\|\Phi x - y\|_2^2 + \epsilon\|bx - w\|_2^2\right)$$

  Again, the solution path is piecewise linear in ϵ

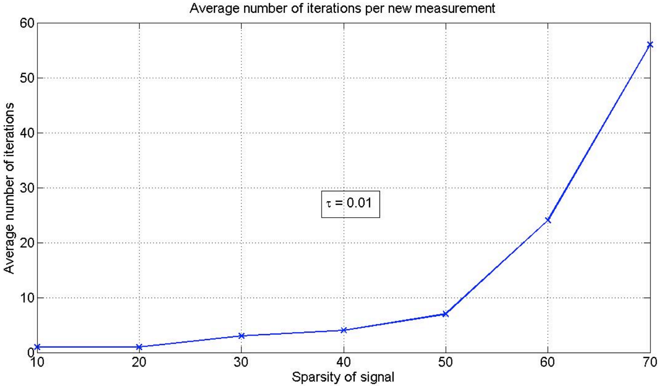
- Optimality conditions

$$\Phi_\Gamma^T(\Phi x - y) + \epsilon b_\Gamma^T(bx - w) = -\tau \operatorname{sign} x_\Gamma$$
$$\|\Phi_{\Gamma^c}^T(\Phi x - y) + \epsilon b_{\Gamma^c}^T(bx - w)\|_\infty < \tau$$

- From critical point x^{ϵ_0}, update direction is

$$\partial x = \begin{cases} (\Phi_\Gamma^T\Phi_\Gamma + \epsilon_0 b_\Gamma^T b_\Gamma))^{-1} b_\Gamma^T(w - bx^{\epsilon_0}) & \text{on } \Gamma \\ 0 & \text{off } \Gamma \end{cases}$$

Number of steps per update



Measurements $m = 150$
Signal length $n = 256$

Summary of ℓ_1 filtering

- Instead of solving new programs from scratch, work the new data in slowly using homotopy continuation
- Proper homotopy formulation allows us to (easily) use optimality conditions to "hop" along the path of solutions
- Each "hop" costs $O(mn)$ — a few matrix-vector multiplies
- Small number of "hops" if the solutions are closely related