Seismic trace interpolation via sparsity promoting reweighted algorithms

Hassan Mansour, Ozgur Yilmaz, Felix Herrmann, and
Tristan van Leeuwen

SLIM Consortium meeting

SLIM
Seismic Laboratory for Imaging and Modeling
the University of British Columbia

Collaboration

Joint work in part with:

- Özgür Yılmaz (UBC, Mathematics)
- Rayan Saab (Duke University, Mathematics)
- Michael Friedlander (UBC, Computer Science)
- Felix Herrmann (UBC, Earth and Ocean Science)
- Tristan Van Leeuwen (UBC, Earth and Ocean Science)

Outline

Part 1: Compressed sensing and sparse recovery

- Overview of sparse recovery from sub-Nyquist sampling.

Outline

Part 1: Compressed sensing and sparse recovery

- Overview of sparse recovery from sub-Nyquist sampling.

Part 2: Weighted ℓ_1 minimization

- Sparse recovery with partial support information.

Outline

Part 1: Compressed sensing and sparse recovery

- Overview of sparse recovery from sub-Nyquist sampling.

Part 2: Weighted ℓ_1 minimization

- Sparse recovery with partial support information.

Part 3: Optimization for sparse recovery

- The WSPGL1 algorithm.

Outline

Part 1: Compressed sensing and sparse recovery

- Overview of sparse recovery from sub-Nyquist sampling.

Part 2: Weighted ℓ_1 minimization

- Sparse recovery with partial support information.

Part 3: Optimization for sparse recovery

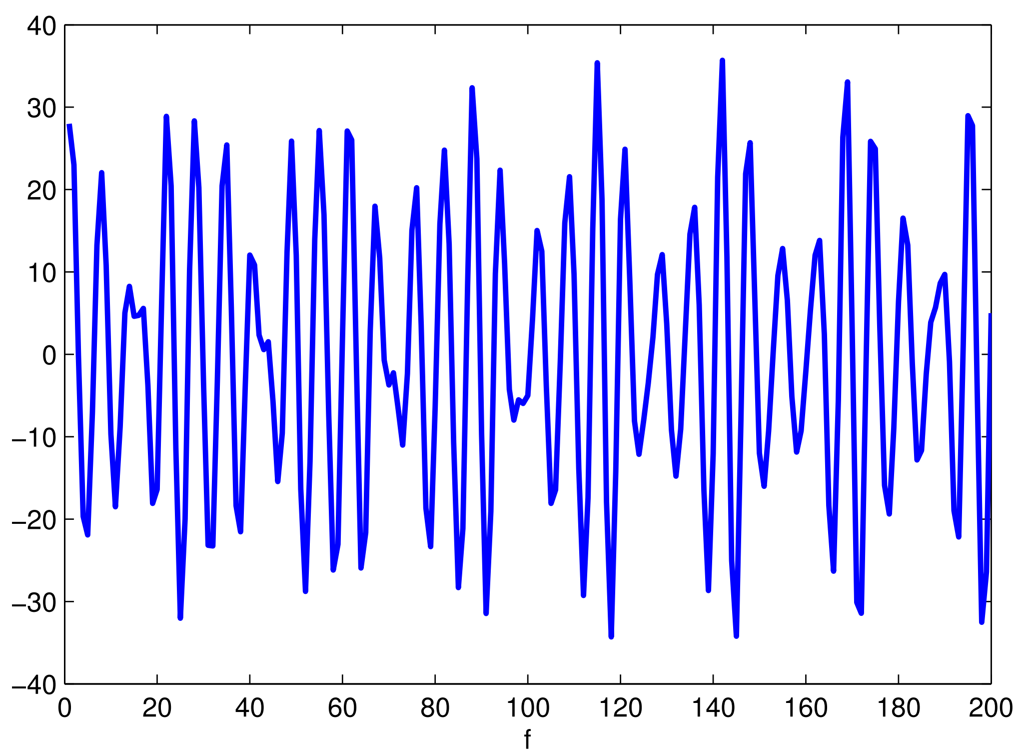
- The WSPGL1 algorithm.

Part 4: Sparse randomized Kaczmarz

- Application to least-squares migration.

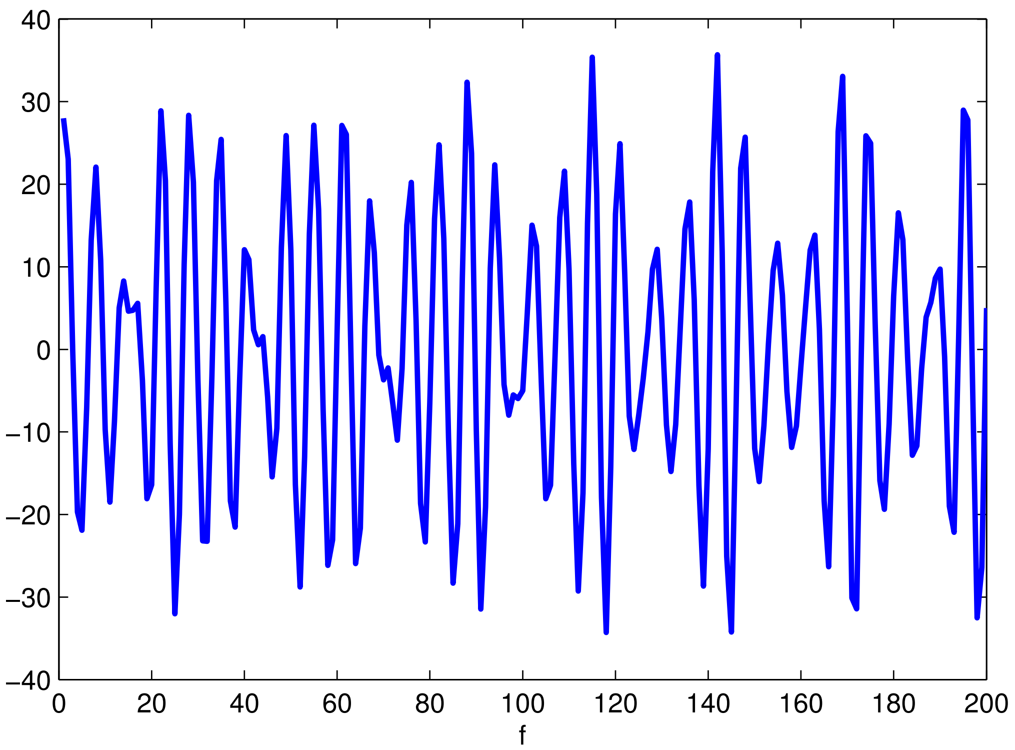
Compressed sensing: sub-Nyquist data acquisition

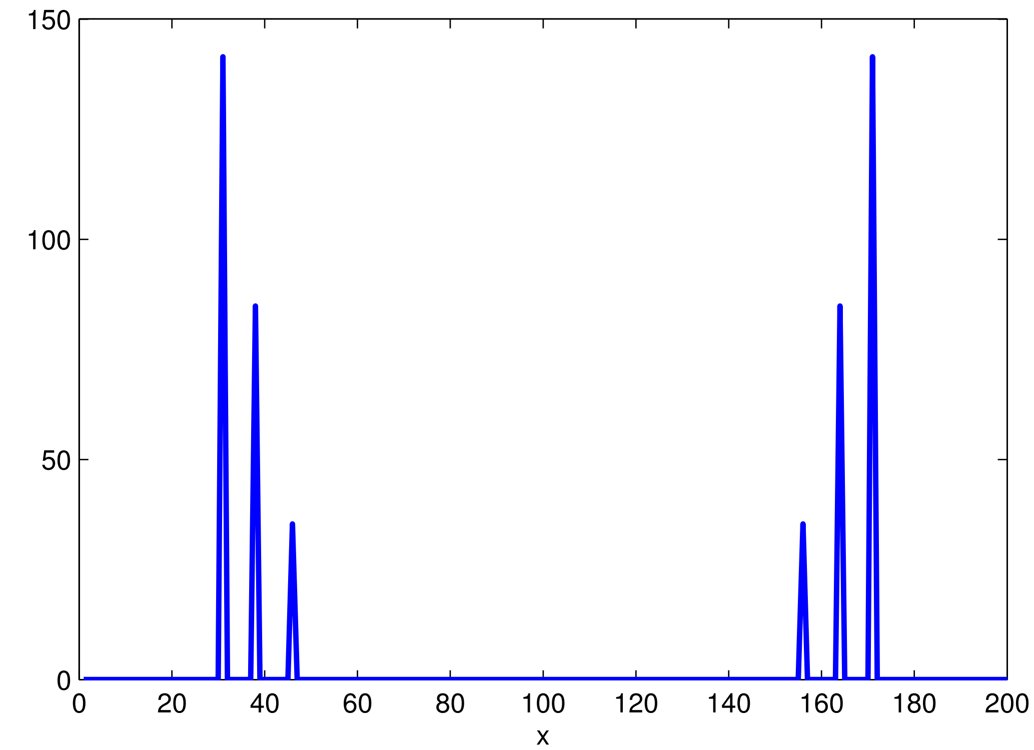
- We wish to acquire a signal $\mathbf{f}$ using compressive measurements $\mathbf{y}$.
- $\mathbf{f}$ admits a sparse or compressible representation $\mathbf{x}$ in some domain $\mathbf{D}$.
- Shannon-Nyquist sampling imposes a sampling interval $T \geq \frac{1}{2\Omega}$ (e.g. ≥ 90 samples).
- Compressed sensing addresses the question of how to recovery $\mathbf{x}$ from sub-Nyquist measurements $\mathbf{y}$ (e.g. around 50 random samples).



Compressed sensing: sub-Nyquist data acquisition

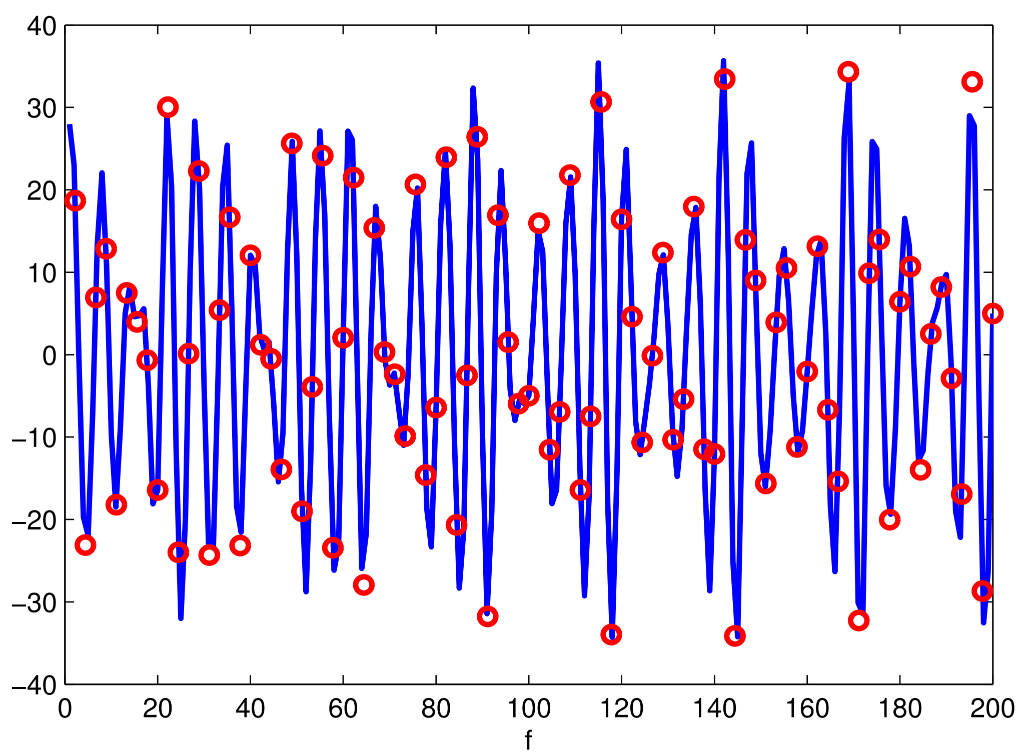
- We wish to acquire a signal $\mathbf{f}$ using compressive measurements $\mathbf{y}$.
- $\mathbf{f}$ admits a sparse or compressible representation $\mathbf{x}$ in some domain $\mathbf{D}$.
- Shannon-Nyquist sampling imposes a sampling interval $T \geq \frac{1}{2\Omega}$ (e.g. ≥ 90 samples).
- Compressed sensing addresses the question of how to recovery $\mathbf{x}$ from sub-Nyquist measurements $\mathbf{y}$ (e.g. around 50 random samples).

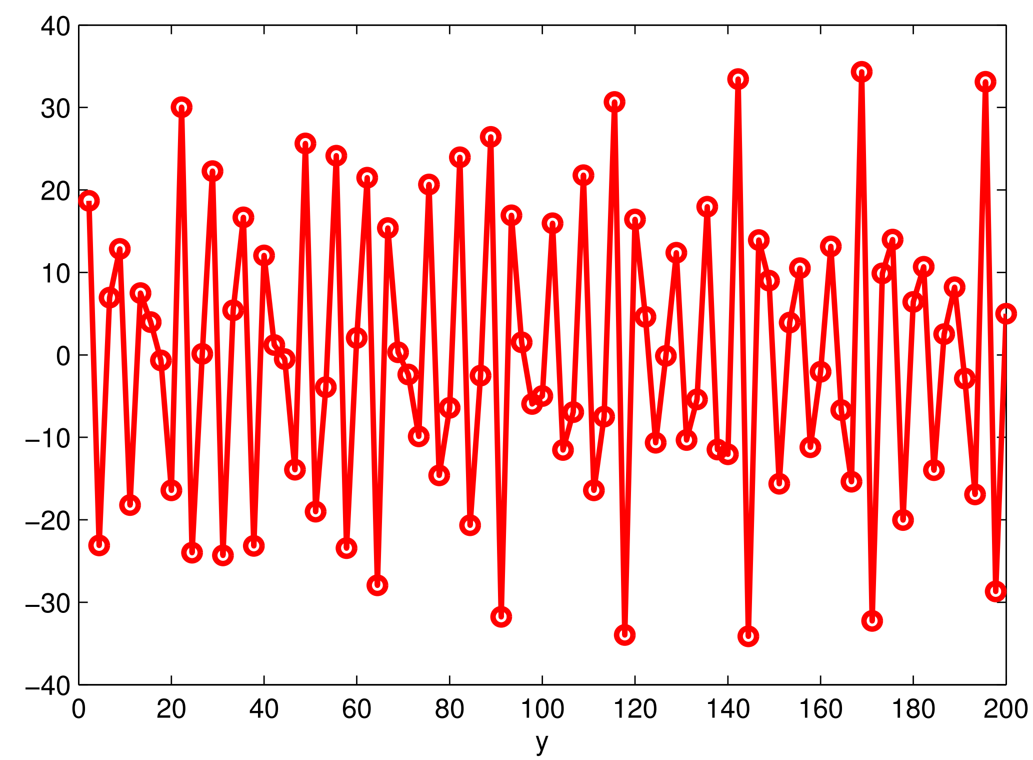




Compressed sensing: sub-Nyquist data acquisition

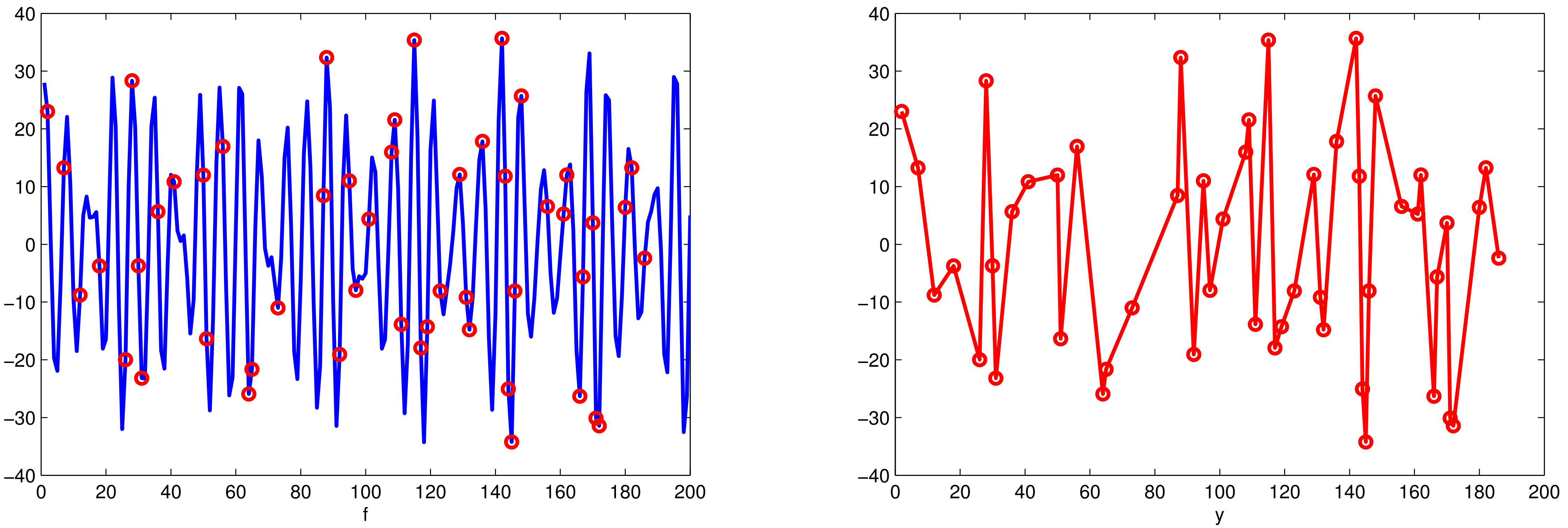
- We wish to acquire a signal $\mathbf{f}$ using compressive measurements $\mathbf{y}$.
- $\mathbf{f}$ admits a sparse or compressible representation $\mathbf{x}$ in some domain $\mathbf{D}$.
- Shannon-Nyquist sampling imposes a sampling interval $T \geq \frac{1}{2\Omega}$ (e.g. ≥ 90 samples).
- Compressed sensing addresses the question of how to recovery $\mathbf{x}$ from sub-Nyquist measurements $\mathbf{y}$ (e.g. around 50 random samples).





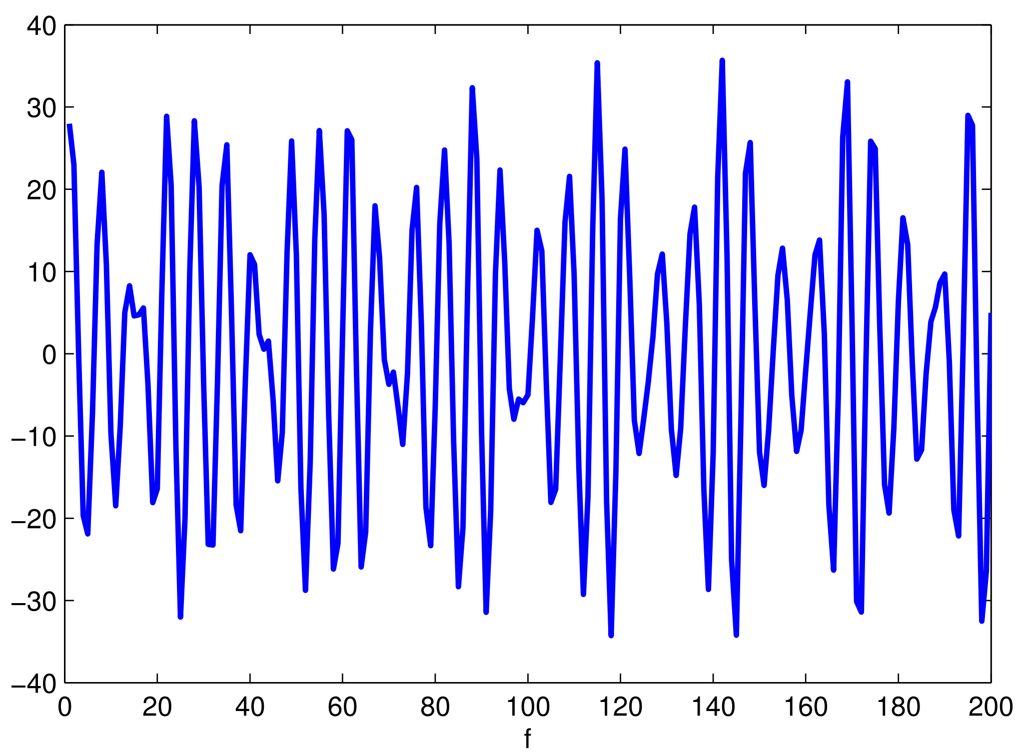
Compressed sensing: sub-Nyquist data acquisition

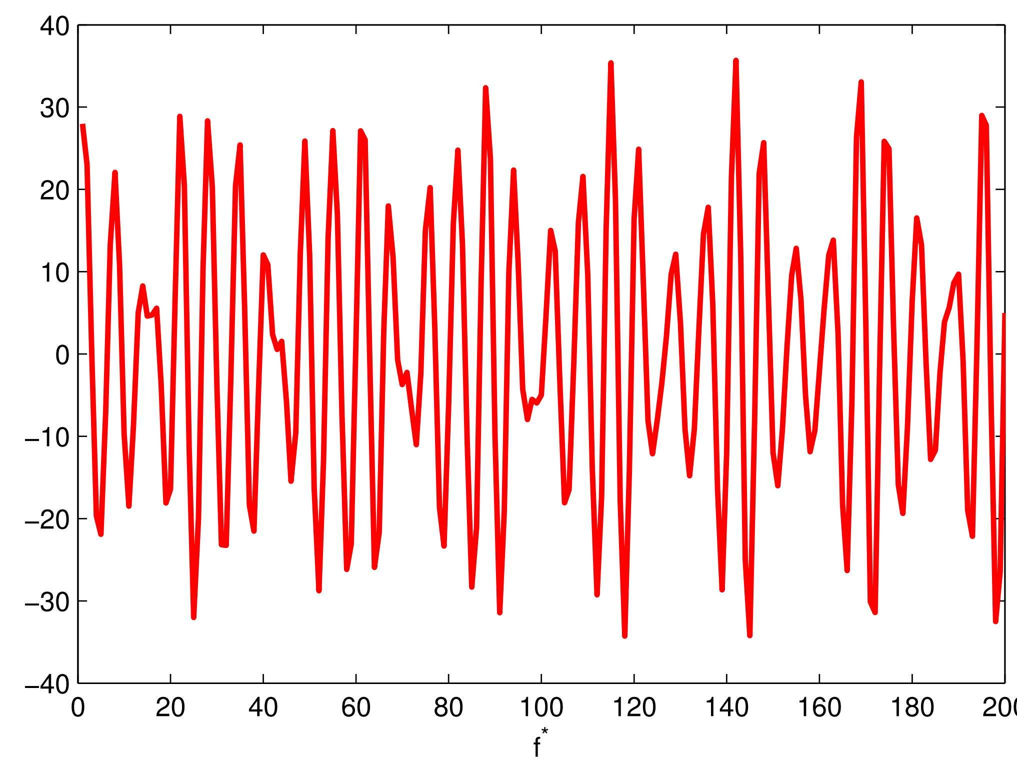
- We wish to acquire a signal $\mathbf{f}$ using compressive measurements $\mathbf{y}$.
- $\mathbf{f}$ admits a sparse or compressible representation $\mathbf{x}$ in some domain $\mathbf{D}$.
- Shannon-Nyquist sampling imposes a sampling interval $T \geq \frac{1}{2\Omega}$ (e.g. ≥ 90 samples).
- Compressed sensing addresses the question of how to recovery $\mathbf{x}$ from sub-Nyquist measurements $\mathbf{y}$ (e.g. around 50 random samples).



Compressed sensing: sub-Nyquist data acquisition

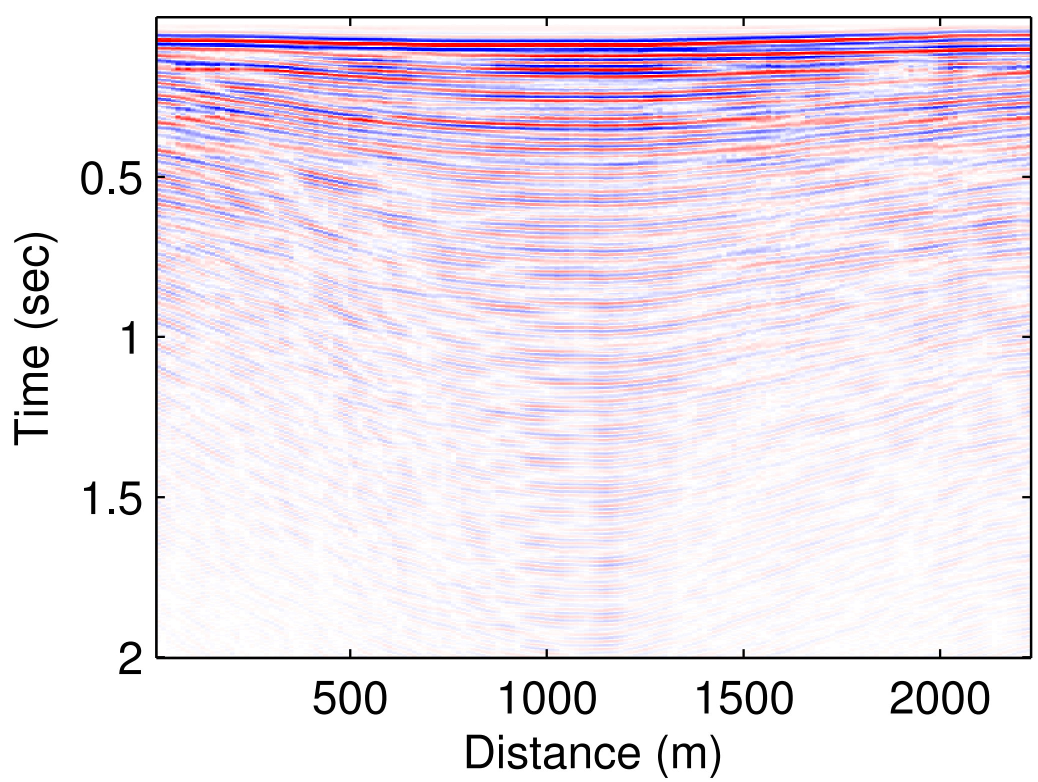
- We wish to acquire a signal $\mathbf{f}$ using compressive measurements $\mathbf{y}$.
- $\mathbf{f}$ admits a sparse or compressible representation $\mathbf{x}$ in some domain $\mathbf{D}$.
- Shannon-Nyquist sampling imposes a sampling interval $T \geq \frac{1}{2\Omega}$ (e.g. ≥ 90 samples).
- Compressed sensing addresses the question of how to recovery $\mathbf{x}$ from sub-Nyquist measurements $\mathbf{y}$ (e.g. around 50 random samples).

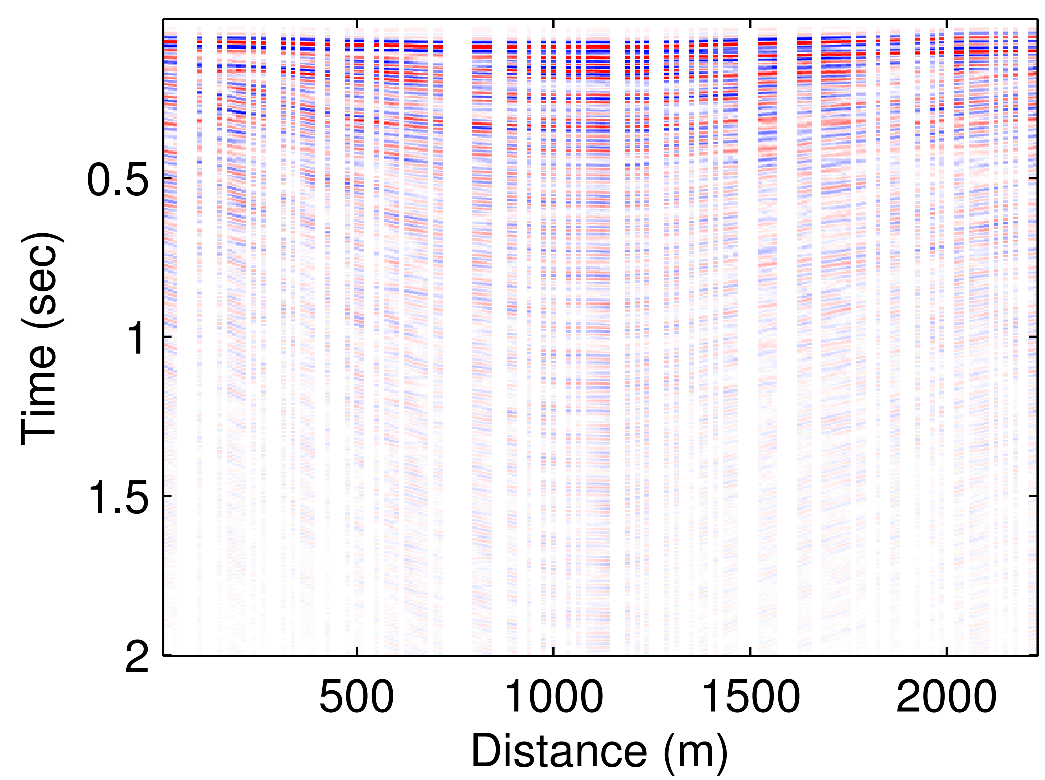




Example: Seismic data interpolation

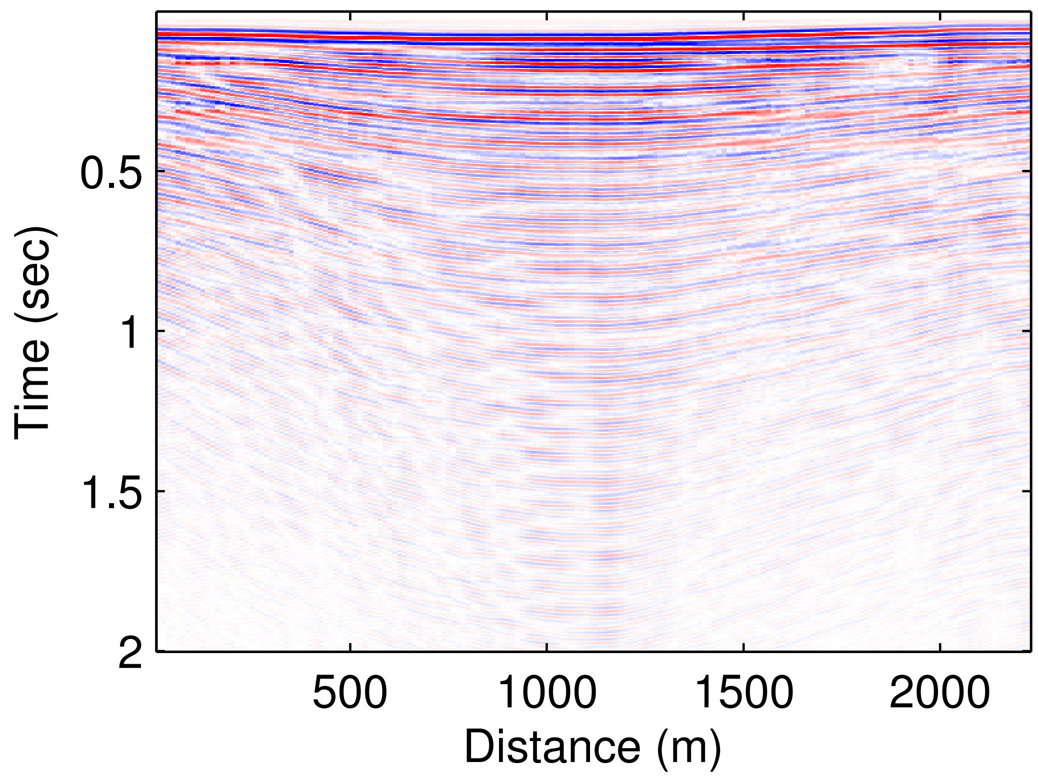
- Economical acquisition of seismic traces that are sparse in the curvelet domain.

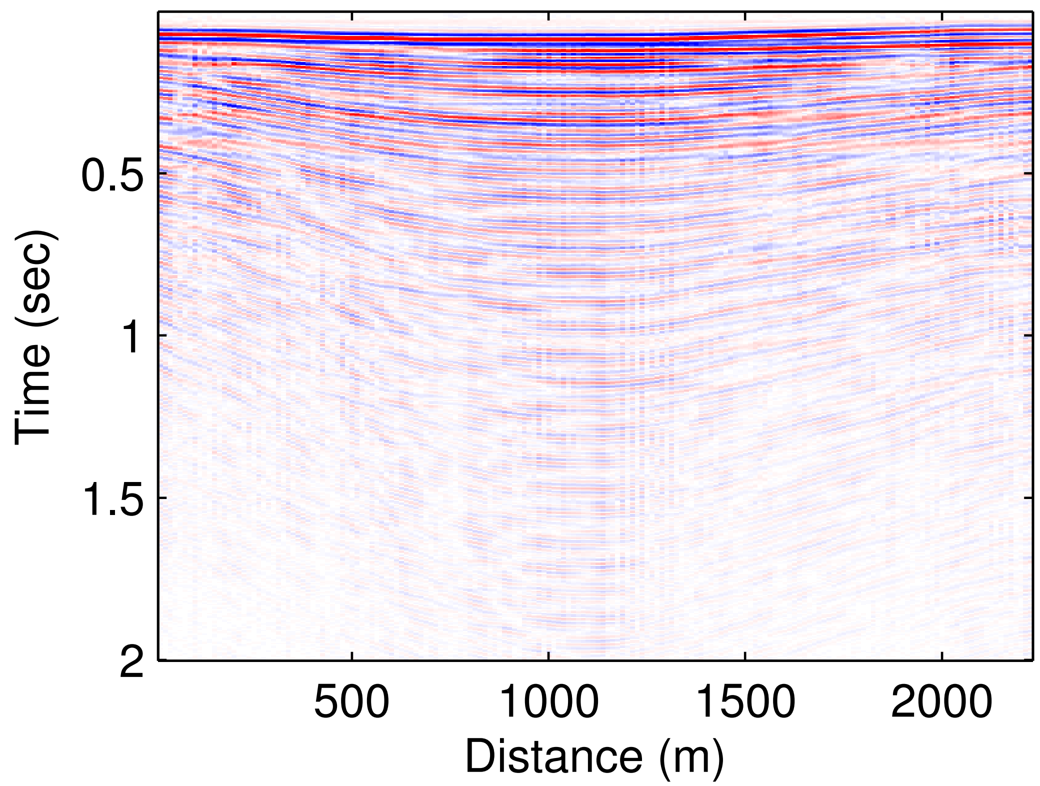




Example: Seismic data interpolation

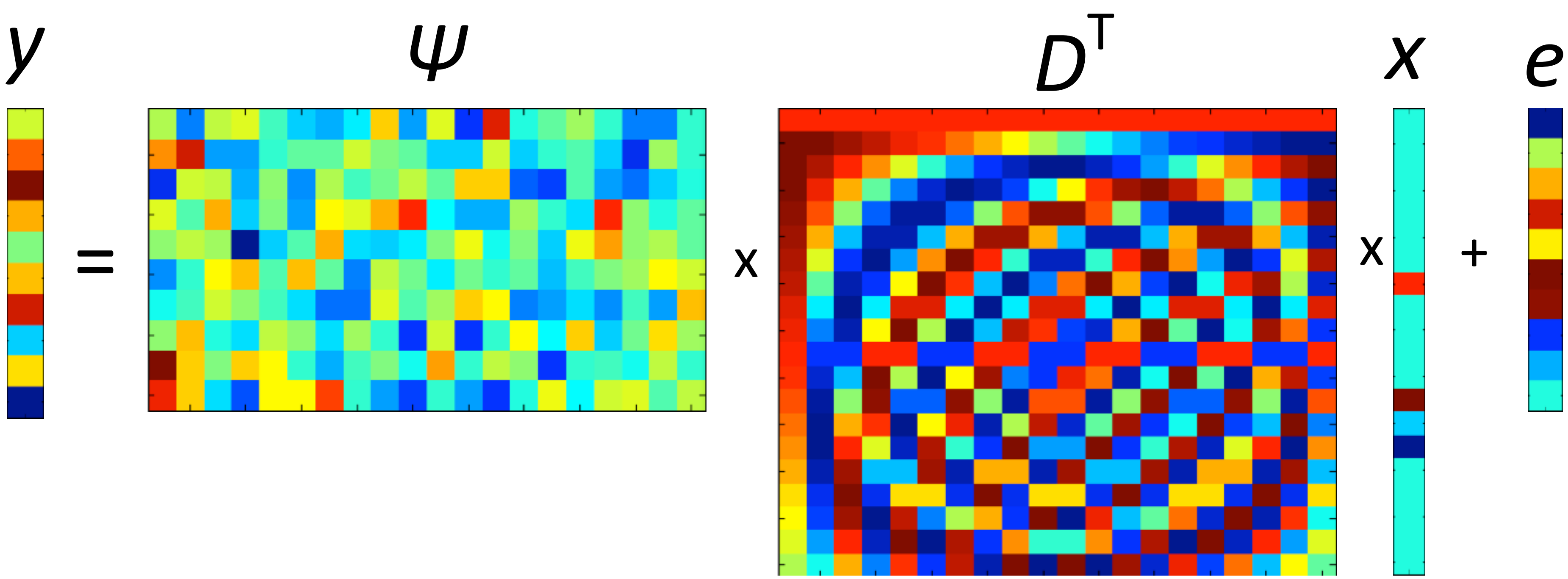
- Economical acquisition of seismic traces that are sparse in the curvelet domain.





Compressed sensing basics

- We want to recover a k-sparse signal $\mathbf{x} \in \mathbb{R}^N$.
- Given $n \ll N$ linear and noisy sub-Nyquist measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $A = \mathbf{\Psi}\mathbf{D}^T$.
- Under certain conditions on $\mathbf{x}$ and $\mathbf{A}$, the signal $\mathbf{x}$ can be recovered from $\mathbf{y}$ by solving certain optimization problems:
  - The combinatorial ℓ_0 minimization problem.
  - The polynomial-time ℓ_1 minimization problem.
  - Other algorithms, e.g.: OMP, CoSaMP, AMP, IRLS ...



Compressed sensing basics

- We want to recover a k-sparse signal $\mathbf{x} \in \mathbb{R}^N$.
- Given $n \ll N$ linear and noisy sub-Nyquist measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $A = \mathbf{\Psi}\mathbf{D}^T$.
- Under certain conditions on $\mathbf{x}$ and $\mathbf{A}$, the signal $\mathbf{x}$ can be recovered from $\mathbf{y}$ by solving certain optimization problems:
  - The combinatorial ℓ_0 minimization problem.
  - The polynomial-time ℓ_1 minimization problem.
  - Other algorithms, e.g.: OMP, CoSaMP, AMP, IRLS,...

Definition: Restricted Isometry Property (RIP) (Candés and Tao '05)

The RIP constant δ_k is defined as the smallest constant such that $\forall x \in \Sigma_k^N$

$$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2,$$

Compressed sensing basics

- We want to recover a k-sparse signal $\mathbf{x} \in \mathbb{R}^N$.
- Given $n \ll N$ linear and noisy sub-Nyquist measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $A = \mathbf{\Psi}\mathbf{D}^T$.
- Under certain conditions on $\mathbf{x}$ and $\mathbf{A}$, the signal $\mathbf{x}$ can be recovered from $\mathbf{y}$ by solving certain optimization problems:
  - The combinatorial ℓ_0 minimization problem.
  - The polynomial-time ℓ_1 minimization problem.
  - Other algorithms, e.g.: OMP, CoSaMP, AMP, IRLS,...

Constrained ℓ_0-minimization

- $$\min_{\mathbf{u} \in \mathbb{R}^N} \|\mathbf{u}\|_0 \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{x}$$

Compressed sensing basics

- We want to recover a k-sparse signal $\mathbf{x} \in \mathbb{R}^N$.
- Given $n \ll N$ linear and noisy sub-Nyquist measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $A = \mathbf{\Psi}\mathbf{D}^T$.
- Under certain conditions on $\mathbf{x}$ and $\mathbf{A}$, the signal $\mathbf{x}$ can be recovered from $\mathbf{y}$ by solving certain optimization problems:
  - The combinatorial ℓ_0 minimization problem.
  - The polynomial-time ℓ_1 minimization problem.
  - Other algorithms, e.g.: OMP, CoSaMP, AMP, IRLS,...

Constrained ℓ_1-minimization

- $$\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{u}\|_1 \quad \text{subject to } \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \le \|\mathbf{e}\|_2, \quad \|\mathbf{u}\|_1 = \sum_{i=1}^{N} |u_i|$$

Compressed sensing basics

- We want to recover a k-sparse signal $\mathbf{x} \in \mathbb{R}^N$.
- Given $n \ll N$ linear and noisy sub-Nyquist measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $A = \mathbf{\Psi}\mathbf{D}^T$.
- Under certain conditions on $\mathbf{x}$ and $\mathbf{A}$, the signal $\mathbf{x}$ can be recovered from $\mathbf{y}$ by solving certain optimization problems:
  - The combinatorial ℓ_0 minimization problem.
  - The polynomial-time ℓ_1 minimization problem.
  - Other algorithms, e.g.: OMP, CoSaMP, AMP, IRLS,...

Stability and Robustness

- If $k < n/2$ and $\mathbf{A}$ has the RIP with $\delta_{2k} < 1$, then ℓ_0 minimization recovers $\mathbf{x}$ exactly.
- When $k \lesssim n/\log(N/n)$ and under stricter conditions on the RIP of $\mathbf{A}$, solving the ℓ_1-minimization problem also recovers $\mathbf{x}$.

Stability and Robustness

- If $k < n/2$ and $\mathbf{A}$ has the RIP with $\delta_{2k} < 1$, then ℓ_0 minimization recovers $\mathbf{x}$ exactly.
- When $k \lesssim n/\log(N/n)$ and under stricter conditions on the RIP of $\mathbf{A}$, solving the ℓ_1-minimization problem also recovers $\mathbf{x}$.

Theorem (Candés, Romberg, Tao '06); (Donoho)

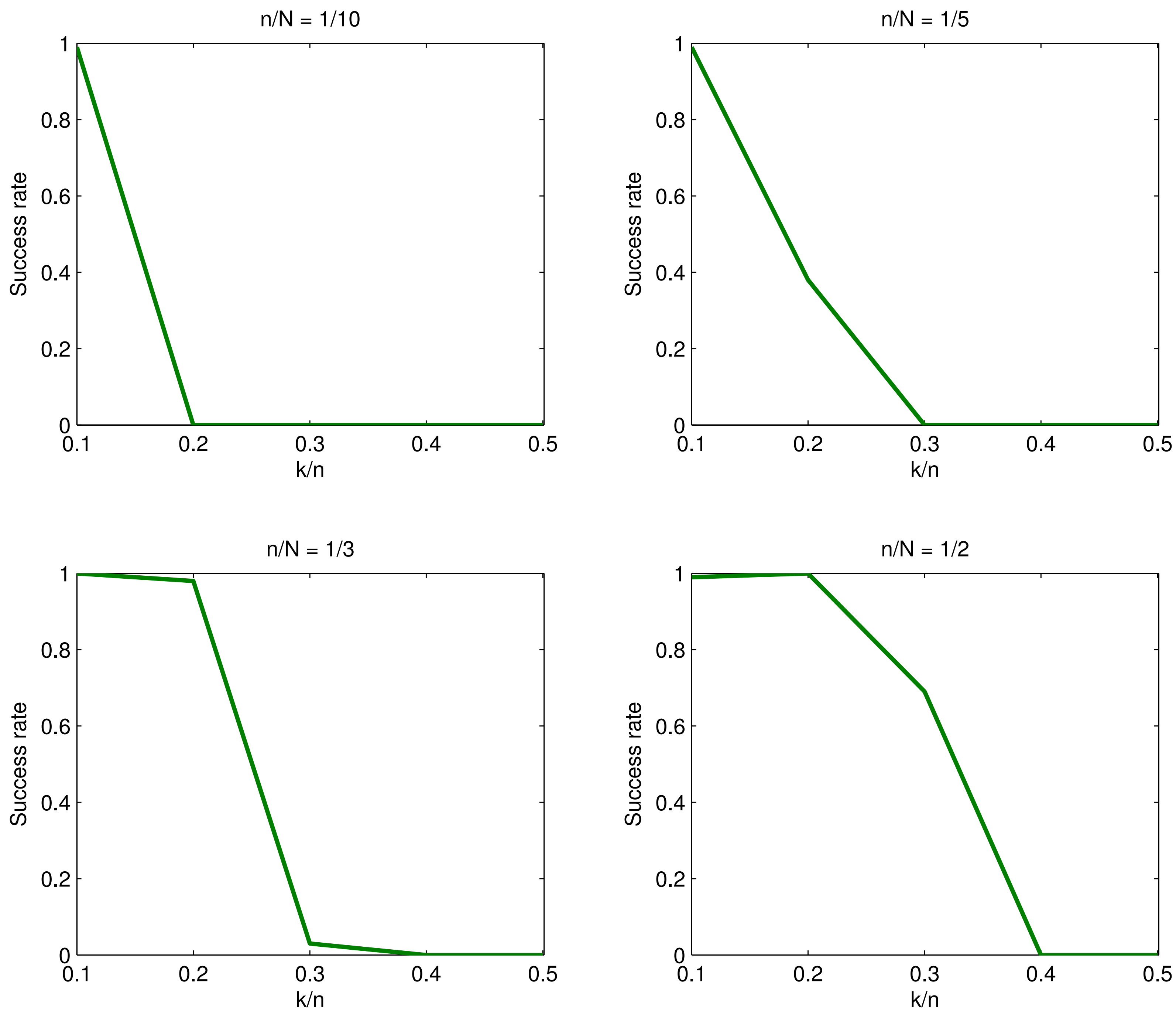
If for some $a > 1$ the matrix $\mathbf{A}$ satisfies the RIP with $\delta_{(a+1)k} < \frac{a-1}{a+1}$,

then the solution $\mathbf{x}^*$ to the ℓ_1 minimization problem obeys

$$\|\mathbf{x}^* - \mathbf{x}\|_2 \leq C_0\|e\|_2^2 + C_1 k^{-1/2}\|\mathbf{x} - \mathbf{x}_k\|_1$$

The $\ell_1 - \ell_0$ gap

- Recovery using ℓ_1 minimization.



Bridging the $\ell_1 - \ell_0$ gap

- Incorporate support information: weighted ℓ_1 minimization (FMSY '12).
- Optimization for sparse recovery: the WSPGL1 algorithm (Mansour '12).

Part 1: Compressed sensing and sparse recovery

Part 2: Weighted ℓ_1 minimization

Part 3: ℓ_1 solvers and the WSPGL1 algorithm

Part 4: Sparse randomized Kaczmarz

Beyond ℓ_1 minimization

- Suppose k, n and N are such that ℓ_1-minimization fails to recover $\mathbf{x}$.
- Suppose we have prior information on the support of $\mathbf{x}$.
- How do we incorporate this knowledge in the recovery algorithm while keeping the measurement process non-adaptive?

Inexact recovery using ℓ_1 minimization

- Eg. when $k > \hat{k} \approx n/\log(N/n)$

Beyond ℓ_1 minimization

- Suppose k, n and N are such that ℓ_1-minimization fails to recover $\mathbf{x}$.
- Suppose we have prior information on the support of $\mathbf{x}$.
- How do we incorporate this knowledge in the recovery algorithm while keeping the measurement process non-adaptive?

Recovery using prior information

- Eg. when $k > \hat{k} \approx n/\log(N/n)$
- Eg. indices 1, 3, and 6 are non-zero.

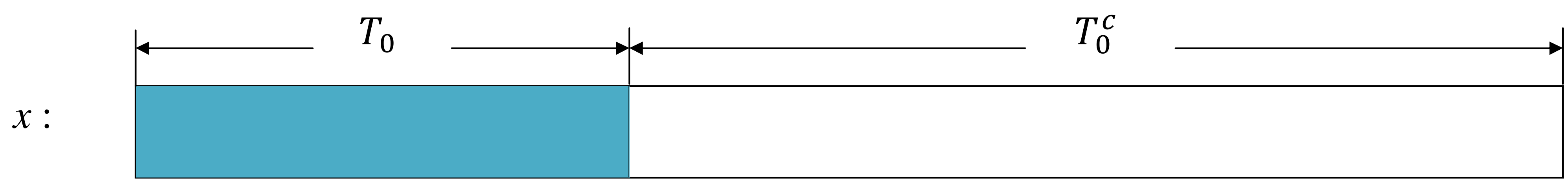
Beyond ℓ_1 minimization

- Suppose k, n and N are such that ℓ_1-minimization fails to recover $\mathbf{x}$.
- Suppose we have prior information on the support of $\mathbf{x}$.
- How do we incorporate this knowledge in the recovery algorithm while keeping the measurement process non-adaptive?

Weighted ℓ_1 minimization

- Suppose that $\mathbf{x}$ is an arbitrary signal in $\mathbb{R}^N$ and let $T_0 = \text{supp}(\mathbf{x}_k)$.
- Let $\widetilde{T}$ be a known support estimate that is partially accurate.
- Define the weighted ℓ_1 norm $\|\mathbf{x}\|_{1,\mathrm{w}} := \sum_i \mathrm{w}_i |x_i|$ and the problem

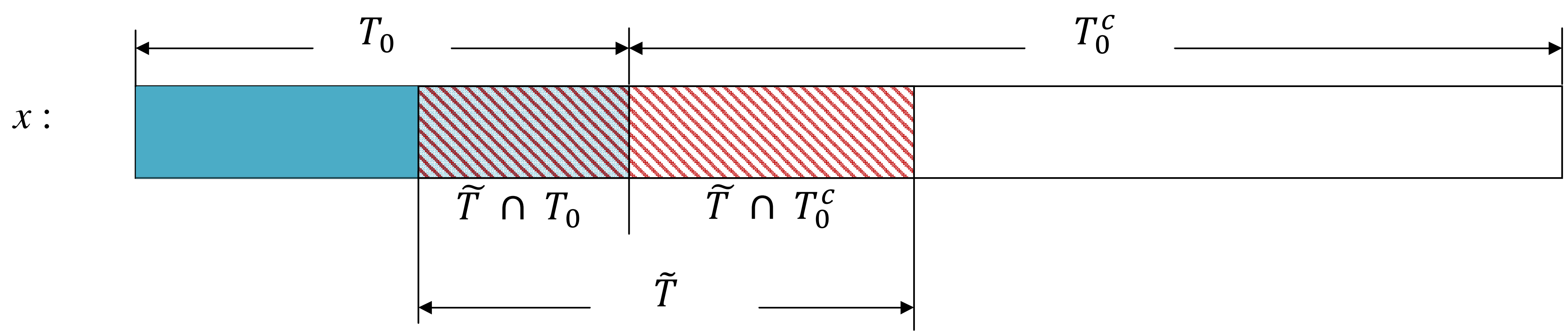
$$\min_{\mathbf{x}} \|\mathbf{x}\|_{1,\mathrm{w}} \text{ subject to } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon \quad \text{with} \quad \mathrm{w}_i = \begin{cases} 1, & i \in \widetilde{T}^c, \\ \omega, & i \in \widetilde{T}. \end{cases}$$



Weighted ℓ_1 minimization

- Suppose that $\mathbf{x}$ is an arbitrary signal in $\mathbb{R}^N$ and let $T_0 = \text{supp}(\mathbf{x}_k)$.
- Let $\widetilde{T}$ be a known support estimate that is partially accurate.
- Define the weighted ℓ_1 norm $\|\mathbf{x}\|_{1,\mathrm{w}} := \sum_i \mathrm{w}_i |x_i|$ and the problem

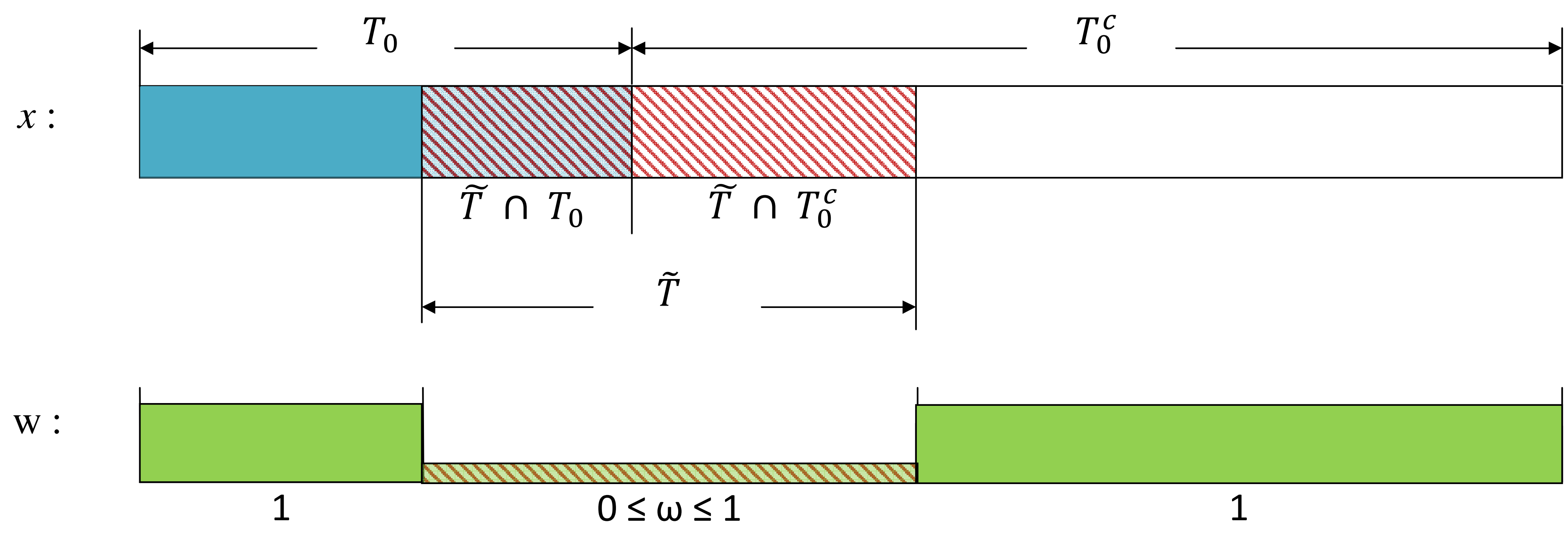
$$\min_{\mathbf{x}} \ \|\mathbf{x}\|_{1,\mathrm{w}} \text{ subject to } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon \quad \text{with} \quad \mathrm{w}_i = \begin{cases} 1, & i \in \widetilde{T}^c, \\ \omega, & i \in \widetilde{T}. \end{cases}$$



Weighted ℓ_1 minimization

- Suppose that $\mathbf{x}$ is an arbitrary signal in $\mathbb{R}^N$ and let $T_0 = \text{supp}(\mathbf{x}_k)$.
- Let $\widetilde{T}$ be a known support estimate that is partially accurate.
- Define the weighted ℓ_1 norm $\|\mathbf{x}\|_{1,\mathrm{w}} := \sum_i \mathrm{w}_i |x_i|$ and the problem

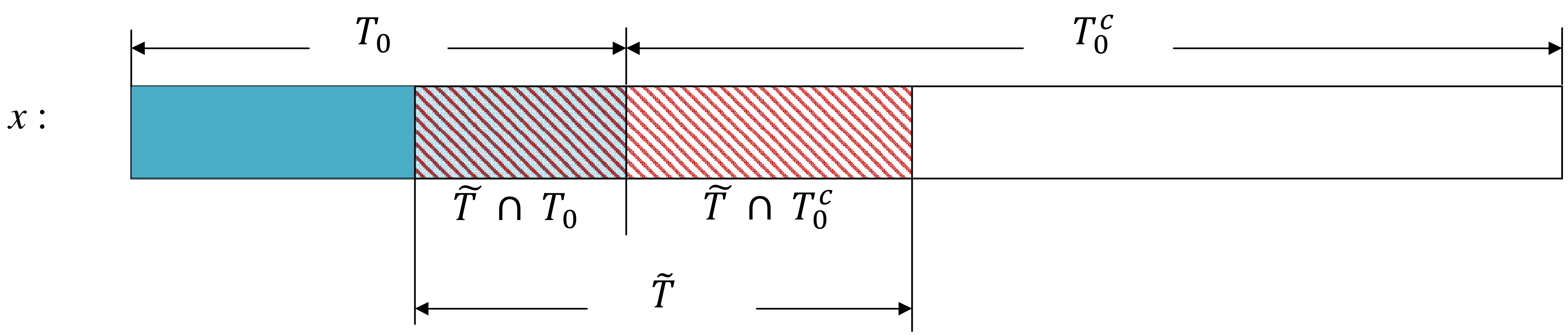
$$\min_{\mathbf{x}} \|\mathbf{x}\|_{1,\mathrm{w}} \text{ subject to } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \le \epsilon \quad \text{with} \quad \mathrm{w}_i = \begin{cases} 1, & i \in \widetilde{T}^c, \\ \omega, & i \in \widetilde{T}. \end{cases}$$



(FMSY '12) (Vaswani and Lu) (Khajehnejad et al.) (L. Jacques)

Stability and Robustness

- Two parameters determine the performance of weighted ℓ_1:
  - $\rho = \frac{|\widetilde{T}|}{|T_0|}$ is the relative size of $\widetilde{T}$.
  - $\alpha = \frac{|\widetilde{T} \cap T_0|}{|\widetilde{T}|}$ is the accuracy of $\widetilde{T}$.



Stability and Robustness

- Two parameters determine the performance of weighted ℓ_1:
  - $\rho = \frac{|\widetilde{T}|}{|T_0|}$ is the relative size of $\widetilde{T}$.
  - $\alpha = \frac{|\widetilde{T} \cap T_0|}{|\widetilde{T}|}$ is the accuracy of $\widetilde{T}$.

Theorem (FMSY '12)

If for some $a \geq (1-\alpha)\rho$, $a > 1$, the matrix $\mathbf{A}$ satisfies $\delta_{(a+1)k} < \frac{a-\gamma^2}{a+\gamma^2}$.

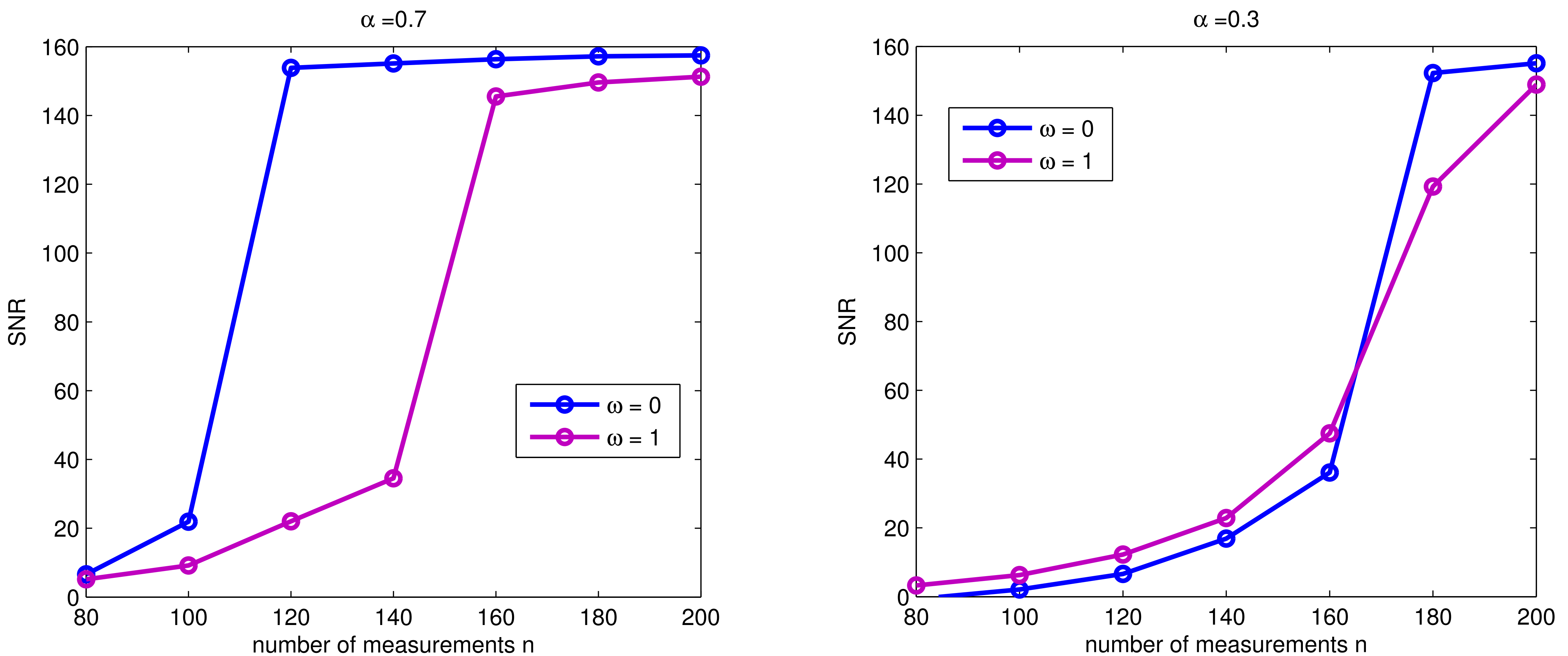
Then the solution $\mathbf{x}^*$ to the weighted ℓ_1 problem obeys

$$\|\mathbf{x}^* - \mathbf{x}\|_2 \leq C_0'(\gamma)\epsilon + C_1'(\gamma)k^{-1/2}\left(\omega\|\mathbf{x}_{T_0^c}\|_1 + (1-\omega)\|\mathbf{x}_{\widetilde{T}^c \cap T_0^c}\|_1\right).$$

- $\gamma = \left(\omega + (1-\omega)\sqrt{1+\rho-2\alpha\rho}\right)$

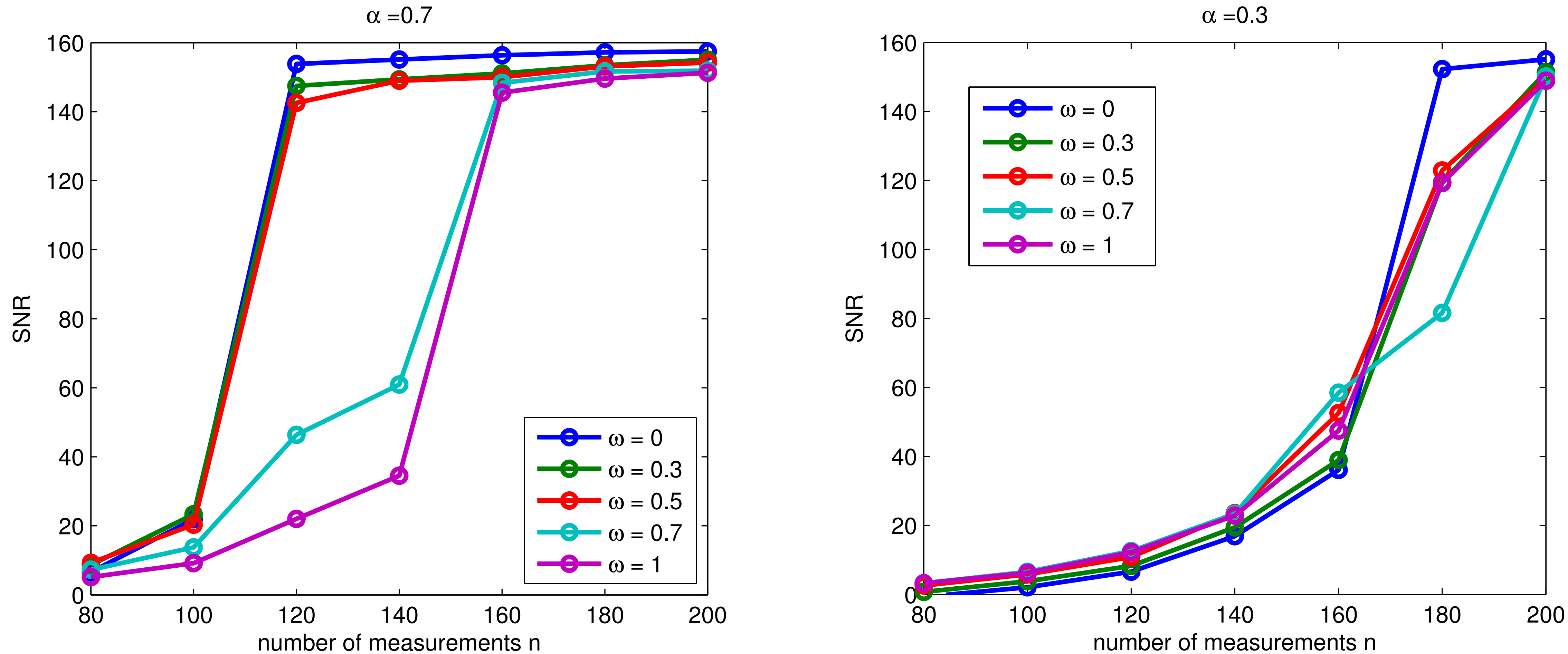
Recovery of Sparse Signals

- SNR averaged over 20 experiments for k-sparse signals x with $k = 40$, and $N = 500$.
- The noise free case:



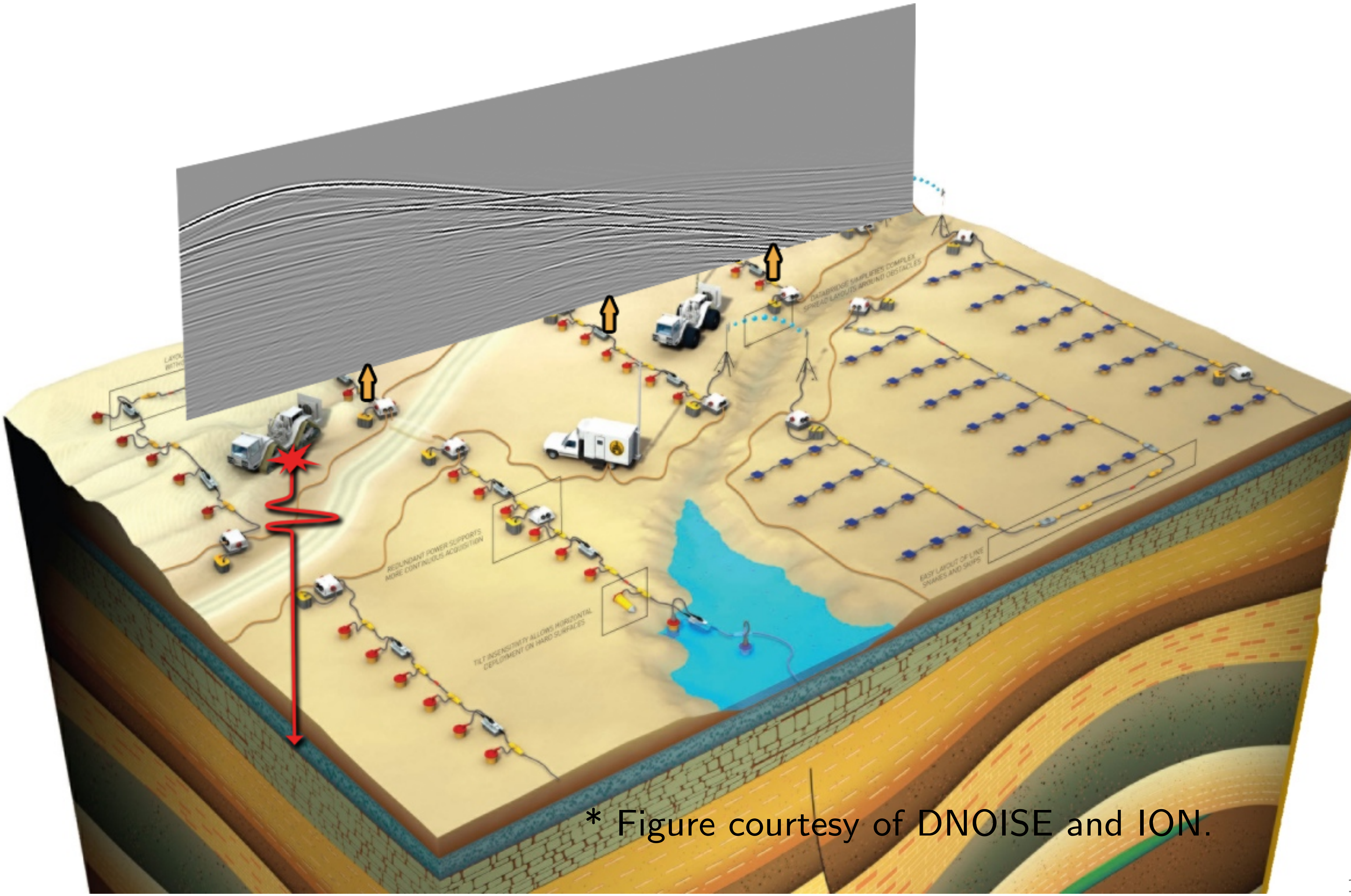
Recovery of Sparse Signals

- SNR averaged over 20 experiments for k-sparse signals x with $k = 40$, and $N = 500$.
- The noise free case:



Application to seismic trace interpolation

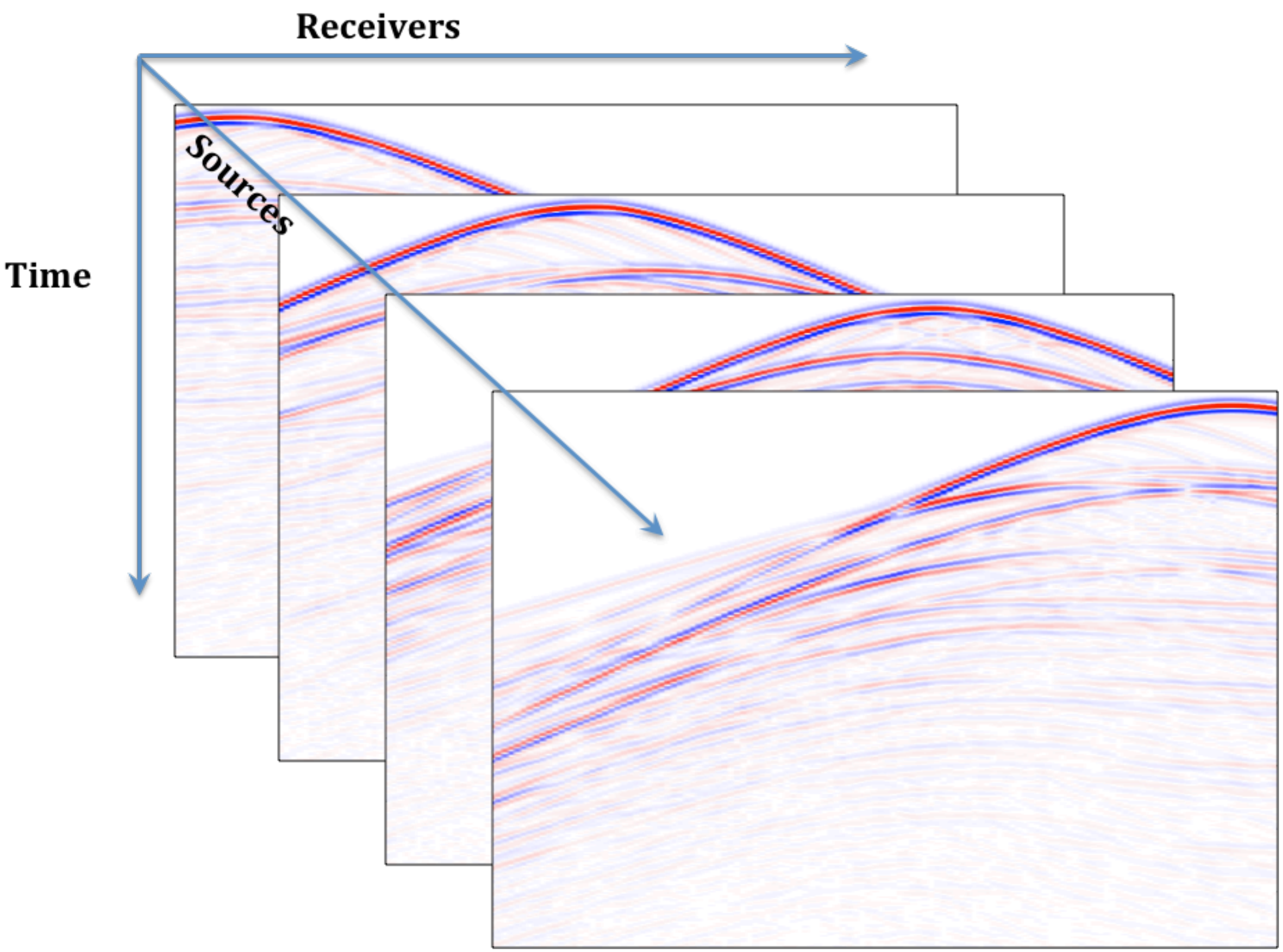
Seismic data acquisition



* Figure courtesy of DNOISE and ION.

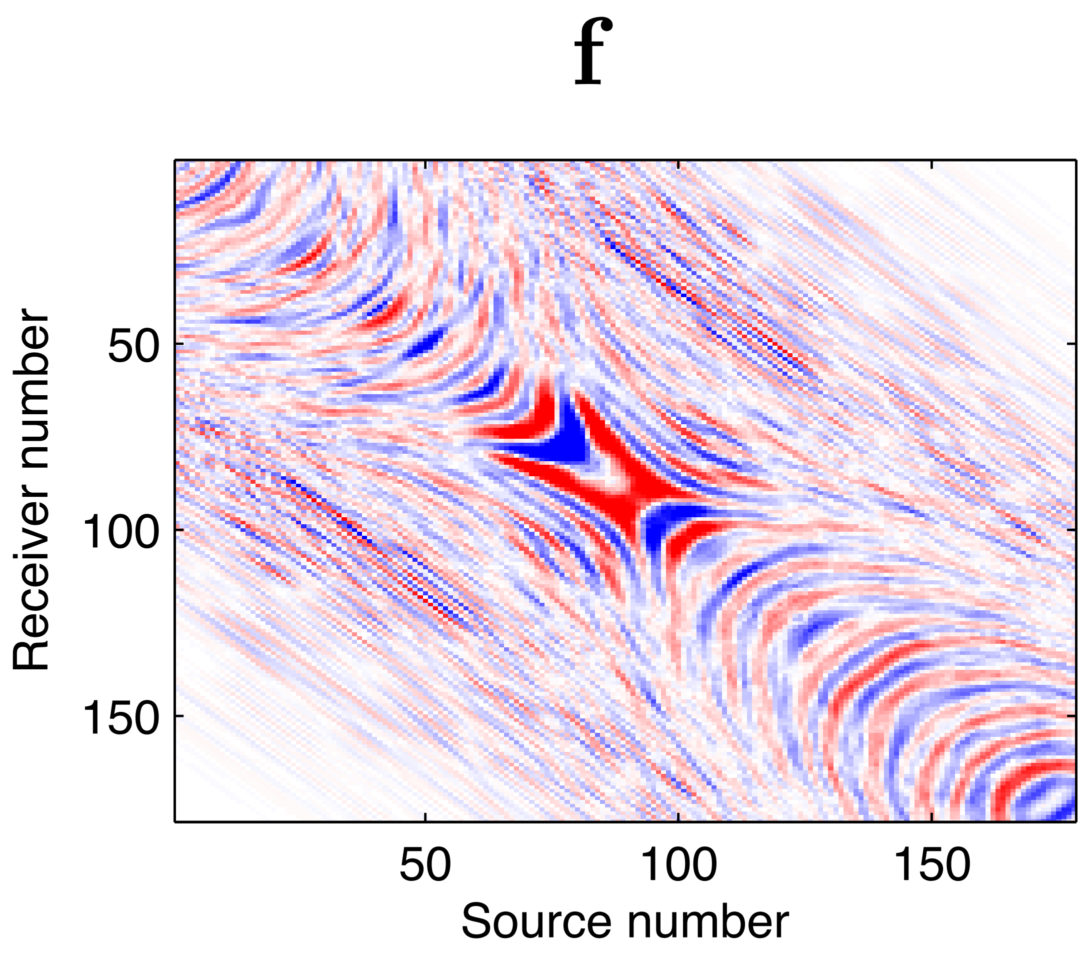
Randomized acquisition of seismic lines

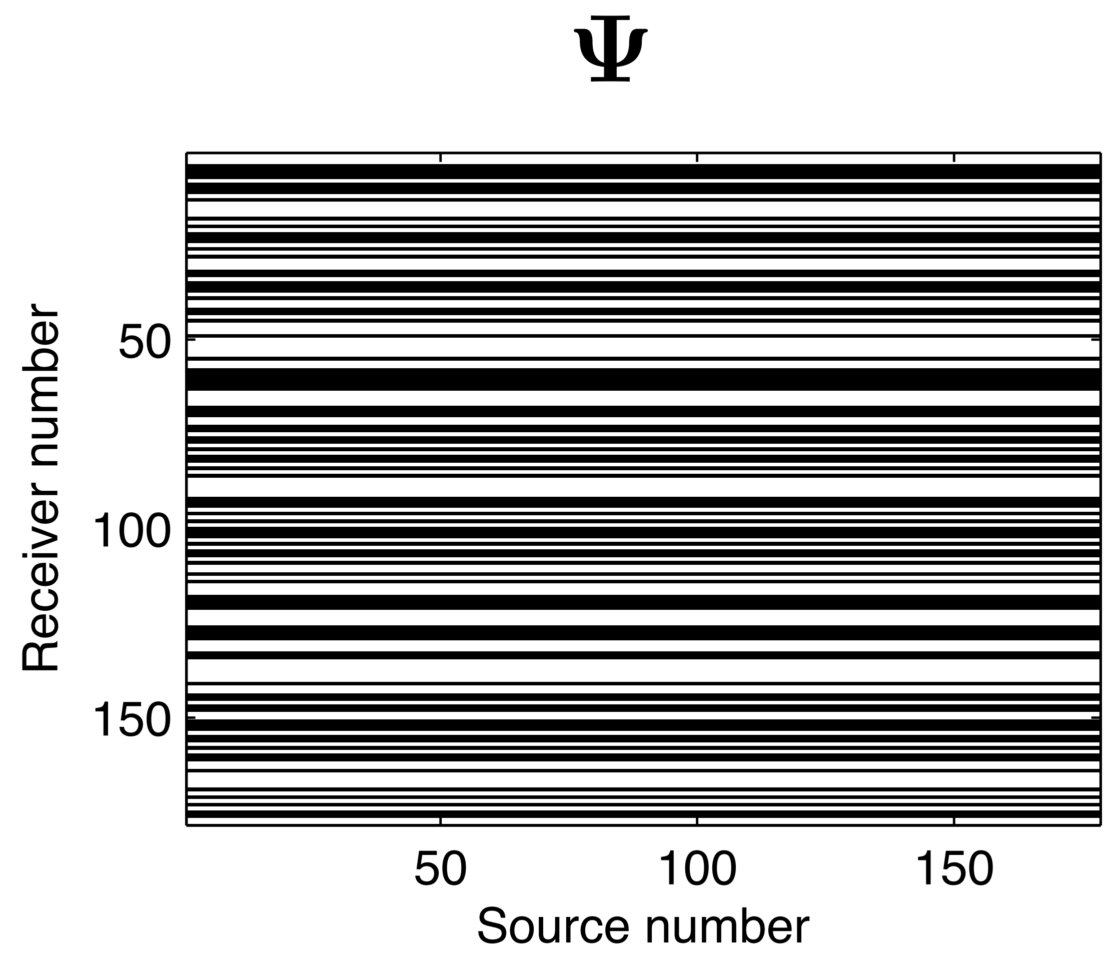
- Consider a seismic line with 178 sources, 178 receivers, and 500 time samples.



Randomized acquisition of seismic lines

- Consider a seismic line with 178 sources, 178 receivers, and 500 time samples.
- The receiver spread is randomly subsampled using the mask Ψ.

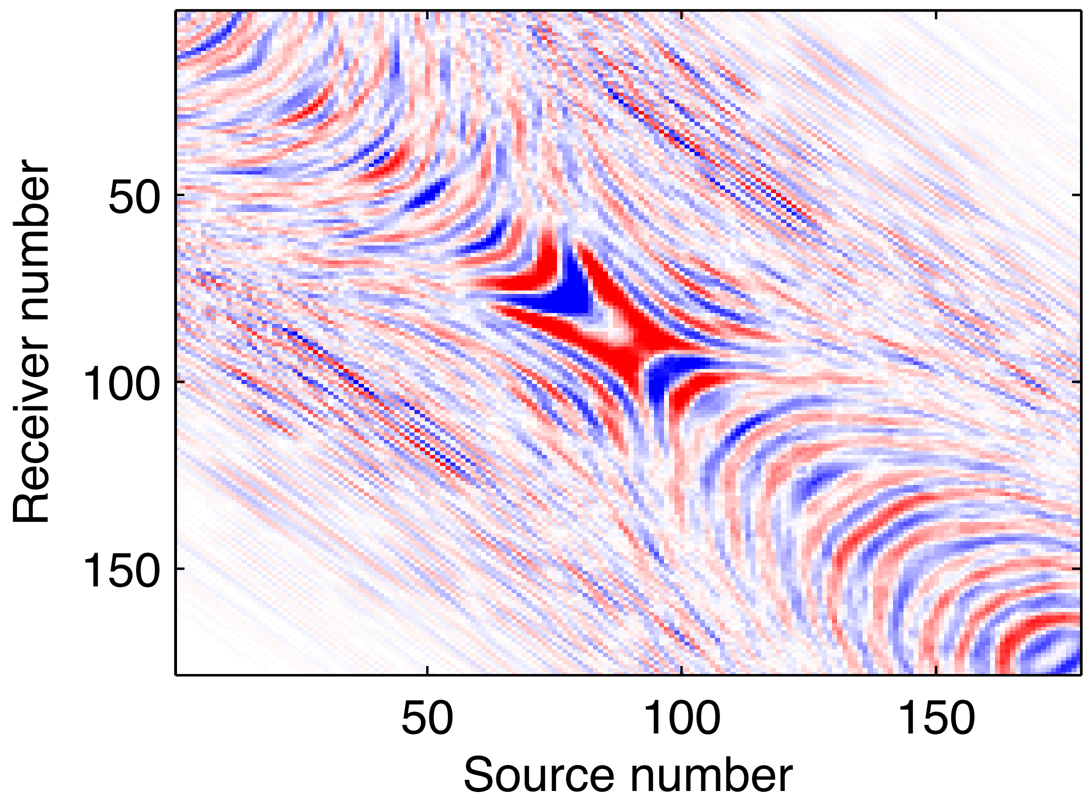




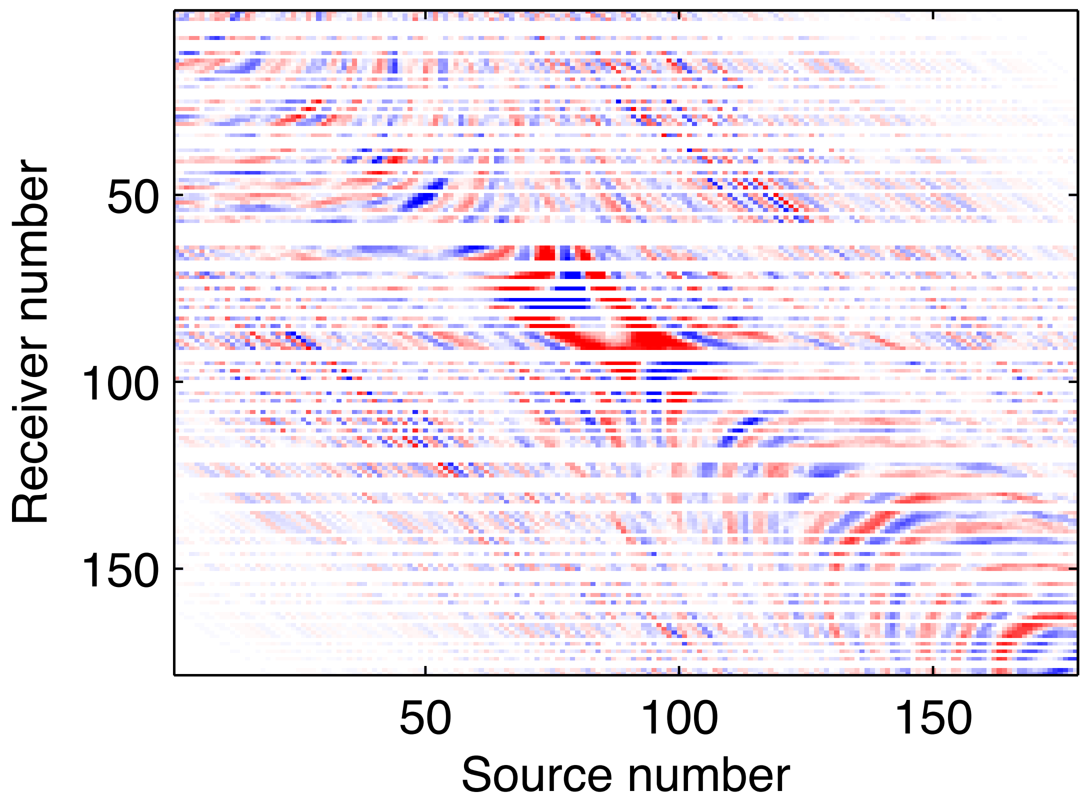
Randomized acquisition of seismic lines

- Consider a seismic line with 178 sources, 178 receivers, and 500 time samples.
- The receiver spread is randomly subsampled using the mask Ψ.

$\mathbf{f}$

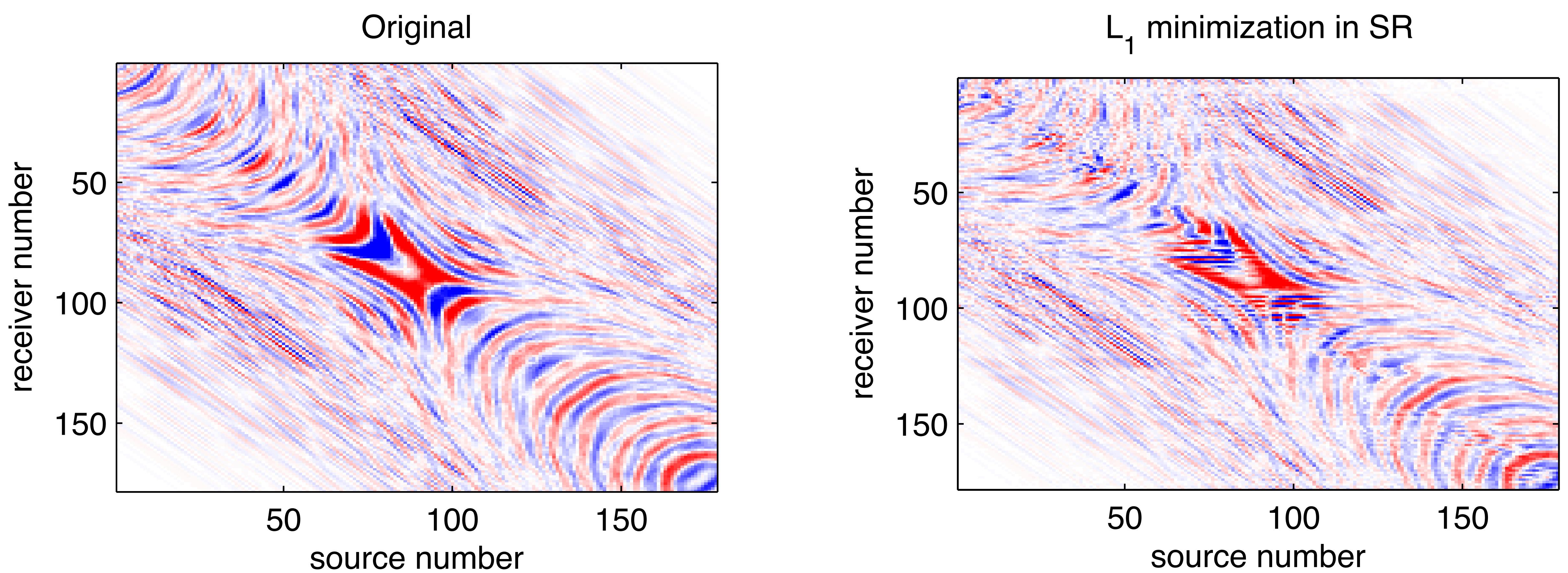


$\mathbf{y} = \mathbf{\Psi f}$



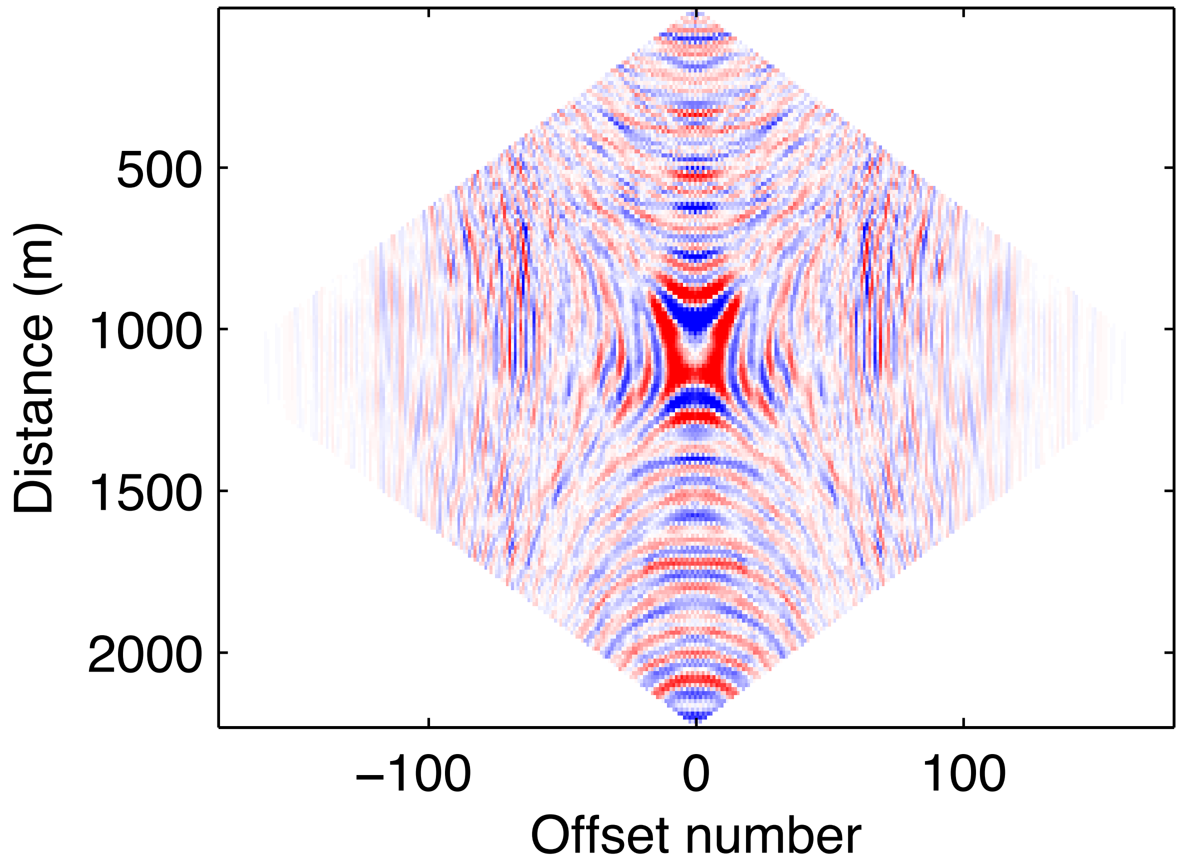
Randomized acquisition of seismic lines

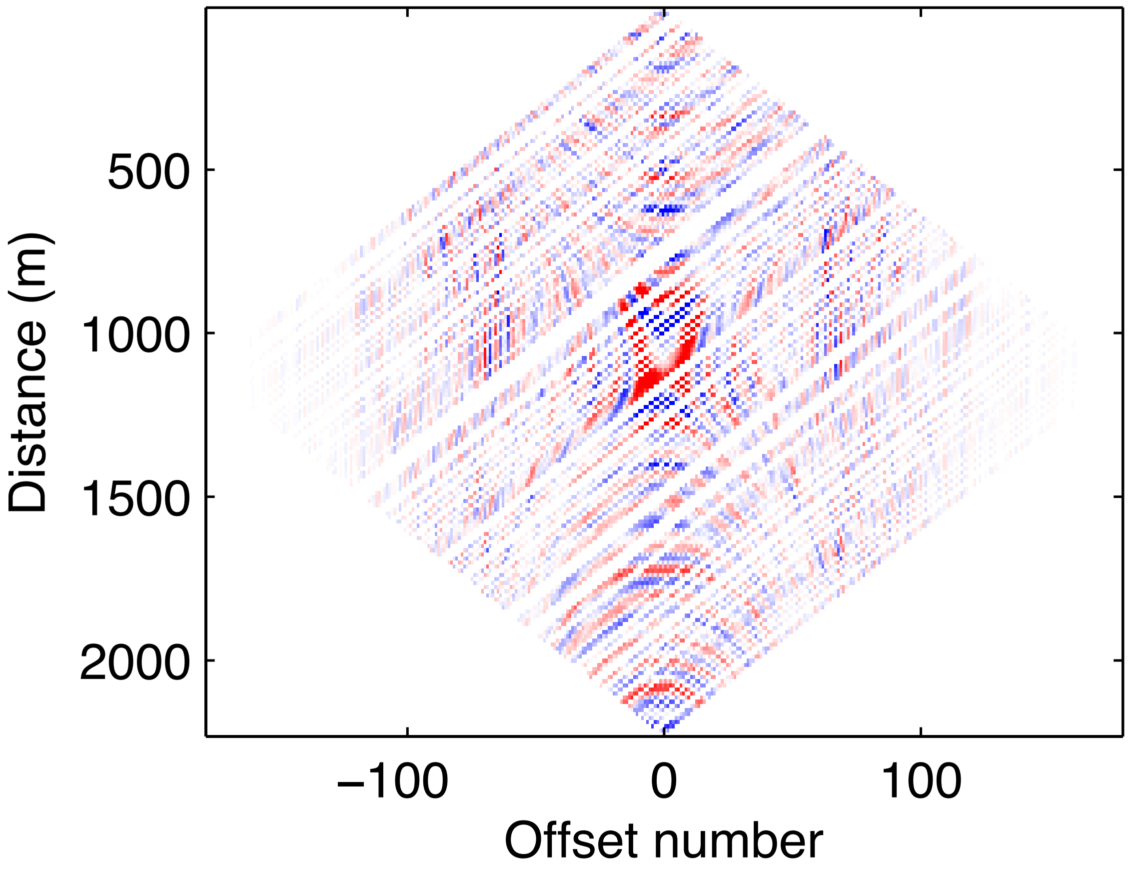
- Consider a seismic line with 178 sources, 178 receivers, and 500 time samples.
- Recovery using ℓ_1 minimization on frequency slices.



What more can be done?

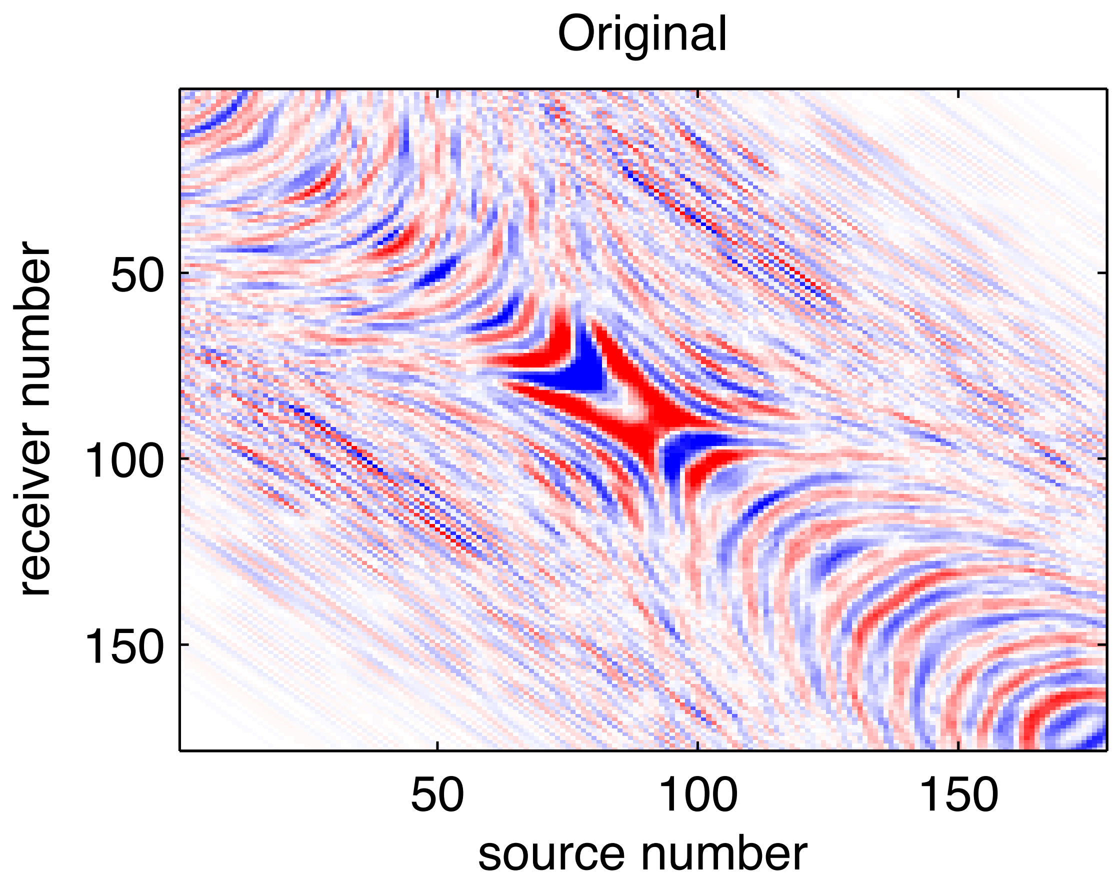
- Improve the RIP of $\mathbf{A} = \mathbf{\Psi D^H}$ by changing the interaction of $\mathbf{\Psi}$ and $\mathbf{D}^H$.
- E.g.: Perform recovery in the midpoint-offset domain.

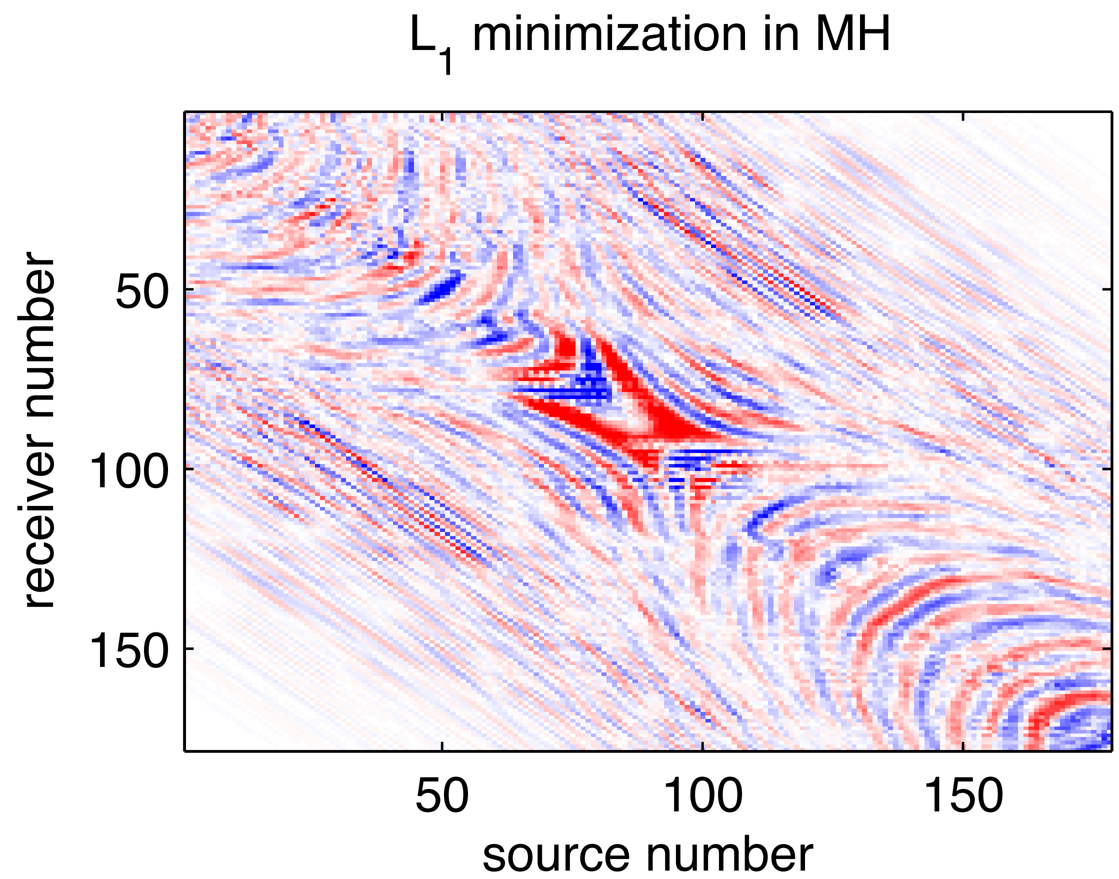




What more can be done?

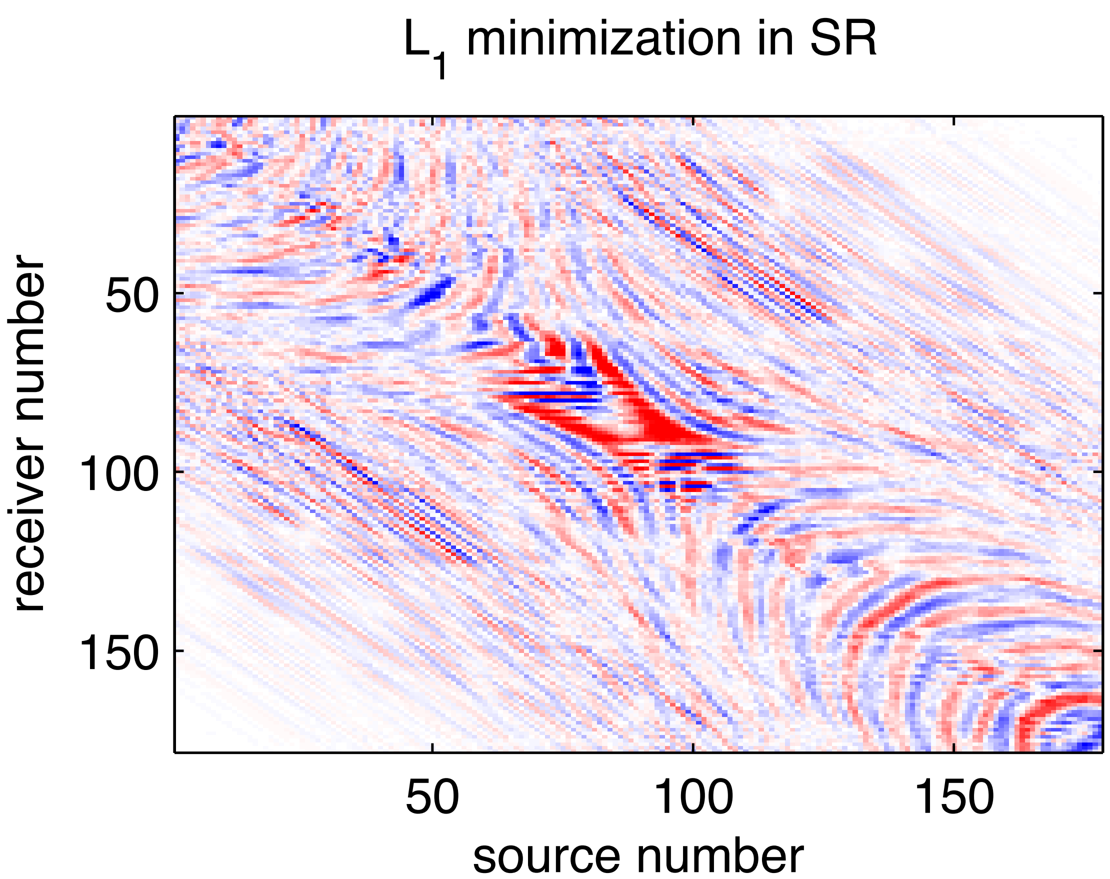
- Incorporate support information using weighted-ℓ_1 minimization.
- E.g.: Adjacent frequency slices and offset slices have highly correlated curvelet domain support sets.

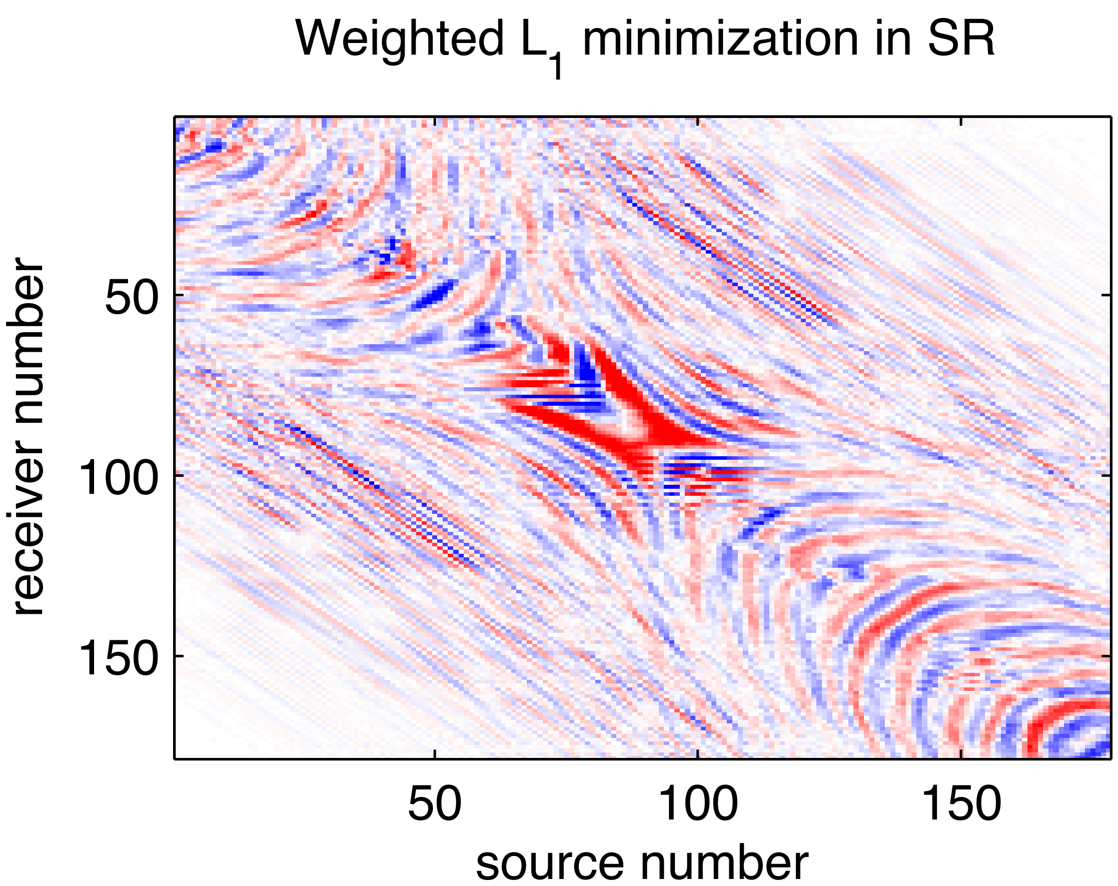




What more can be done?

- Incorporate support information using weighted-ℓ_1 minimization.
- E.g.: Adjacent frequency slices and offset slices have highly correlated curvelet domain support sets.

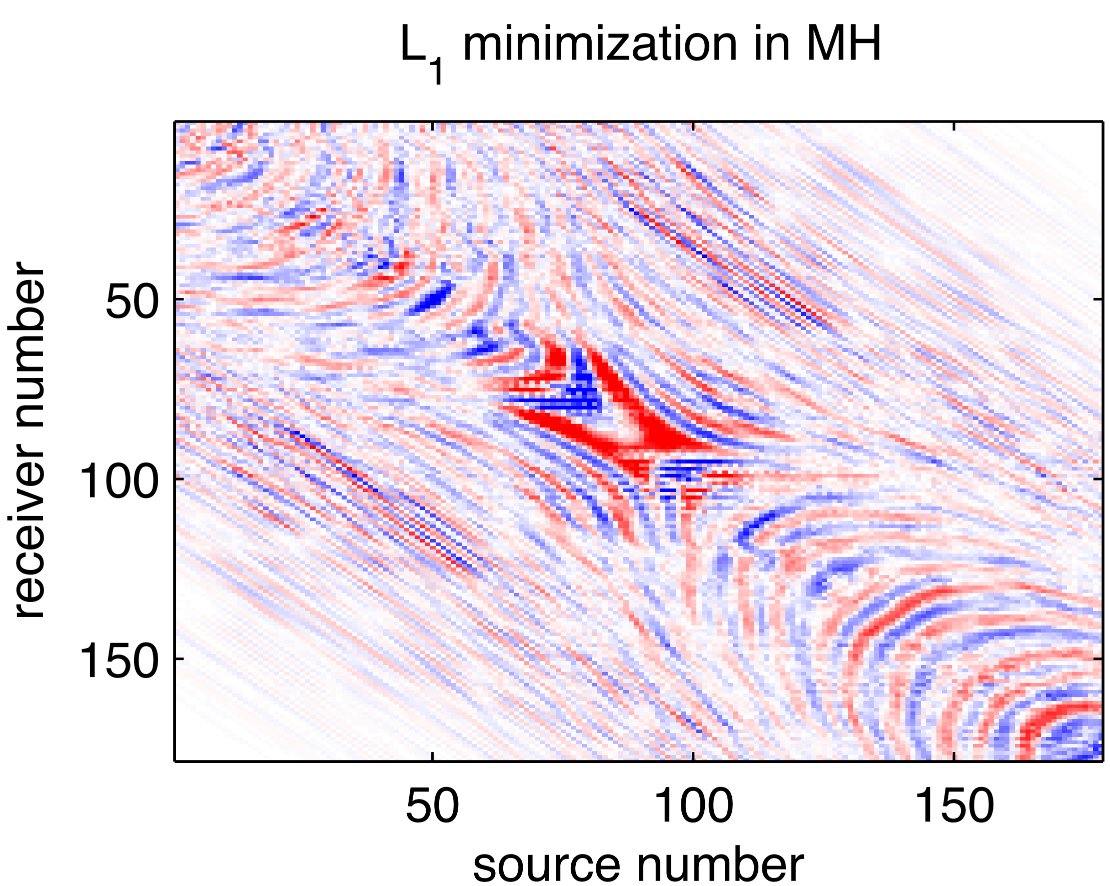


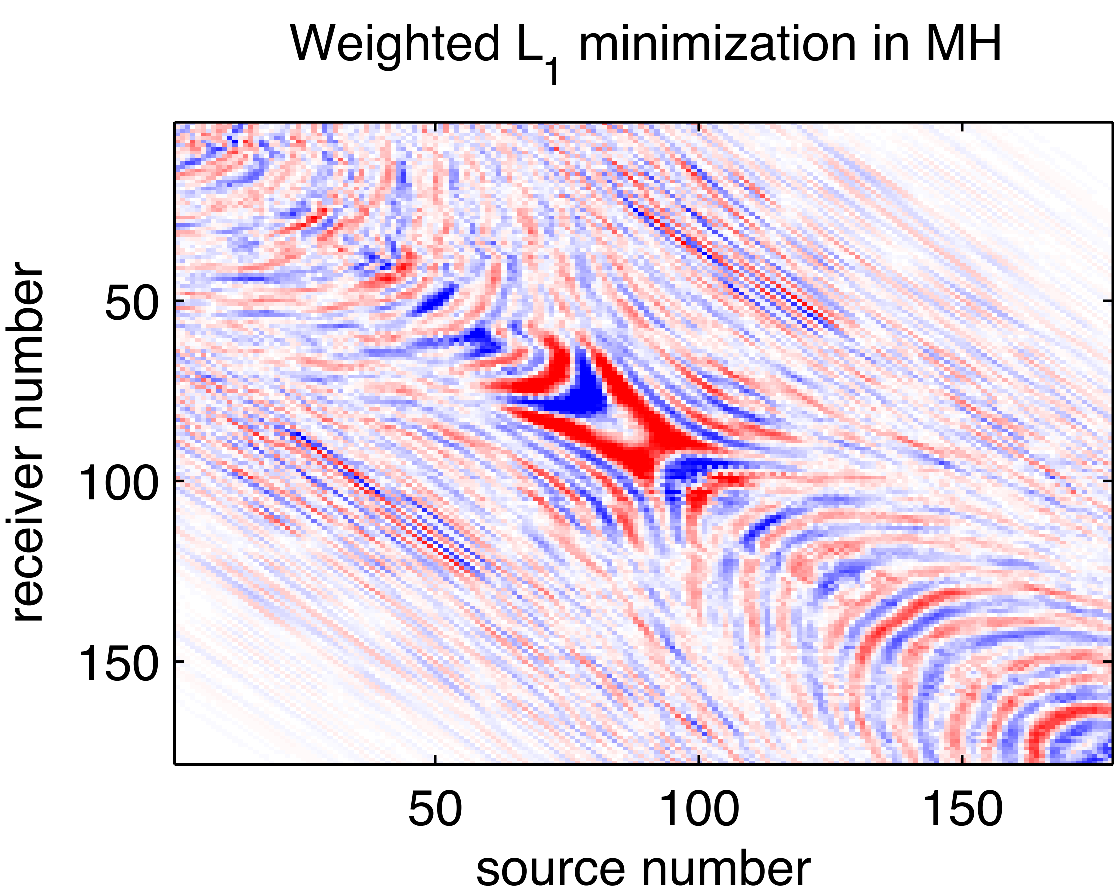


Monday, 3 December, 12

What more can be done?

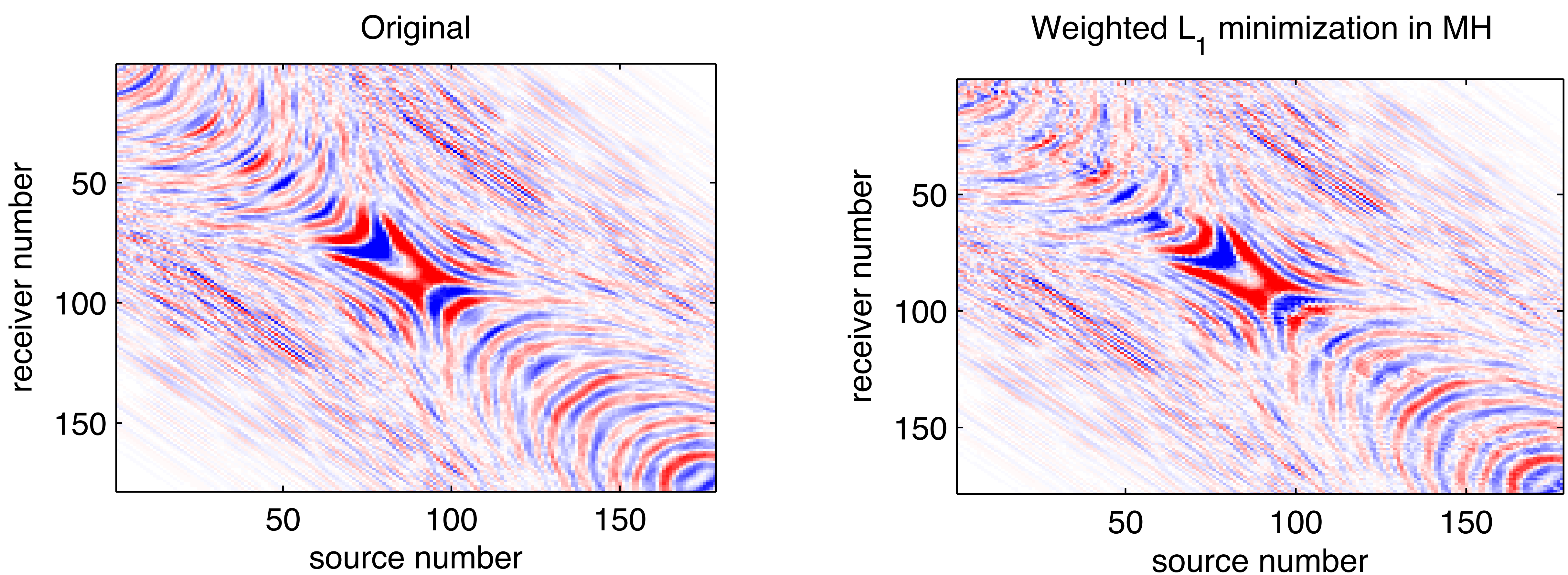
- Incorporate support information using weighted-ℓ_1 minimization.
- E.g.: Adjacent frequency slices and offset slices have highly correlated curvelet domain support sets.





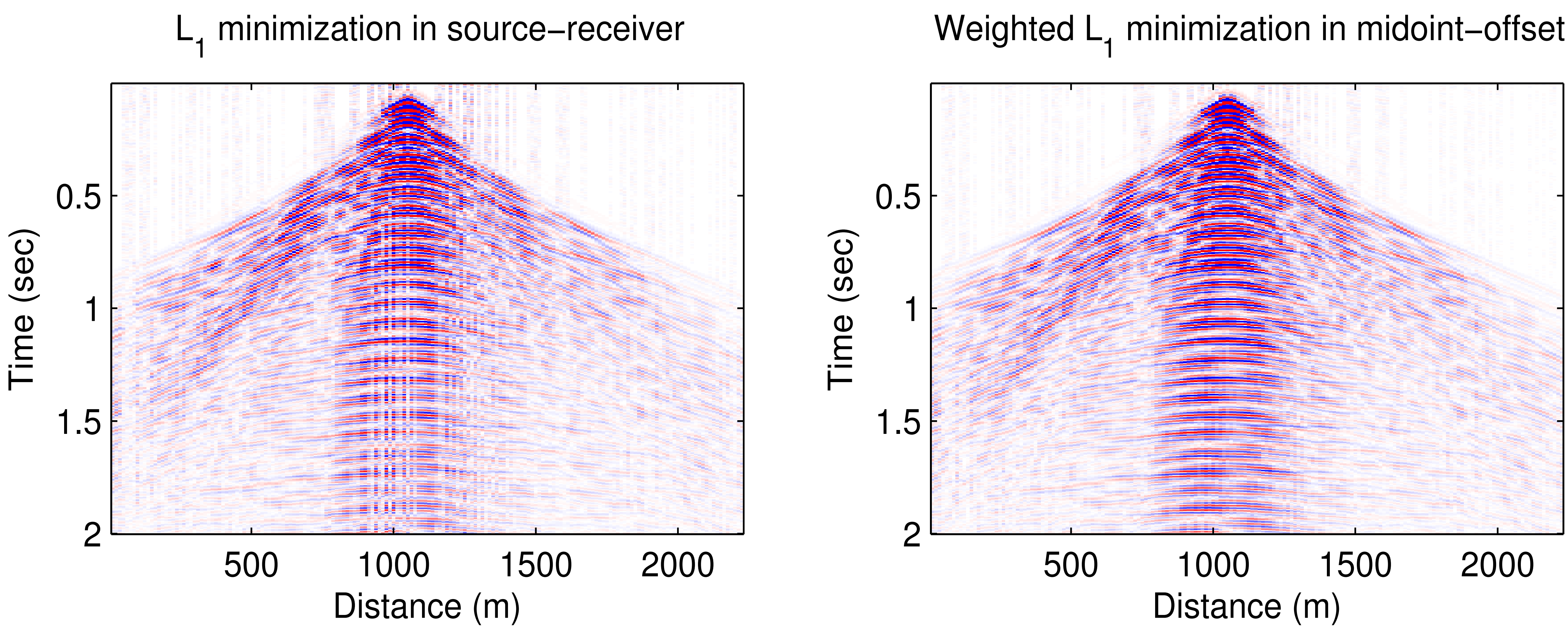
What more can be done?

- Incorporate support information using weighted-ℓ_1 minimization.
- E.g.: Adjacent frequency slices and offset slices have highly correlated curvelet domain support sets.



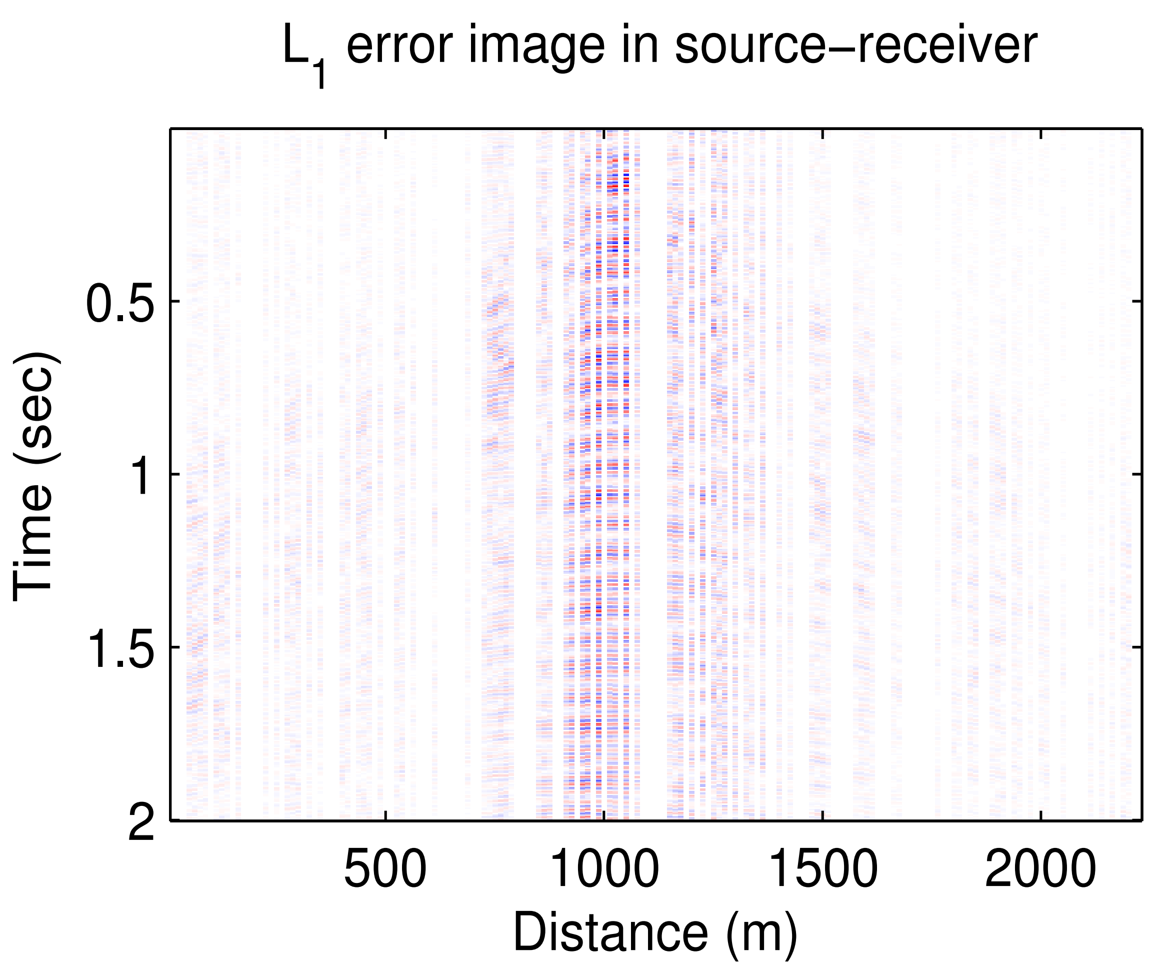
What more can be done?

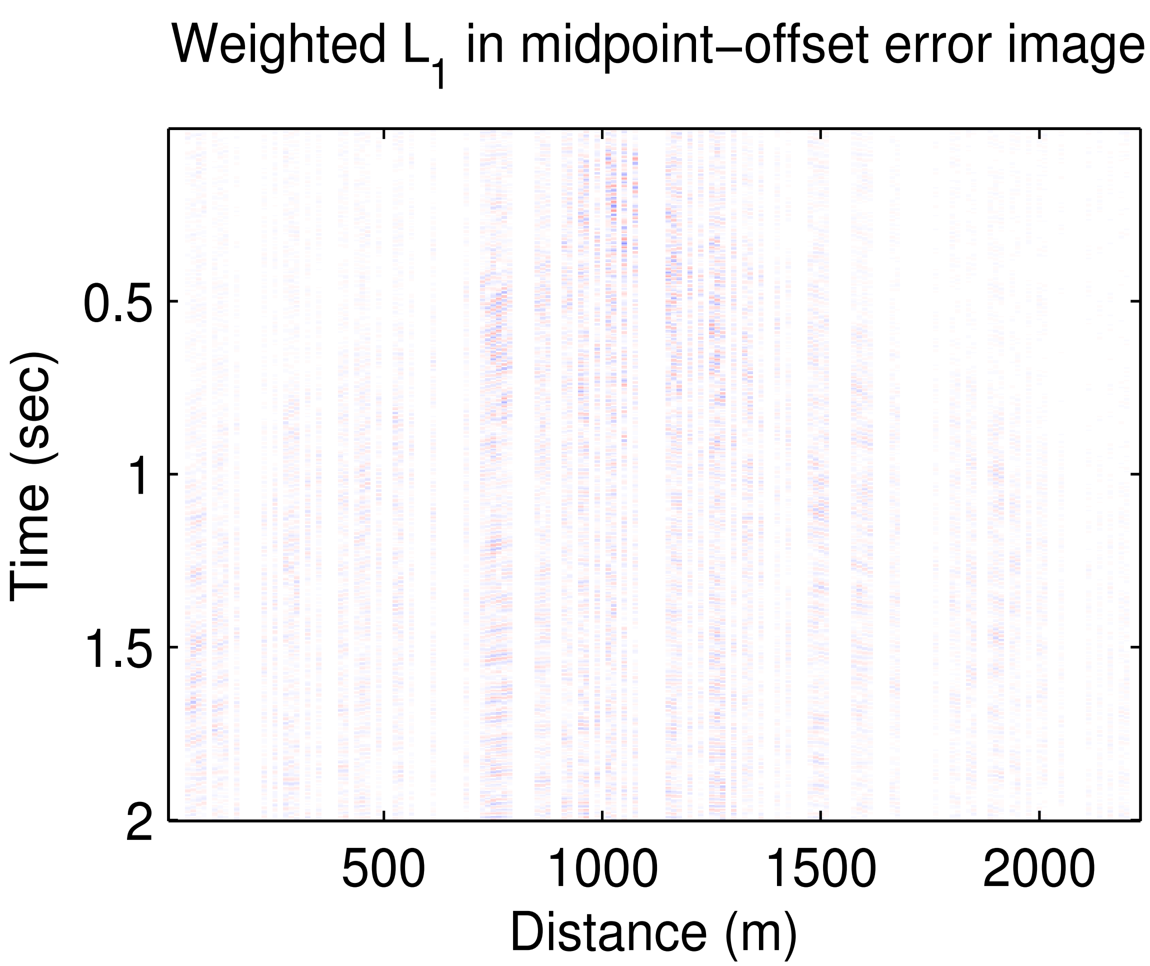
- Incorporate support information using weighted-ℓ_1 minimization.
- E.g.: Adjacent frequency slices and offset slices have highly correlated curvelet domain support sets.



What more can be done?

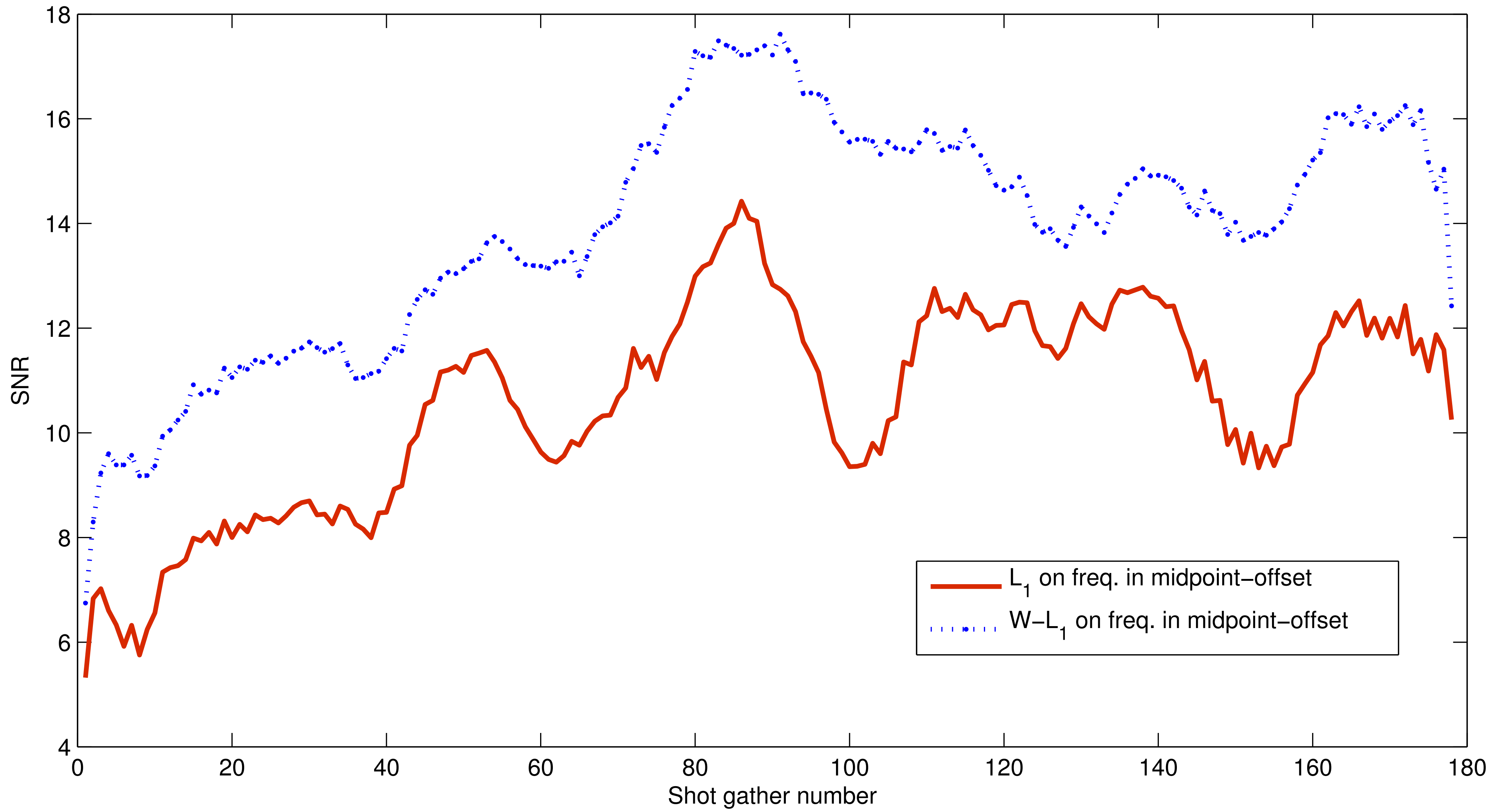
- Incorporate support information using weighted-ℓ_1 minimization.
- E.g.: Adjacent frequency slices and offset slices have highly correlated curvelet domain support sets.





Seismic recovery using weighted ℓ_1 minimization

(Mansour, Herrmann, Yilmaz '12)



Recap of weighted ℓ_1

- If a prior support estimate is available, then weighted ℓ_1 minimization guarantees better recovery when $\alpha > 0.5$.
  - Can we extend this analysis to multiple weighting sets?

    Yes! (Mansour, Yilmaz '11)
- What if we had no prior support estimate:
  - How would an iterative weighted ℓ_1 algorithm that incorporates support accuracy perform?

    The SDRL1 algorithm. (Mansour, Yilmaz '12) (CWB '08)
  - Is there a computationally efficient algorithm that achieves the gains of re-weighted ℓ_1?

    The WSPGL1 algorithm. (Mansour '12) (Asif and Romberg '12)

Recap of weighted ℓ_1

- If a prior support estimate is available, then weighted ℓ_1 minimization guarantees better recovery when $\alpha > 0.5$.
  - Can we extend this analysis to multiple weighting sets?

    Yes! (Mansour, Yilmaz '11)
- What if we had no prior support estimate:
  - How would an iterative weighted ℓ_1 algorithm that incorporates support accuracy perform?

    The SDRL1 algorithm. (Mansour, Yilmaz '12) (CWB '08)
  - Is there a computationally efficient algorithm that achieves the gains of re-weighted ℓ_1?

    The WSPGL1 algorithm. (Mansour '12) (Asif and Romberg '12)

Recap of weighted ℓ_1

- If a prior support estimate is available, then weighted ℓ_1 minimization guarantees better recovery when $\alpha > 0.5$.
  - Can we extend this analysis to multiple weighting sets?

    Yes! (Mansour, Yilmaz '11)
- What if we had no prior support estimate:
  - How would an iterative weighted ℓ_1 algorithm that incorporates support accuracy perform?

    The SDRL1 algorithm. (Mansour, Yilmaz '12) (CWB '08)
  - Is there a computationally efficient algorithm that achieves the gains of re-weighted ℓ_1?

    The WSPGL1 algorithm. (Mansour '12) (Asif and Romberg '12)

Recap of weighted ℓ_1

- If a prior support estimate is available, then weighted ℓ_1 minimization guarantees better recovery when $\alpha > 0.5$.
  - Can we extend this analysis to multiple weighting sets?
    Yes! (Mansour, Yilmaz '11)
- What if we had no prior support estimate:
  - How would an iterative weighted ℓ_1 algorithm that incorporates support accuracy perform?
    The SDRL1 algorithm. (Mansour, Yilmaz '12) (CWB '08)
  - Is there a computationally efficient algorithm that achieves the gains of re-weighted ℓ_1?
    The WSPGL1 algorithm. (Mansour '12) (Asif and Romberg '12)

Part 1: Compressed sensing and sparse recovery

Part 2: Weighted ℓ_1 minimization

Part 3: ℓ_1 solvers and the WSPGL1 algorithm

Part 4: Sparse randomized Kaczmarz

A BPDN solver

- van den Berg and Friedlander '08 developed the *Spectral Projected Gradient for ℓ_1 minimization* (SPGL1) algorithm.
  - Given $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, want to solve the ℓ_1 problem

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{u}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \leq \epsilon$$

  - If $\tau^* = \|\mathbf{x}\|_1$ is known, then $\mathbf{x}^*$ can be found by solving the following LASSO problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau^*$$

  - SPGL1 develops an efficient framework for finding the correct τ^*.

A BPDN solver

- van den Berg and Friedlander '08 developed the *Spectral Projected Gradient for ℓ_1 minimization* (SPGL1) algorithm.
  - Given $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, want to solve the ℓ_1 problem

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{u}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \leq \epsilon$$

  - If $\tau^* = \|\mathbf{x}\|_1$ is known, then $\mathbf{x}^*$ can be found by solving the following LASSO problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau^*$$

  - SPGL1 develops an efficient framework for finding the correct τ^*.

A BPDN solver

- van den Berg and Friedlander '08 developed the *Spectral Projected Gradient for ℓ_1 minimization* (SPGL1) algorithm.
  - Given $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, want to solve the ℓ_1 problem

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{u}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \leq \epsilon$$

  - If $\tau^* = \|\mathbf{x}\|_1$ is known, then $\mathbf{x}^*$ can be found by solving the following LASSO problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau^*$$

  - SPGL1 develops an efficient framework for finding the correct τ^*.

A BPDN solver

- van den Berg and Friedlander '08 developed the *Spectral Projected Gradient for ℓ_1 minimization* (SPGL1) algorithm.
  - Given $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, want to solve the ℓ_1 problem

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{u}\|_1 \text{ subject to } \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \leq \epsilon$$

  - If $\tau^* = \|\mathbf{x}\|_1$ is known, then $\mathbf{x}^*$ can be found by solving the following LASSO problem:

$$\mathbf{x}^* = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau^*$$

  - SPGL1 develops an efficient framework for finding the correct τ^*.

The SPGL1 algorithm (van den Berg, Friedlander '08)

- Solves a sequence of LASSO subproblems (LS$_\tau$)

$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \le \tau_t$$

- Initialize the algorithm at a point $\mathbf{x}^{(0)}$ giving an initial $\tau_0 = \|\mathbf{x}^{(0)}\|_1$.
- Update τ by traversing the Pareto curve defined by the function $\phi(\tau) = \|\mathbf{y} - \mathbf{A}\mathbf{x}^{\tau_t}\|_2$.

$$\tau_{t+1} = \tau_t + \frac{\phi(\tau_t) - \epsilon}{\phi'(\tau_t)},$$

- Stop when $\|\mathbf{y} - \mathbf{A}\mathbf{x}^{\tau_t}\|_2 = \epsilon$.

The SPGL1 algorithm (van den Berg, Friedlander '08)

- Solves a sequence of LASSO subproblems (LS_τ)

$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau_t$$

- Initialize the algorithm at a point $\mathbf{x}^{(0)}$ giving an initial $\tau_0 = \|\mathbf{x}^{(0)}\|_1$.
- Update τ by traversing the Pareto curve defined by the function $\phi(\tau) = \|\mathbf{y} - \mathbf{A}\mathbf{x}^{\tau_t}\|_2$.

$$\tau_{t+1} = \tau_t + \frac{\phi(\tau_t) - \epsilon}{\phi'(\tau_t)},$$

- Stop when $\|\mathbf{y} - \mathbf{A}\mathbf{x}^{\tau_t}\|_2 = \epsilon$.

Monday, 3 December, 12

The SPGL1 algorithm (van den Berg, Friedlander '08)

- Solves a sequence of LASSO subproblems (LS_τ)

$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u}-\mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau_t$$

- Initialize the algorithm at a point $\mathbf{x}^{(0)}$ giving an initial $\tau_0 = \|\mathbf{x}^{(0)}\|_1$.
- Update τ by traversing the Pareto curve defined by the function $\phi(\tau) = \|\mathbf{y}-\mathbf{A}\mathbf{x}^{\tau_t}\|_2$.

$$\tau_{t+1} = \tau_t + \frac{\phi(\tau_t)-\epsilon}{\phi'(\tau_t)},$$

- Stop when $\|\mathbf{y}-\mathbf{A}\mathbf{x}^{\tau_t}\|_2 = \epsilon$.

The SPGL1 algorithm (van den Berg, Friedlander '08)

- Solves a sequence of LASSO subproblems (LS_τ)

$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{Au} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_1 \leq \tau_t$$

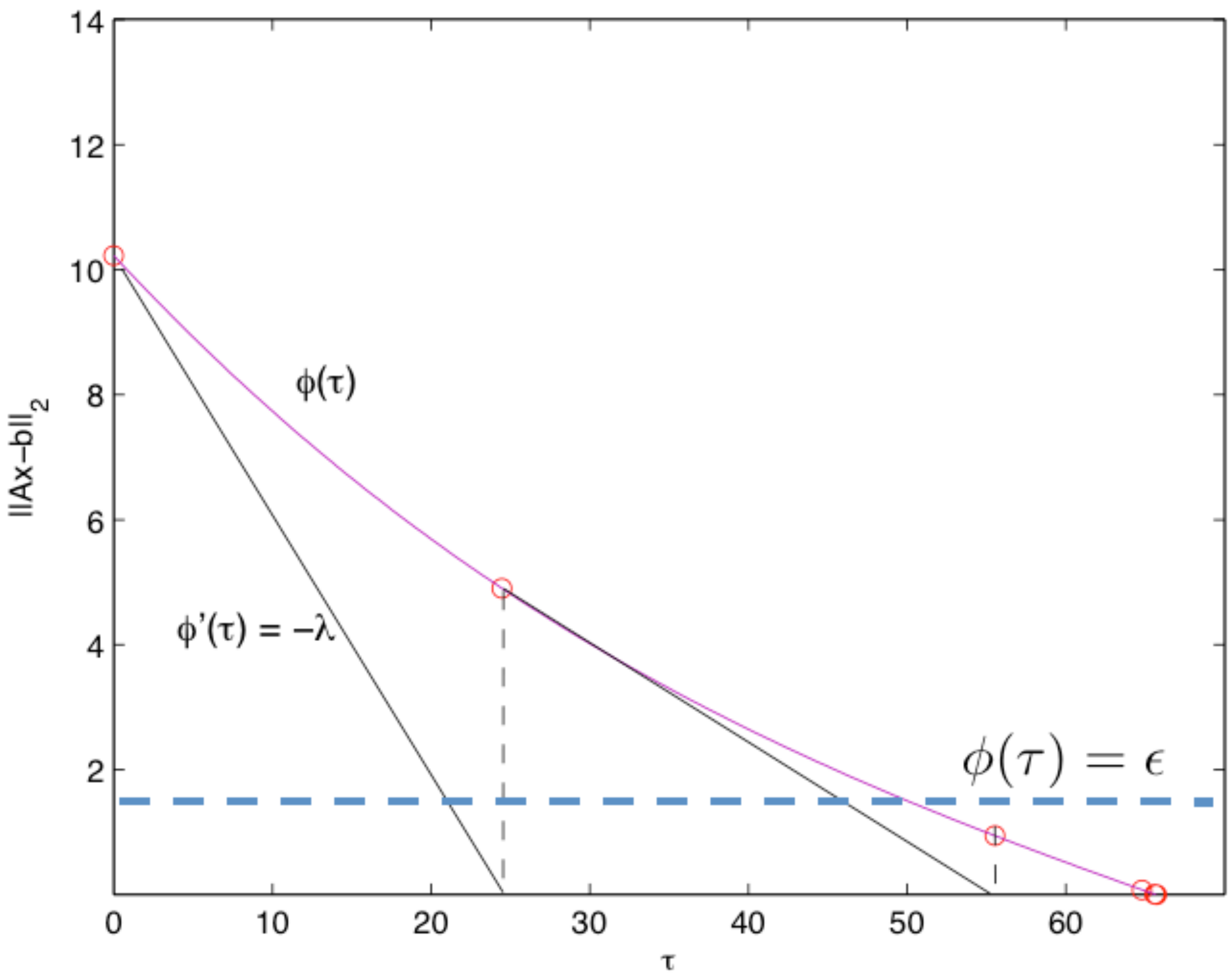
- Initialize the algorithm at a point $\mathbf{x}^{(0)}$ giving an initial $\tau_0 = \|\mathbf{x}^{(0)}\|_1$.
- Update τ by traversing the Pareto curve defined by the function $\phi(\tau) = \|\mathbf{y} - \mathbf{Ax}^{\tau_t}\|_2$.

$$\tau_{t+1} = \tau_t + \frac{\phi(\tau_t) - \epsilon}{\phi'(\tau_t)},$$

- Stop when $\|\mathbf{y} - \mathbf{Ax}^{\tau_t}\|_2 = \epsilon$.

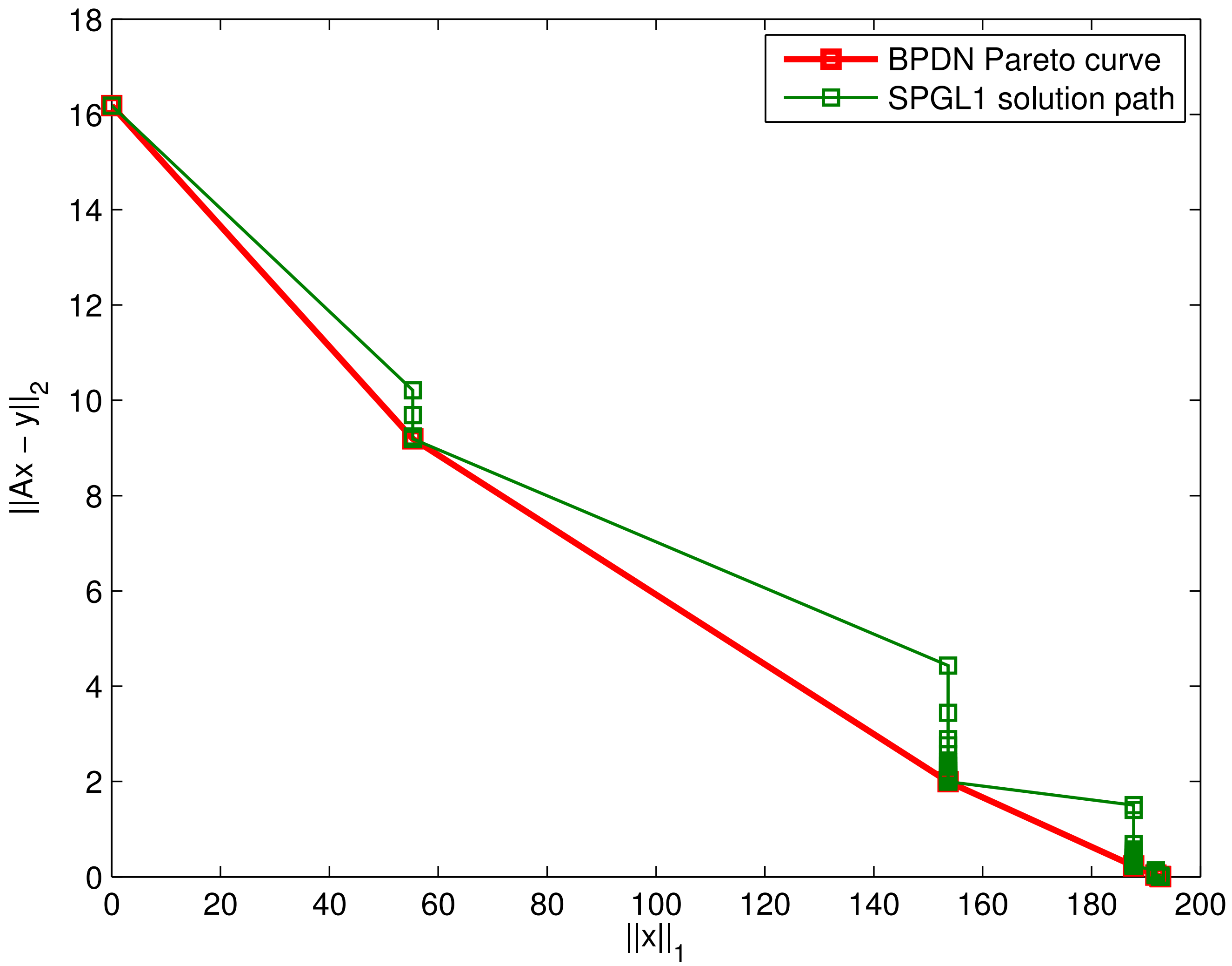
Traversing the Pareto curve

- Traces the optimal tradeoff between $\|\mathbf{y} - \mathbf{A}\mathbf{x}^\tau\|_2$ and $\|\mathbf{x}^\tau\|_1$.
- The solution to the ℓ_1 problem is found at $\phi(\tau) = \epsilon$.



Traversing the Pareto curve

- Traces the optimal tradeoff between $\|\mathbf{y} - \mathbf{A}\mathbf{x}^\tau\|_2$ and $\|\mathbf{x}^\tau\|_1$.
- The solution to the ℓ_1 problem is found at $\phi(\tau) = \epsilon$.



The WSPGL1 algorithm (Mansour '12)

- What if we incorporate support information in the LASSO subproblems?

- Solve a sequence of weighted LASSO subproblems.

$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u}-\mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_{1,\mathrm{w}} \le \tau_t$$

- Update the weight vector based on the solution of the previous subproblem.

$$\mathrm{w}_i = \begin{cases} \omega, & i \in \widetilde{T} \\ 1, & i \in \widetilde{T}^c \end{cases}, \quad \text{where} \quad \widetilde{T} = \mathrm{supp}(\mathbf{x}^{t-1}|_k).$$

Monday, 3 December, 12

The WSPGL1 algorithm (Mansour '12)

- What if we incorporate support information in the LASSO subproblems?
- Solve a sequence of weighted LASSO subproblems.

$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u}-\mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_{1,\mathrm{w}} \leq \tau_t$$

- Update the weight vector based on the solution of the previous subproblem.

$$\mathrm{w}_i = \begin{cases} \omega, & i \in \widetilde{T} \\ 1, & i \in \widetilde{T}^c \end{cases}, \quad \text{where} \quad \widetilde{T} = \mathrm{supp}(\mathbf{x}^{t-1}|_k).$$

The WSPGL1 algorithm (Mansour '12)

- What if we incorporate support information in the LASSO subproblems?

- Solve a sequence of weighted LASSO subproblems.

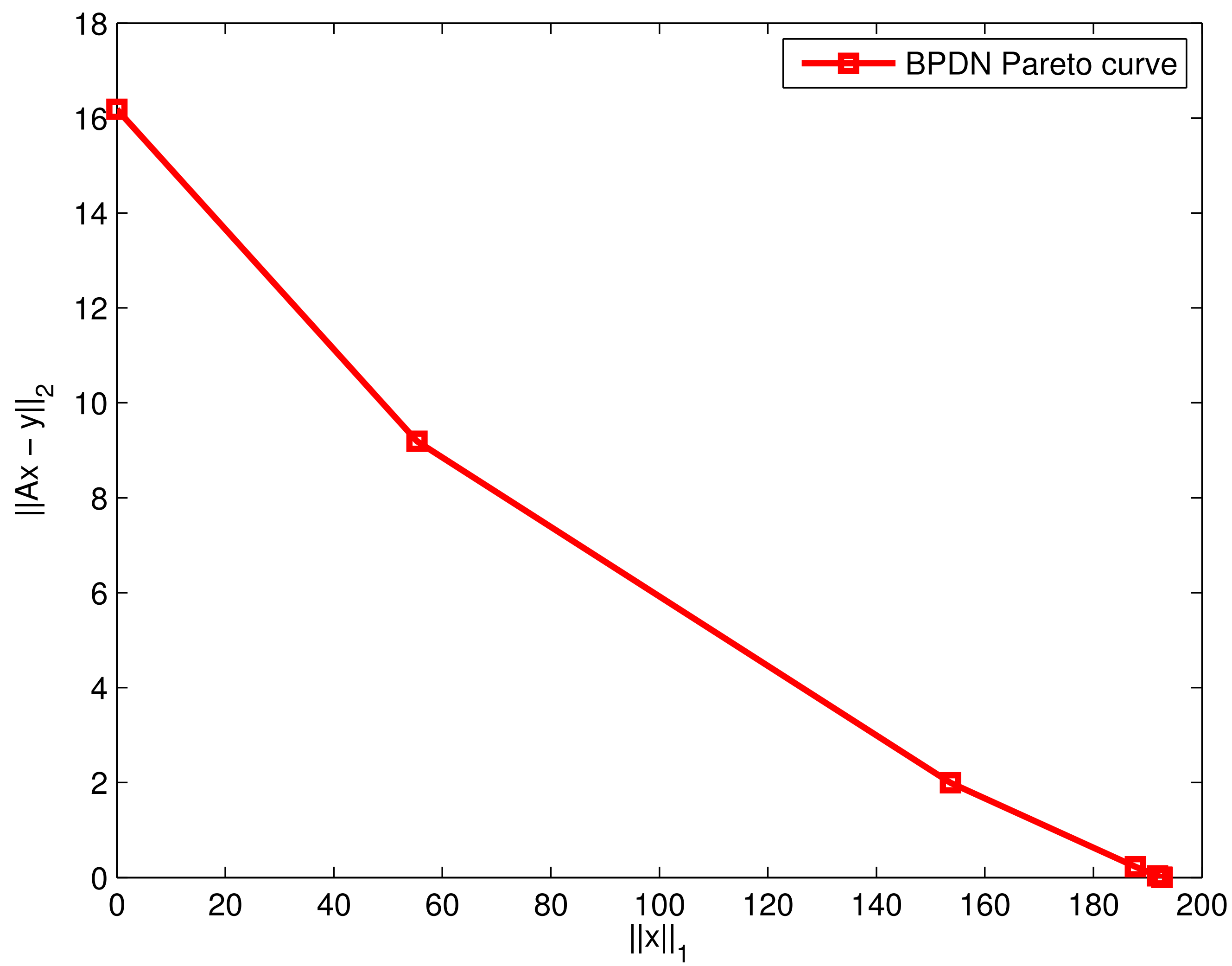
$$\mathbf{x}^{\tau_t} = \arg\min_{\mathbf{u}\in\mathbb{R}^N} \|\mathbf{A}\mathbf{u} - \mathbf{y}\|_2 \text{ subject to } \|\mathbf{u}\|_{1,\mathrm{w}} \leq \tau_t$$

- Update the weight vector based on the solution of the previous subproblem.

$$\mathrm{w}_i = \begin{cases} \omega, & i \in \widetilde{T} \\ 1, & i \in \widetilde{T}^c \end{cases}, \quad \text{where} \quad \widetilde{T} = \operatorname{supp}(\mathbf{x}^{t-1}|_k).$$

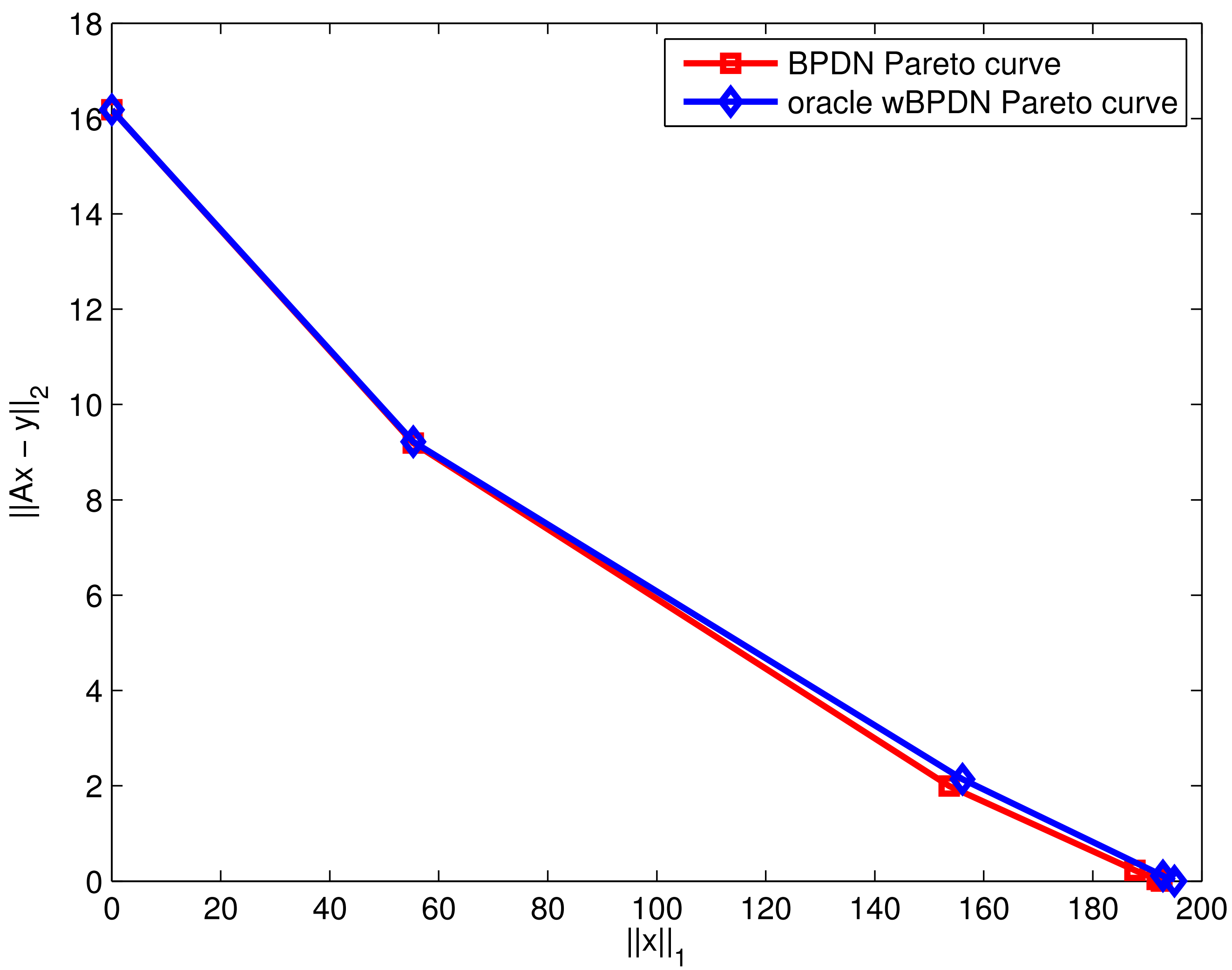
WSPGL1 and the Pareto curve

- The Pareto curve changes with the definition of every new weighted LASSO subproblem.



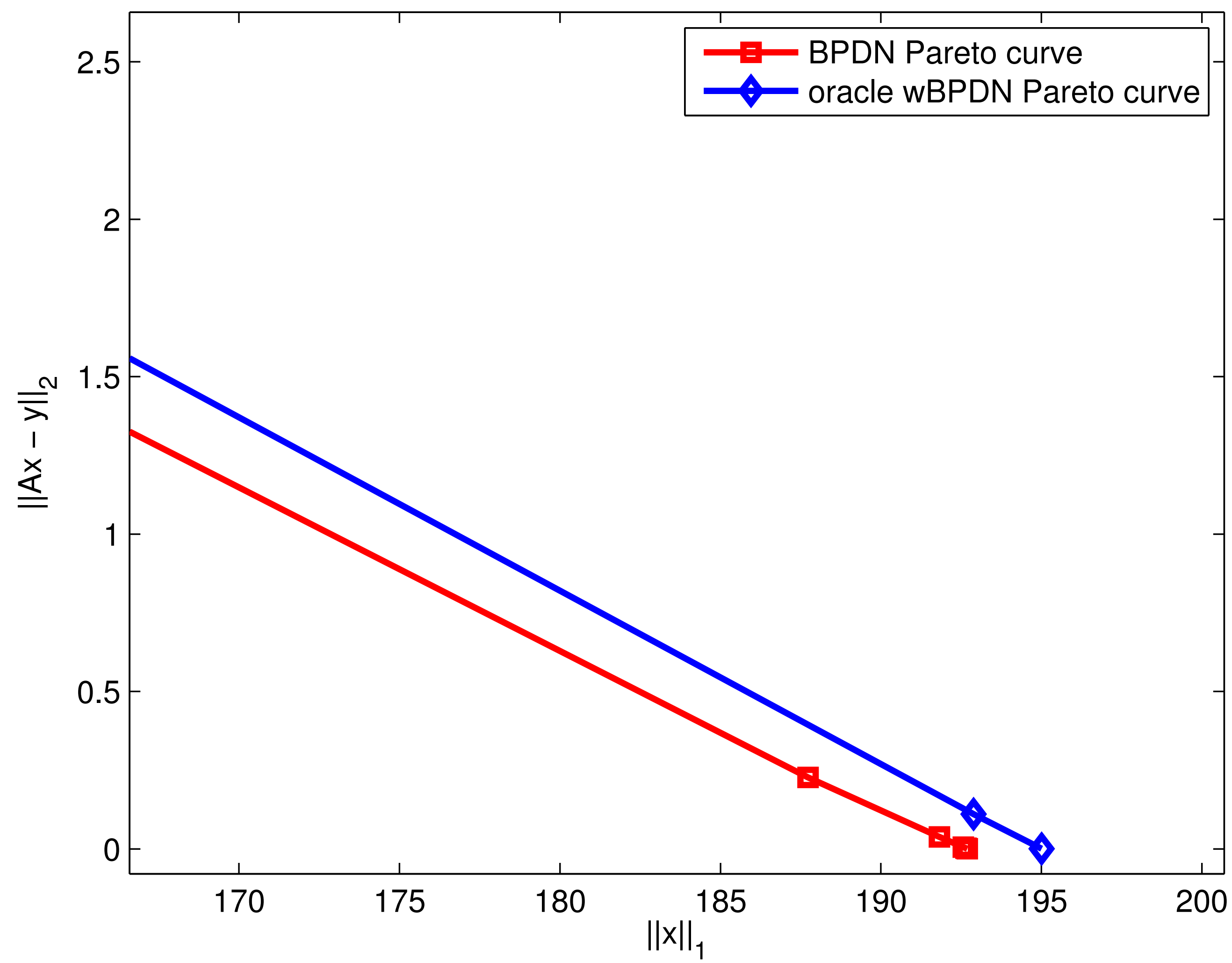
WSPGL1 and the Pareto curve

- The Pareto curve changes with the definition of every new weighted LASSO subproblem.



WSPGL1 and the Pareto curve

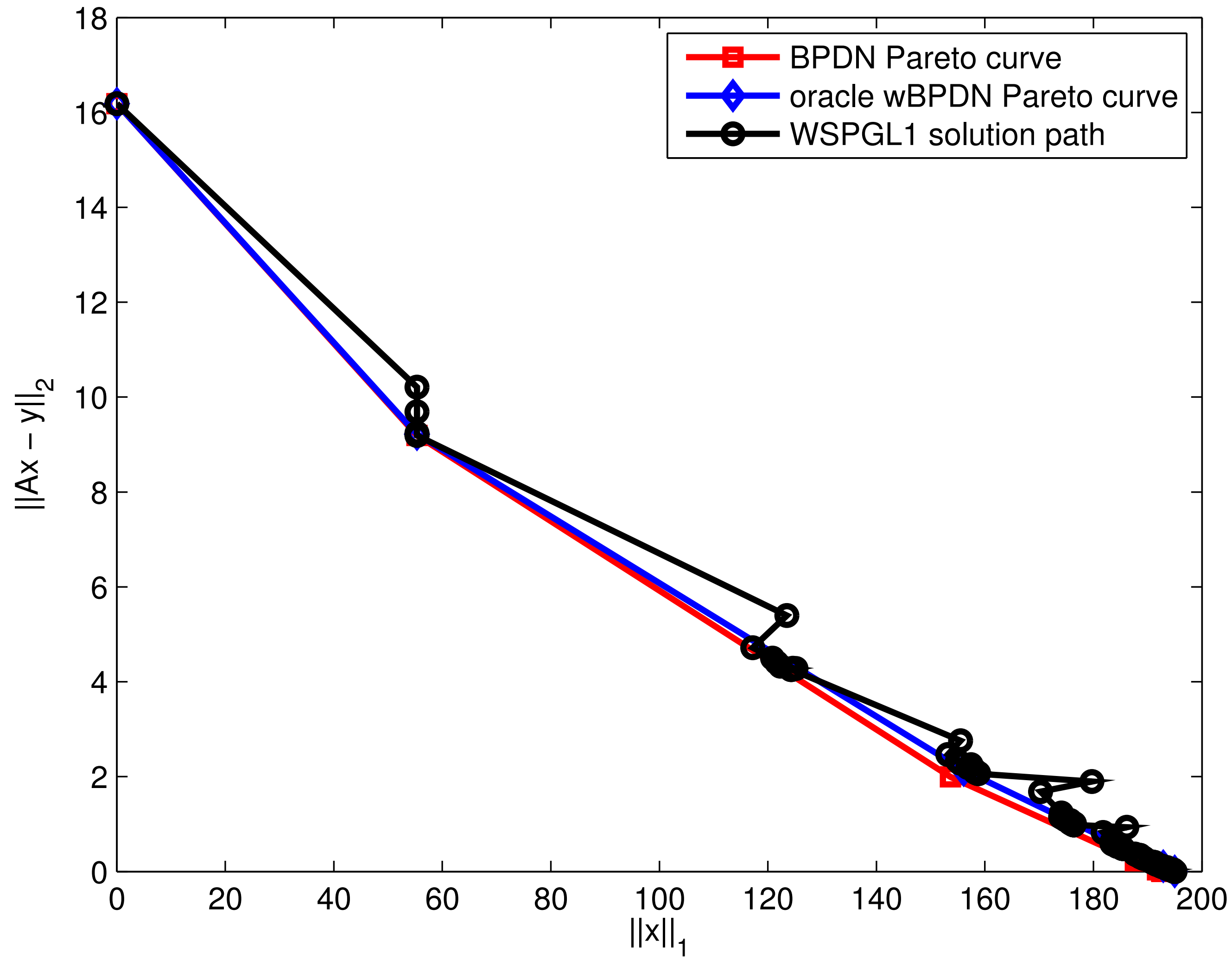
- The Pareto curve changes with the definition of every new weighted LASSO subproblem.



Monday, 3 December, 12

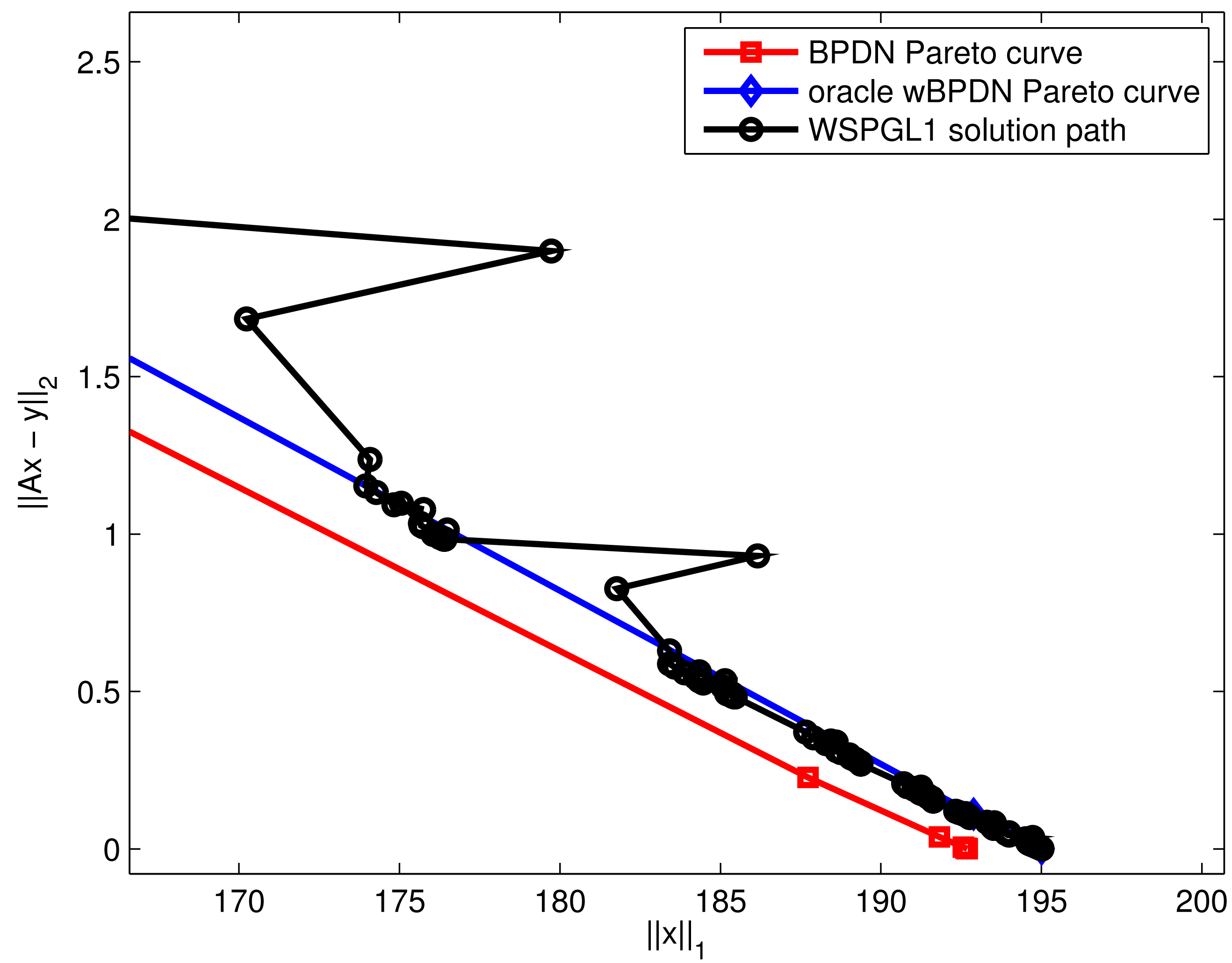
WSPGL1 and the Pareto curve

- The Pareto curve changes with the definition of every new weighted LASSO subproblem.



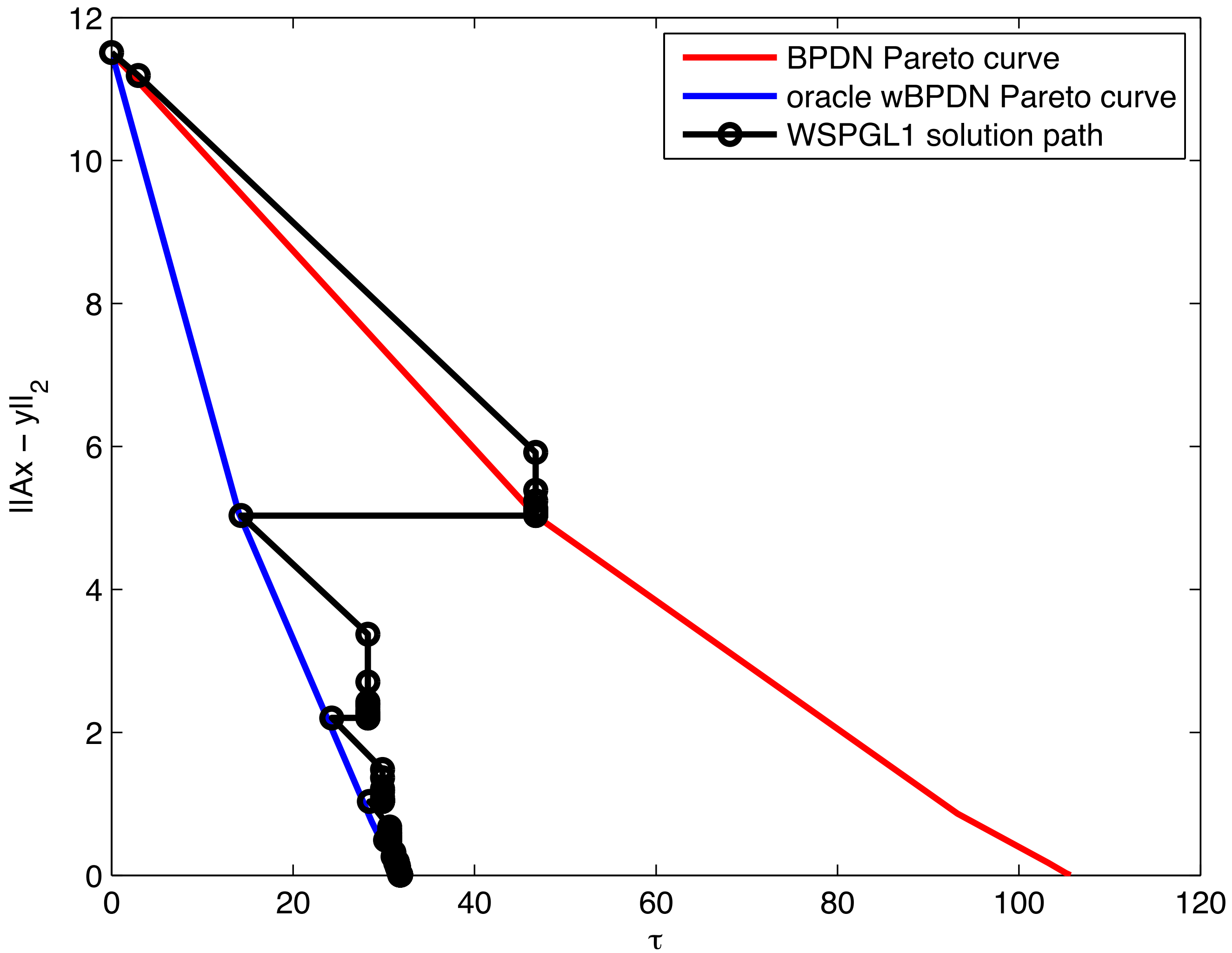
WSPGL1 and the Pareto curve

- The Pareto curve changes with the definition of every new weighted LASSO subproblem.



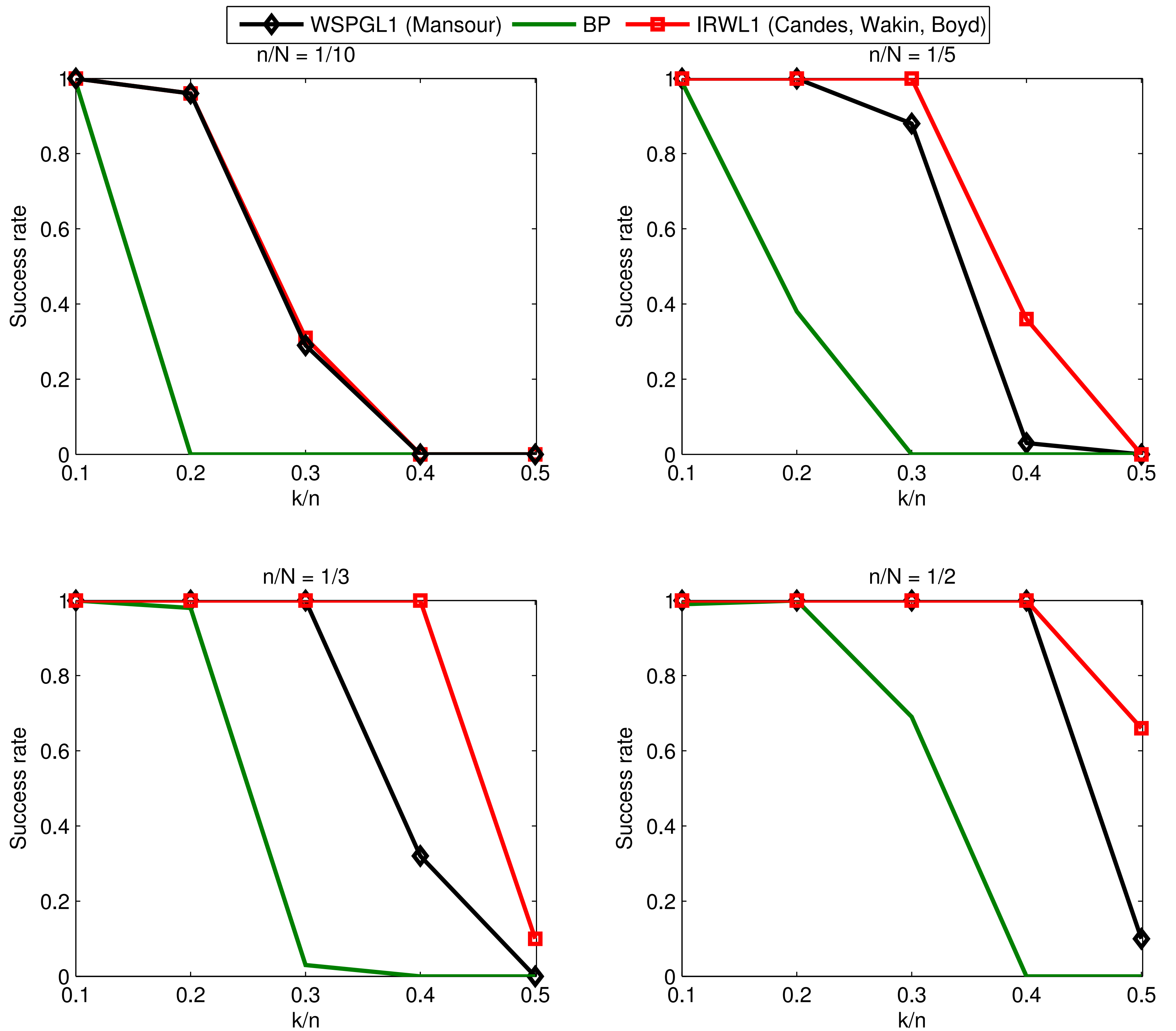
WSPGL1 and the Pareto curve

- The Pareto curve changes with the definition of every new weighted LASSO subproblem.

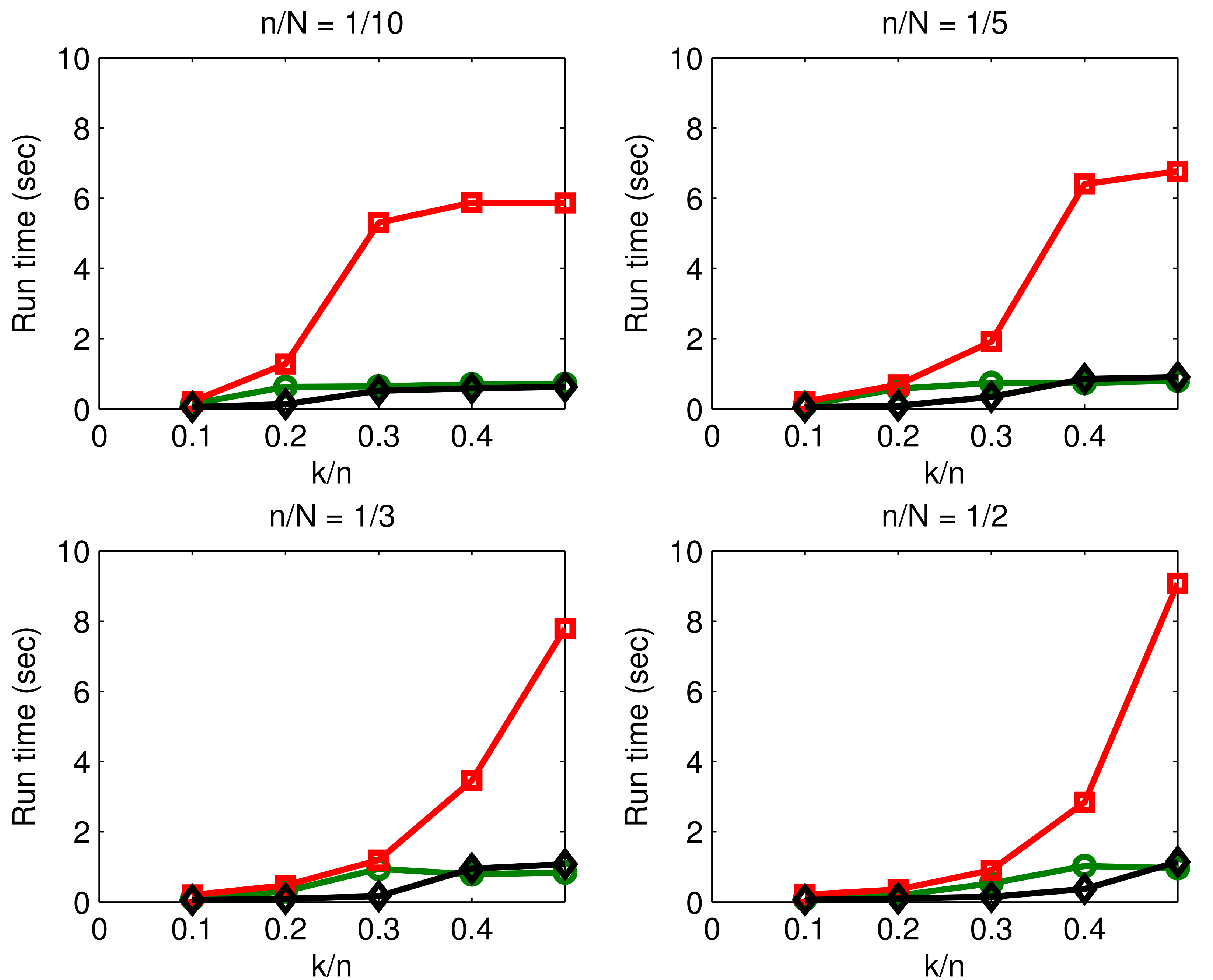


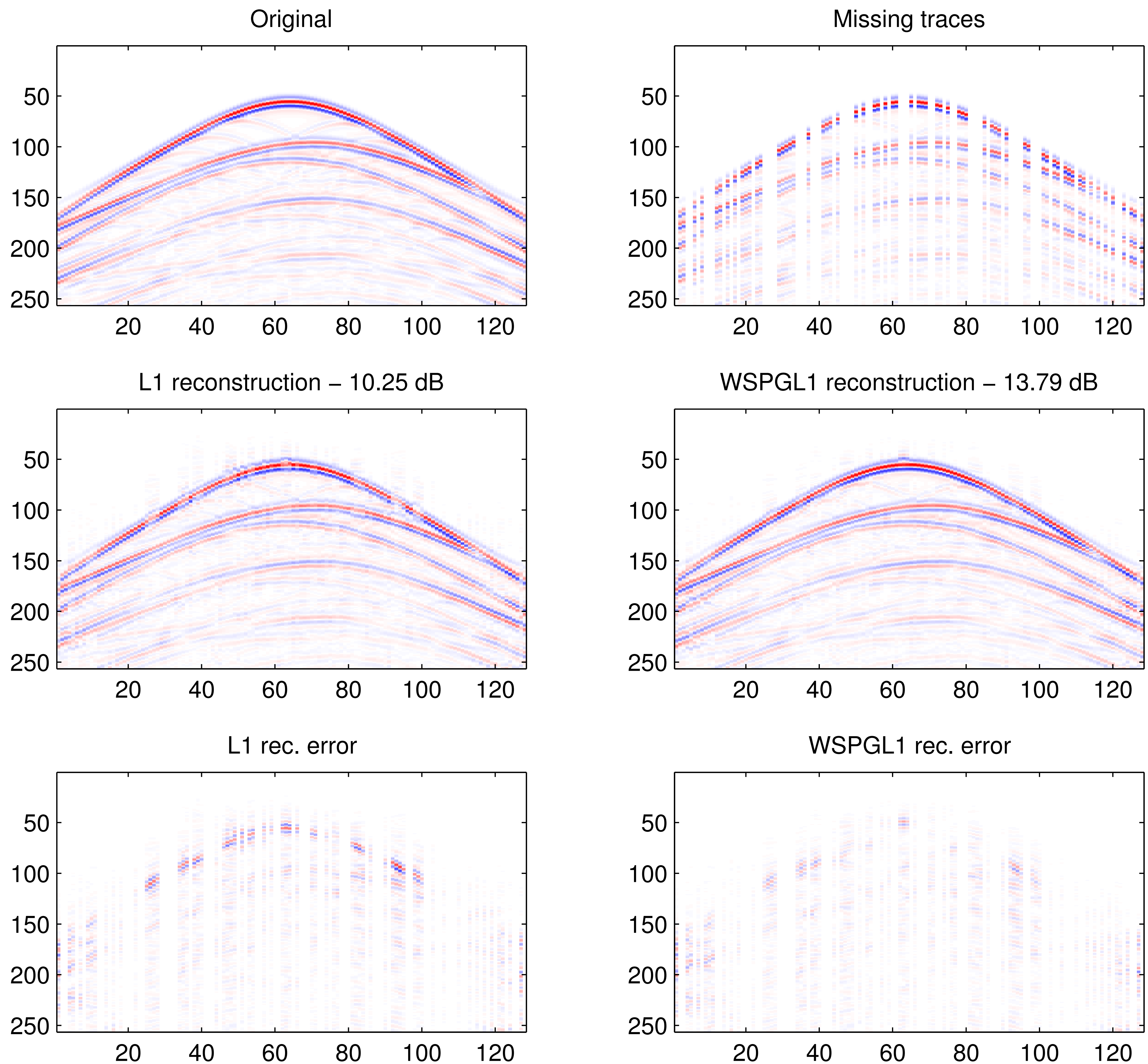
Exact recovery rate (sparse signal, no noise)

N = 1000



Algorithm runtime




Original
Missing traces
L1 reconstruction – 10.25 dB
WSPGL1 reconstruction – 13.79 dB
L1 rec. error
WSPGL1 rec. error
50
100
150
200
250
20
40
60
80
100
120

Part 1: Compressed sensing and sparse recovery

Part 2: Weighted ℓ_1 minimization

Part 3: ℓ_1 solvers and the WSPGL1 algorithm

Part 4: Sparse randomized Kaczmarz

Randomized Kaczmarz (Strohmer, Vershynin '06)

- Consider the overdetermined linear system: $Ax = b$.
- The randomized Kaczmarz (RK) algorithm solves for x by acting on individual rows of A.
- In every iteration j:
  - Select a row indexed by a_i indexed by $i \in \{1, \dots, m\}$ with probability $\frac{\|a_i\|_2^2}{\|A\|_F^2}$.
  - Project x_{j-1} onto the solution space of $\langle a_i, x \rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle a_i, x_{j-1} \rangle}{\|a_i\|_2^2} a_i^T$$

- RK is simple, memory efficient, and converges linearly.

Randomized Kaczmarz (Strohmer, Vershynin '06)

- Consider the overdetermined linear system: $Ax = b$.
- The randomized Kaczmarz (RK) algorithm solves for x by acting on individual rows of A.
- In every iteration j:
  - Select a row indexed by a_i indexed by $i \in \{1, \ldots, m\}$ with probability $\frac{\|a_i\|_2^2}{\|A\|_F^2}$.
  - Project x_{j-1} onto the solution space of $\langle a_i, x\rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle a_i, x_{j-1}\rangle}{\|a_i\|_2^2} a_i^T$$

- RK is simple, memory efficient, and converges linearly.

Randomized Kaczmarz (Strohmer, Vershynin '06)

- Consider the overdetermined linear system: $Ax = b$.
- The randomized Kaczmarz (RK) algorithm solves for x by acting on individual rows of A.
- In every iteration j:
  - Select a row indexed by a_i indexed by $i \in \{1, \ldots m\}$ with probability $\frac{\|a_i\|_2^2}{\|A\|_F^2}$.
  - Project x_{j-1} onto the solution space of $\langle a_i, x \rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle a_i, x_{j-1} \rangle}{\|a_i\|_2^2} a_i^T$$

- RK is simple, memory efficient, and converges linearly.

Randomized Kaczmarz (Strohmer, Vershynin '06)

- Consider the overdetermined linear system: $Ax = b$.
- The randomized Kaczmarz (RK) algorithm solves for x by acting on individual rows of A.
- In every iteration j:
  - Select a row indexed by a_i indexed by $i \in \{1, \dots m\}$ with probability $\frac{\|a_i\|_2^2}{\|A\|_F^2}$.
  - Project x_{j-1} onto the solution space of $\langle a_i, x \rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle a_i, x_{j-1} \rangle}{\|a_i\|_2^2} a_i^T$$

- RK is simple, memory efficient, and converges linearly.

Sparse randomized Kaczmarz (Mansour, Yilmaz)

- If x is sparse, can we speed up the convergence of RK? Certainly!
- Using the same row selection as RK, in every iteration j:
  - Identify the support estimate $S = \text{supp}(x_{j-1}|_{\max\{k, n-j+1\}})$.
  - Define the weight vector w_j such that

$$w_j(l) = \begin{cases} 1 & , l \in S \\ \frac{1}{\sqrt{j}} & , l \in S^c \end{cases}$$

  - Approximately project x_{j-1} onto the solution space of [illegible] using

[illegible]

Sparse randomized Kaczmarz (Mansour, Yilmaz)

- If x is sparse, can we speed up the convergence of RK? Certainly!
- Using the same row selection as RK, in every iteration j:
  - Identify the support estimate $S = \text{supp}(x_{j-1}|_{\max\{\hat{k}, n-j+1\}})$.
  - Define the weight vector w_j such that

$$\mathrm{w}_j(l) = \begin{cases} 1 & , l \in S \\ \frac{1}{\sqrt{j}} & , l \in S^c \end{cases}$$

  - Approximately project x_{j-1} onto the solution space of $\langle \mathrm{w}_j \odot a_i, x \rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle \mathrm{w}_j \odot a_i, x_{j-1} \rangle}{\|\mathrm{w}_j \odot a_i\|_2^2} (\mathrm{w}_j \odot a_i)^T$$

Sparse randomized Kaczmarz (Mansour, Yilmaz)

- If x is sparse, can we speed up the convergence of RK? Certainly!
- Using the same row selection as RK, in every iteration j:
  - Identify the support estimate $S = \text{supp}(x_{j-1}|_{\max\{\hat{k}, n-j+1\}})$.
  - Define the weight vector w_j such that

$$\mathrm{w}_j(l) = \begin{cases} 1 & , l \in S \\ \frac{1}{\sqrt{j}} & , l \in S^c \end{cases}$$

  - Approximately project x_{j-1} onto the solution space of $\langle \mathrm{w}_j \odot a_i, x \rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle \mathrm{w}_j \odot a_i, x_{j-1} \rangle}{\|\mathrm{w}_j \odot a_i\|_2^2} (\mathrm{w}_j \odot a_i)^T$$

Sparse randomized Kaczmarz (Mansour, Yilmaz)

- If x is sparse, can we speed up the convergence of RK? Certainly!
- Using the same row selection as RK, in every iteration j:
  - Identify the support estimate $S = \text{supp}(x_{j-1}|_{\max\{\hat{k}, n-j+1\}})$.
  - Define the weight vector w_j such that

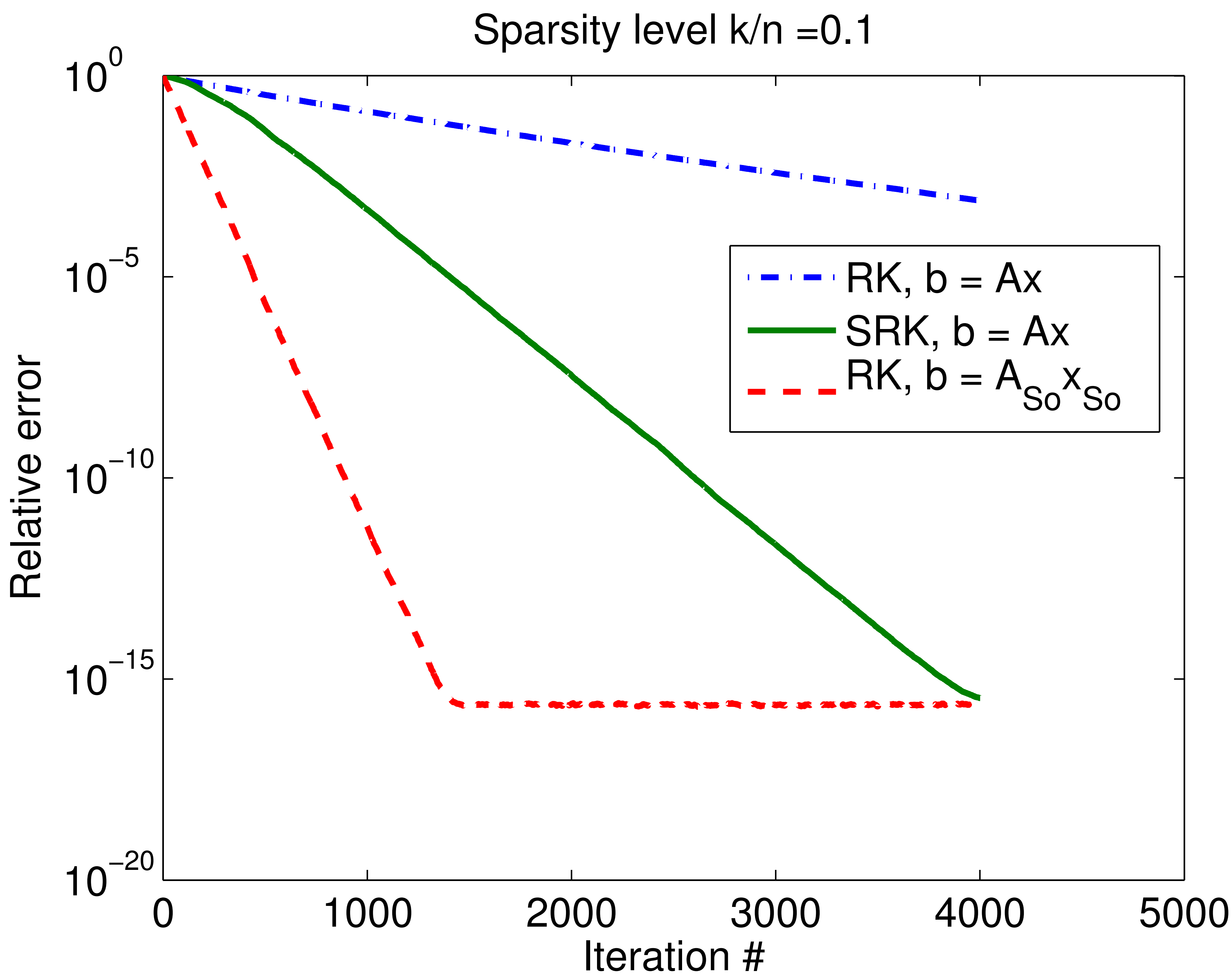
$$\mathrm{w}_j(l) = \begin{cases} 1 & , l \in S \\ \frac{1}{\sqrt{j}} & , l \in S^c \end{cases}$$

  - Approximately project x_{j-1} onto the solution space of $\langle \mathrm{w}_j \odot a_i, x \rangle = b(i)$ using

$$x_j = x_{j-1} + \frac{b(i) - \langle \mathrm{w}_j \odot a_i, x_{j-1} \rangle}{\|\mathrm{w}_j \odot a_i\|_2^2} (\mathrm{w}_j \odot a_i)^T$$

Convergence rates: overdetermined system

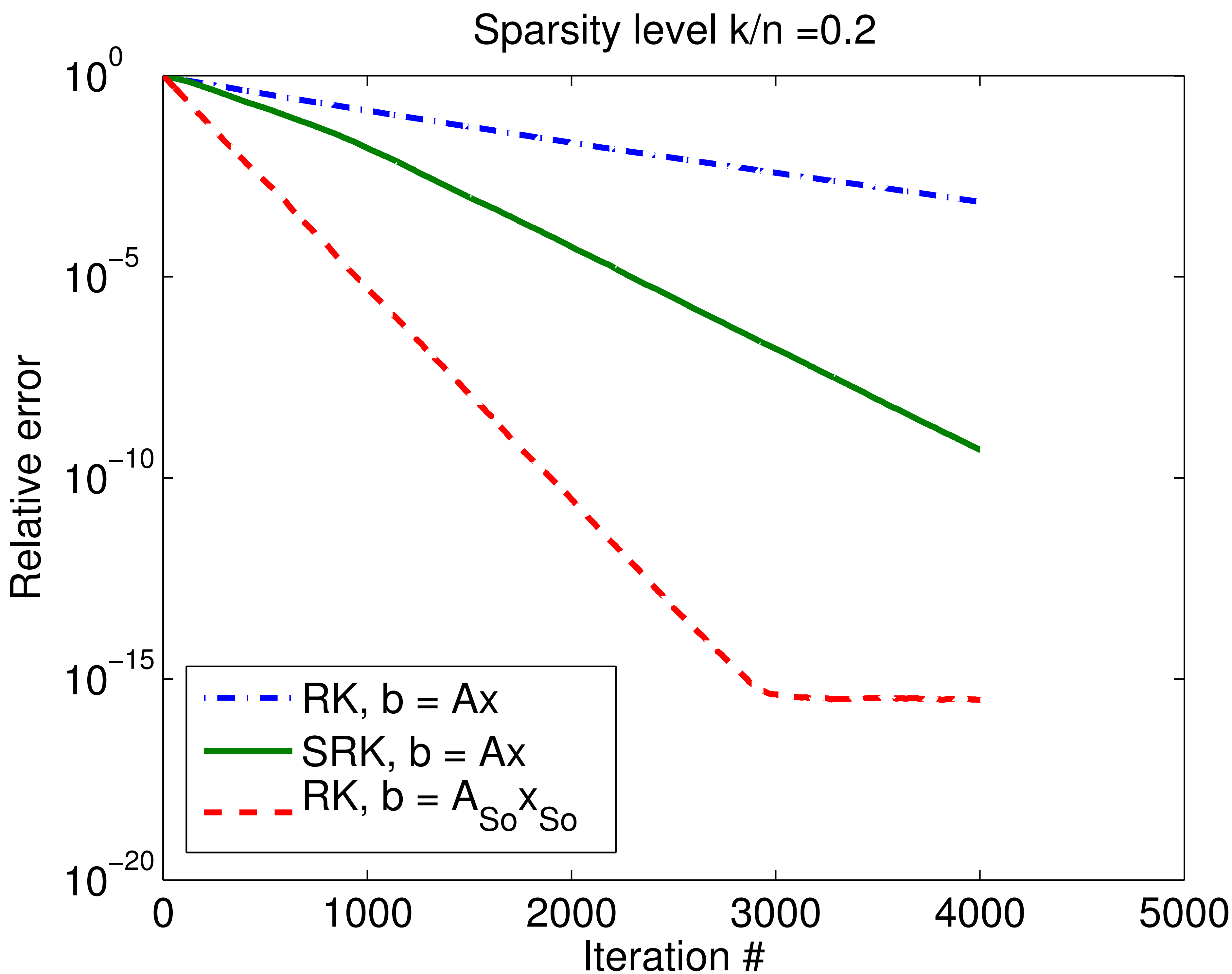
1000×200 Gaussian matrix A



Monday, 3 December, 12

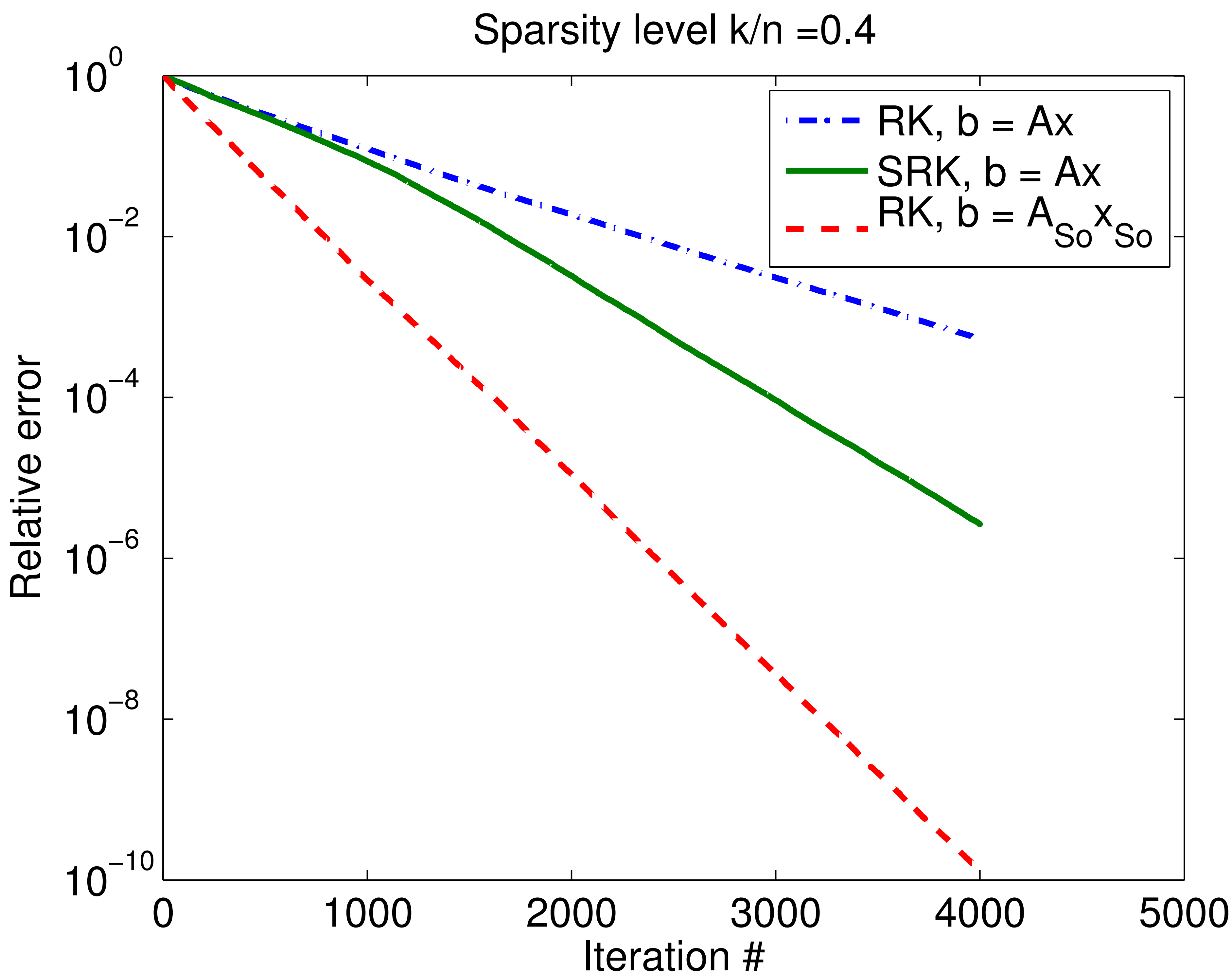
Convergence rates: overdetermined system

1000×200 Gaussian matrix A



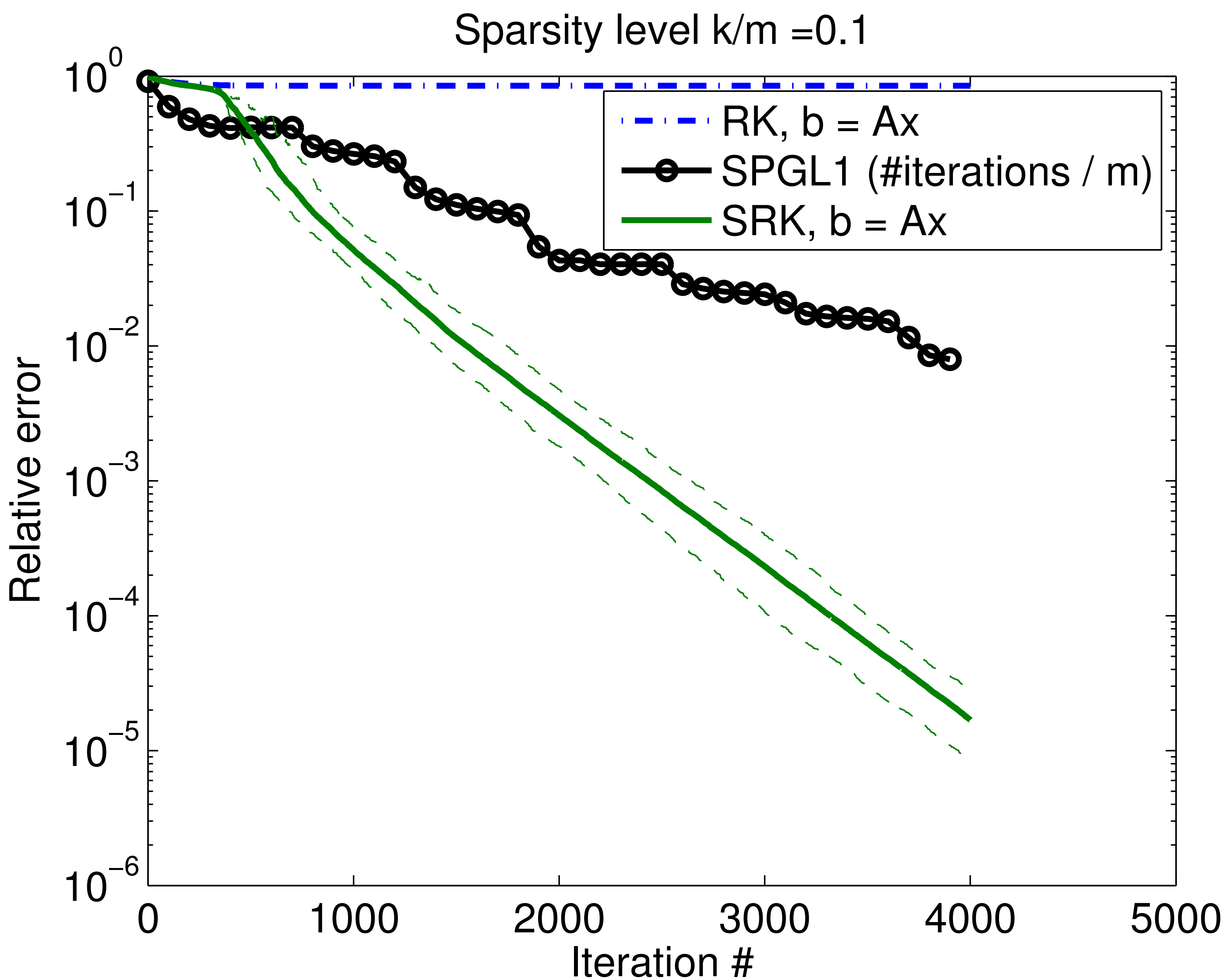
Convergence rates: overdetermined system

1000×200 Gaussian matrix A



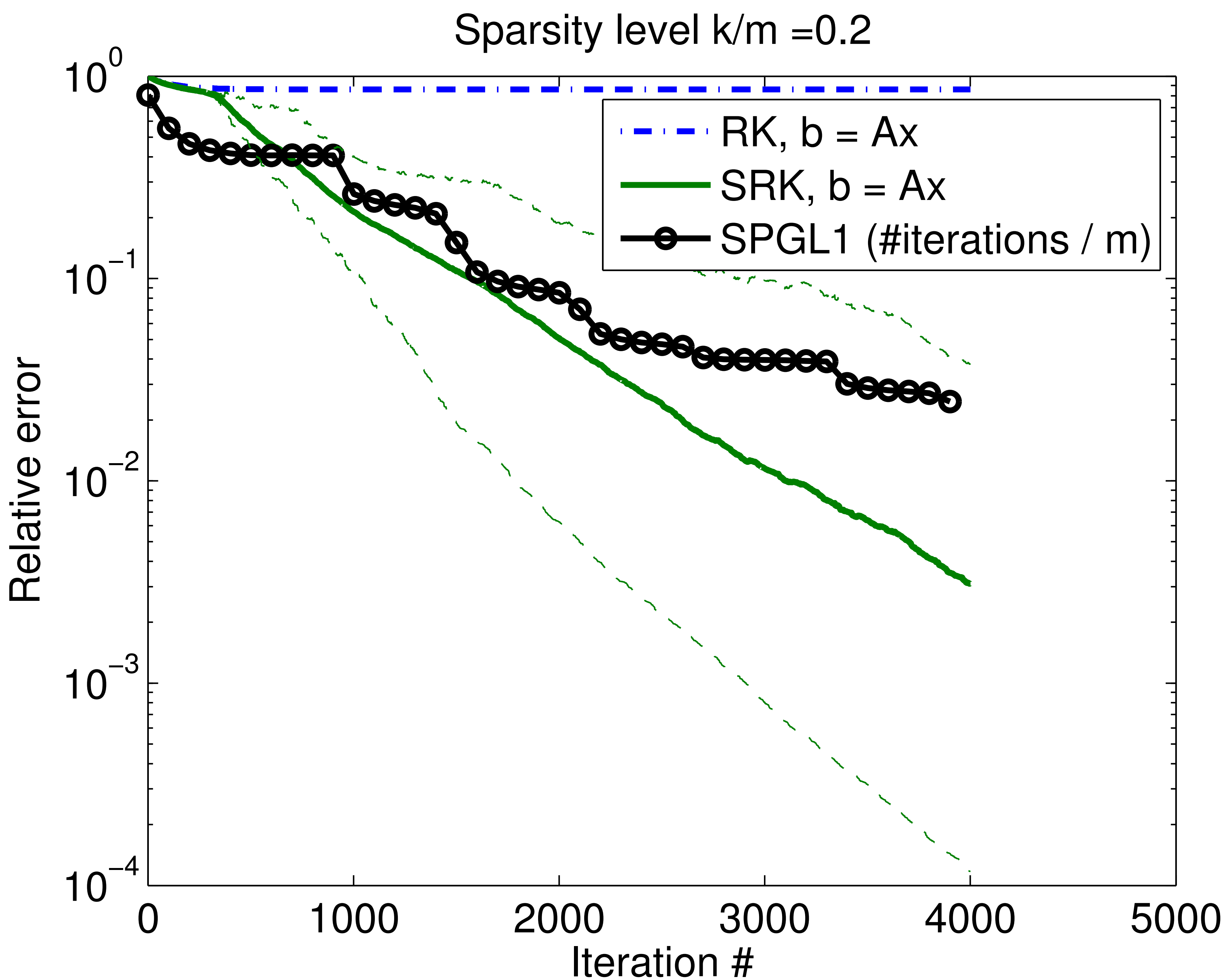
Convergence rates: underdetermined system

100×400 Gaussian matrix A



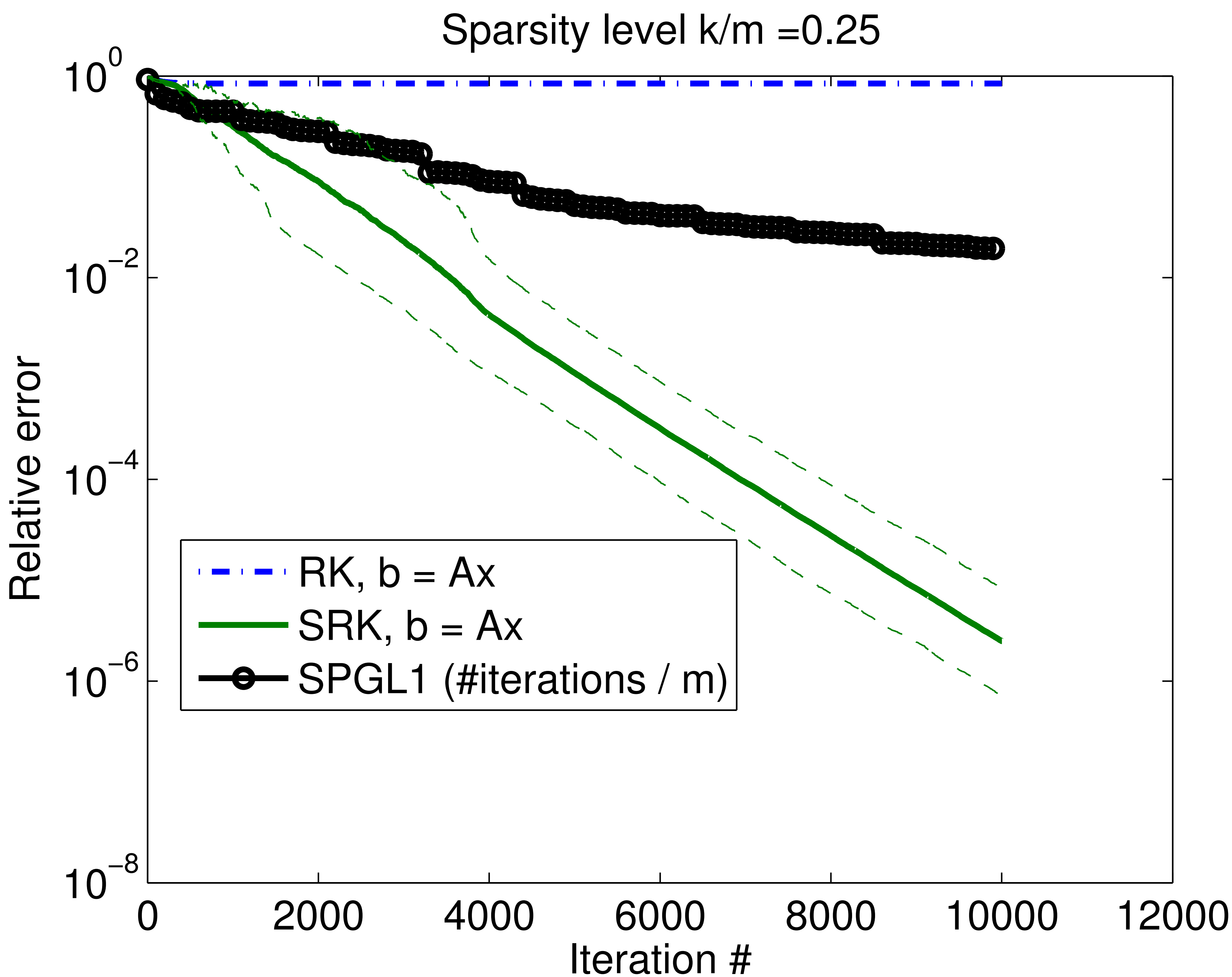
Convergence rates: underdetermined system

100×400 Gaussian matrix A



Convergence rates: underdetermined system

100×400 Gaussian matrix A



Extensions and Works In Progress (with T. van Leeuwen)

- FWI put simply is a massive nonlinear least-squares problem with an expensive Jacobian:

$$m^* = \arg\min_m \frac{1}{2}\|d - \mathcal{F}[m, Q]\|_2^2$$

m: velocity model

d: multi-source multi-frequency data residue

Q: sources

$\mathcal{F}[m, Q]$: discretization of the inverse Helmholtz operator

Extensions and Works In Progress (with T. van Leeuwen)

- Linearized least squares migration:

$$\delta\tilde{m} = \arg\min_{\delta m} \frac{1}{2}\|\delta d - J[m_0, Q]\delta m\|_2^2$$

Huge overdetermined system!

δm: model update

δd: multi-source multi-frequency data residue

m_0: background velocity model

Q: sources

$J[m_0, Q] := \nabla\mathcal{F}[m_0, Q]$: linearized Born-scattering operator

Extensions and Works In Progress (with T. van Leeuwen)

- Linearized least squares migration:

$$\delta\tilde{m} = \arg\min_{\delta m} \frac{1}{2}\|\delta d - J[m_0, Q]\delta m\|_2^2$$

- Apply a sparse randomized Kaczmarz approach to solving the least-squares migration problem.
- The algorithm can also be applied matrix-free:

$$x_j = x_{j-1} + (W_j J_i)^\dagger \left(b(i) - \langle W_j J_i, x_{j-1}\rangle\right)$$

Extensions and Works In Progress (with T. van Leeuwen)

- Linearized least squares migration:

$$\delta\tilde{m} = \arg\min_{\delta m} \frac{1}{2}\|\delta d - J[m_0, Q]\delta m\|_2^2$$

- The data δd is a function of the $\#rec$, $\#src$, and $\#freq$.
- The operator J_i corresponds to the Born-scattering operator of:
  - single receiver, single source, single frequency
  - simultaneous receivers, single source, single frequency
  - all receivers, simultaneous sources, single frequency (block Kaczmarz)

Extensions and Works In Progress (with T. van Leeuwen)

- Linearized least squares migration:

$$\delta\tilde{m} = \arg\min_{\delta m} \frac{1}{2}\|\delta d - J[m_0, Q]\delta m\|_2^2$$

- The data δd is a function of the $\#rec$, $\#src$, and $\#freq$.
- The operator J_i corresponds to the Born-scattering operator of:
  - single receiver, single source, single frequency
  - simultaneous receivers, single source, single frequency
  - all receivers, simultaneous sources, single frequency (block Kaczmarz)

Extensions and Works In Progress (with T. van Leeuwen)

- Linearized least squares migration:

$$\delta\tilde{m} = \arg\min_{\delta m} \frac{1}{2}\|\delta d - J[m_0, Q]\delta m\|_2^2$$

- The data δd is a function of the $\#rec$, $\#src$, and $\#freq$.
- The operator J_i corresponds to the Born-scattering operator of:
    - single receiver, single source, single frequency
    - simultaneous receivers, single source, single frequency
    - all receivers, simultaneous sources, single frequency (block Kaczmarz)

Extensions and Works In Progress (with T. van Leeuwen)

- Linearized least squares migration:

$$\delta\tilde{m} = \arg\min_{\delta m} \frac{1}{2}\|\delta d - J[m_0, Q]\delta m\|_2^2$$

- The data δd is a function of the $\#rec$, $\#src$, and $\#freq$.
- The operator J_i corresponds to the Born-scattering operator of:
  - single receiver, single source, single frequency
  - simultaneous receivers, single source, single frequency
  - all receivers, simultaneous sources, single frequency (block Kaczmarz)

Conclusion

Scope of this talk:

- Compressed sensing with prior support information.
- The computationally efficient WSPGL1 algorithm.
- Sparse randomized Kaczmarz and its relation to LSM.

Conclusion

Scope of this talk:

- Compressed sensing with prior support information.
- The computationally efficient WSPGL1 algorithm.
- Sparse randomized Kaczmarz and its relation to LSM.

Conclusion

Scope of this talk:

- Compressed sensing with prior support information.
- The computationally efficient WSPGL1 algorithm.
- Sparse randomized Kaczmarz and its relation to LSM.

Thank you

Questions?