Greedy algorithms for multi-channel sparse recovery
Public ISP seminar at UCL

Jean-François Determe
Summary

- **Main topic:** How noise impacts (S)OMP & overview of noise stabilization with SOMP

- **Outline:**
  - Introduction compressive sensing (7 slides -> 9 minutes)
  - Support recovery algorithms (3 slides -> 5 minutes)
  - Multiple measurement vector signal models (4 slides -> 6 minutes)
  - Analysis of SOMP with noise (8 slides -> 12 minutes)
  - SOMP with noise stabilization (6 slides -> 8 minutes)
  - Conclusion (2 minutes)

- **Total time for presentation:** about 40-45 minutes + Q&A

- Presentation = only an overview of my work (no technical details, not every contribution)
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
Compressive sensing (CS)

Idea: Observe and recover a signal \( f \in \mathbb{R}^n \) using \( m \ll n \) linear measurements:

\[ y = \Phi f \in \mathbb{R}^m \]

where \( \Phi \in \mathbb{R}^{m \times n} \) describes the measurement process.

Problem: Since \( m \ll n \), arbitrary signals \( f \) cannot be recovered.

Solution: Assume prior knowledge/structure about \( f \).

\textbf{Sparsity}: \( f \) can be expressed using \( s < m \) vectors from the appropriate o.n. basis \( \Psi \)

\[ f = \Psi x = \sum_{j=1}^{n} x_j \psi_j \]

Few non-zero \( x_j \)
Support explanation

\[ f = \Psi x = \sum_{j=1}^{n} x_j \psi_j = \sum_{j \in \text{supp}(x)} x_j \psi_j \]

\[ \text{supp}(x) := \{ j : x_j \neq 0 \} \]

\[ y = \Psi x = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & \psi_4 & \psi_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = x_1 \psi_1 + x_2 \psi_2 + x_3 \psi_3 + \ldots \]

- Sparsity example:

\[ y = \Psi x = \begin{bmatrix} \psi_1 & \psi_2 & \psi_3 & \psi_4 & \psi_5 \end{bmatrix} \begin{bmatrix} 0 \\ x_2 \\ 0 \\ 0 \\ x_5 \end{bmatrix} = x_2 \psi_2 + x_5 \psi_5 \]

\[ \text{supp}(x) = \{ 2, 5 \} \]
Compressive sensing (CS)

Idea: Observe and recover a signal $f \in \mathbb{R}^n$ using $m << n$ linear measurements:

$$y = \Phi f \in \mathbb{R}^m \text{ where } \Phi \in \mathbb{R}^{m \times n} \text{ describes the measurement process}$$

Problem: Since $m << n$, arbitrary signals $f$ cannot be recovered

Solution: Assume prior knowledge/structure about $f$

**Sparsity**: $f$ can be expressed using $s < m$ vectors from the appropriate o.n. basis $\Psi$

$$f = \Psi x = \sum_{j=1}^{n} x_j \psi_j = \sum_{j \in \text{supp}(x)} x_j \psi_j \quad \rightarrow \quad y = \Phi \Psi x$$

Few non-zero $x_j$

supp$(x) := \{ j : x_j \neq 0 \}$ has low cardinality

In practice: $\Phi$ is generated randomly using sub-Gaussian distributions

$\rightarrow$ $\Phi$ and $\Phi \Psi$ satisfy the required properties for CS with similar probabilities

$\rightarrow$ Simplification: $\Psi = I_{n \times n}$ and $y = \Phi x = \sum_{j=1}^{n} x_j \phi_j = \sum_{j \in \text{supp}(x)} x_j \phi_j$
Compressive sensing (CS)

\[ \Psi = I_{n \times n} \quad \text{and} \quad y = \Phi x = \sum_{j \in \text{supp}(x)} x_j \phi_j \]

In practice: Meas. matrix \( \Phi \) can have Gaussian entries or Rademacher entries (+/- 1 with equal probabilities) + normalization factor

Two ways to understand why random projections are neat

- Recovering \( x \) is more easy using random, diverse projections (very similar projections are not efficient to capture information about \( x \) )
- Proper random entries in \( \Phi \) make the atoms \( \phi_j \) “more orthogonal” to one another. Easier to distinguish atoms in the sum

\[ y = \sum_{j \in \text{supp}(x)} x_j \phi_j \]

Quantity \( \langle y, \phi_j \rangle \) becomes a good proxy for \( x_j \)
Sparse (compressible) 1D signal Example

- D14 wavelets – Level of decomposition = 3
Sparse (compressible) 2D signal

Example

- D8 wavelets – Level of decomposition = 2
**RIP and RICs**

**Idea:** Observe and recover a sparse signal $\mathbf{x} \in \mathbb{R}^n$ using $m \ll n$ linear measurements:

$$\mathbf{y} = \Phi \mathbf{x} \in \mathbb{R}^m \quad \text{where} \quad \Phi \in \mathbb{R}^{m \times n} \text{ describes the measurement process}$$

**Question:** How to quantify how good the measurement matrix $\Phi$ is?

**Solution:** Restricted isometry property (and associated RICs)

**RIP:** $\Phi$ satisfies the RIP (with a RIC of order $s \ \delta_s$) if

$$\left( 1 - \delta_s \right) \| \mathbf{u} \|^2_2 \leq \| \Phi \mathbf{u} \|^2_2 \leq \left( 1 + \delta_s \right) \| \mathbf{u} \|^2_2$$

for any $s$-sparse vector $\mathbf{u}$

**Interpretation:** RICs quantify to what extent a measurement matrix is suitable for CS

<table>
<thead>
<tr>
<th>Good RIC</th>
<th>Bad RIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_s \approx 0$</td>
<td>$\tilde{\delta}_s \approx 1$</td>
</tr>
</tbody>
</table>
Outline

- Introduction to compressive sensing
- **Support recovery algorithms**
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
Support recovery algorithms

Idea: Observe and recover a sparse signal $\mathbf{x} \in \mathbb{R}^n$ using $m \ll n$ linear measurements:

$$\mathbf{y} = \mathbf{\Phi x} \in \mathbb{R}^m$$

where $\mathbf{\Phi} \in \mathbb{R}^{m \times n}$ describes the measurement process.

Several algorithms can recover the support of $\mathbf{x}$ on the basis of $\mathbf{\Phi}$ and $\mathbf{y}$.

Two main classes of support recovery algorithms:

- Algorithms based upon **convex optimization** (e.g., basis pursuit, basis pursuit denoising, and Dantzig selector)
  - Higher computational requirements (CPU time + memory)
  - Best performance (theoretical + numerical)
- **Greedy** algorithms (e.g., MP, OMP, CoSaMP, and SP)
  - Lower computational requirements
  - May be less reliable than, e.g., basis pursuit.

My thesis focuses on greedy algorithms (OMP-like algorithms)
Orthogonal matching pursuit (OMP) tries to express the measurement vector $\mathbf{y}$ using columns from $\Phi$.

→ Generates an estimated support (its size is prescribed beforehand)

- OMP = iterative algorithm
  - Adds one element to estimated support at each iteration
- At each iteration:
  - Look for atom/column $\phi_j$ most closely resembling the measurement vector $\mathbf{y}$ → inner product
  - Add this atom to estimated support
  - Remove the atom contribution to the measurements (→ approximation only)
  If $S_t = \text{estimated support at iteration } t$ => build proxy for

\[
\Phi_{S \setminus S_t} \mathbf{x}_{S \setminus S_t} = \sum_{j \in S \setminus S_t} x_j \phi_j
\]
Performance comparison for greedy algorithms

Noiseless case

\[ y = \Phi x \]

Noisy case

\[ y = \Phi x + e \]

OMP not competitive

OMP competitive
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
Extension of basic CS model: - Multiple measurement vector (MMV) signal model
- Additive Gaussian noise with \( \neq \) variances

\( K \) sparse signals/measurement channels/measurement vectors:

\[
y_k = \Phi x_k + e_k \quad (1 \leq k \leq K)
\]

where \( e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m}) \)
**MMV Signal model (2)**

**Extension of basic CS model:**
- Multiple measurement vector (MMV) signal model
- Additive Gaussian noise with \( \neq \) variances

\[ y_k = \Phi x_k + e_k \quad (1 \leq k \leq K) \quad \text{where} \quad e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m}) \]

With matrices:
\[ Y = \Phi X + E \]
\[ X \in \mathbb{R}^{n \times K}, \quad \Phi \in \mathbb{R}^{m \times n}, \quad \text{and} \quad Y, E \in \mathbb{R}^{m \times K} \]

\[ Y = (y_1, \ldots, y_K) \quad X = (x_1, \ldots, x_K) \quad E = (e_1, \ldots, e_K) \]

**Joint support:**
\[ S := \text{supp}(X) := \bigcup_{1 \leq k \leq K} \text{supp}(x_k) \]

**Objective:** Recover the joint support on the basis of \( \{y_k\}_{1 \leq k \leq K} \) and \( \Phi \).
Remarks on MMV signal models

\[ y_k = \Phi x_k + e_k \quad (1 \leq k \leq K) \text{ where } e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m}) \]

- MMV extensions of SMV algorithms are available (e.g., SOMP, SCoSaMP, SSP)

- Focus of this presentation = SOMP exclusively

- Several applications for MMV signal models:
  - **Source localization**: each measurement vector corresponds to a specific time instant
  - **Localization in 5G networks**
  - **Spectrum sensing/sub-Nyquist acquisition** with the modulated wideband converter
Simultaneous orthogonal matching pursuit (SOMP) tries to jointly express the $K$ measurement vectors $\mathbf{y}_k$ using a *unique* set of columns from $\Phi$.

**Joint support recovery, i.e., one common support** for all the sparse signals $\mathbf{x}_k$.

- **SOMP** = iterative algorithm
  - Adds *one* element to estimated support at each iteration
- At each iteration:
  - Look for atom/column $\phi_j$ most closely resembling *all* the measurement vectors $\mathbf{y}_k$
  - Add this atom to estimated support
  - Remove the atom contribution to the measurements (-> approximation only)

If $S_t = \text{estimated support at iteration } t \Rightarrow \text{build proxy for}$

$$\Phi_{S\setminus S_t} (\mathbf{x}_k)_{S\setminus S_t} = \sum_{j \in S \setminus S_t} (\mathbf{x}_k)_j \phi_j$$
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
Analysis of SOMP with noise

Objective & Main quantities

- Noisy signal model with additive Gaussian measurement noise

- General objective: understand how the additive Gaussian noise affect the performance of SOMP

- Main results:
  - Upper bound on the probability that SOMP fails (i.e., picks an incorrect atom) for \( s+1 \) iterations
  - Corresponding minimal value of \( K \) for a prescribed maximum probability of failure
  - Numerical results confirm the theory is (mostly) correct

\[
y_k = \Phi x_k + e_k \quad \text{where} \quad e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m})
\]
Analysis of SOMP with noise
Quantities without noise

\[ y_k = \Phi x_k + e_k \quad \text{where} \quad e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m}) \]

- Iteration \( t \), quantities from the **noiseless** case:
  \[ \gamma_c^{(t,P)} = \text{Highest value of SOMP metric for correct atoms} \]
  \[ \gamma_i^{(t,P)} = \text{Highest value of SOMP metric for incorrect atoms} \]

\[ \frac{\gamma_c^{(t,P)}}{\gamma_i^{(t,P)}} \geq \Gamma > 1 \text{ for any iteration } t. \]

\( \Gamma > 1 \Rightarrow \text{correct decisions in noiseless case} \)

\[ \gamma_c^{(t,P)} \geq \psi \tau_X \text{ for any iteration } t \]

\[ \Gamma = \frac{1 - \delta \vert S \vert + 1}{\delta \vert S \vert + 1 \sqrt{\vert S \vert}} \]

\[ \tau_X = \min_j \sum_{k=1}^K |X_{j,k}| \]

\[ \psi = \frac{(1 - \delta \vert S \vert)(1 + \delta \vert S \vert)}{1 + \sqrt{\vert S \vert} \delta \vert S \vert} \]

\[ \gamma_c^{(t,P)} - \gamma_i^{(t,P)} \geq \psi \tau_X \left(1 - \frac{1}{\Gamma}\right) \]
Upper bound on the probability of error of SOMP for \( |S| \) iterations

\[
y_k = \Phi x_k + e_k \quad \text{where} \quad e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m})
\]

\[
\xi := \left(1 - \frac{1}{\Gamma}\right) \psi \text{SNR}_{\text{min}} - \sqrt{\frac{2}{\pi}} \omega \sigma =: \alpha
\]

\[
C_s := \sum_{t=0}^{s} \binom{|S|}{t}
\]

- **Upper bound on the probability of error**

\[
nC_{|S|+1} \exp \left[ -\frac{1}{8} K \xi^2 \right]
\]

- **Main interpretations:**
  - \( \xi < 0 \) \(
  \Rightarrow \) probability of failure might be 1 as \( K \to \infty \)
  - Both meas. matrix \( \Phi \) and SNR should be « good » enough when compared to noise
  - Prob. failure decreases exponentially with \( K \) if \( \xi > 0 \)

- Detailed interpretation of each quantity on next slide
Min. value of $K$ for given probability of error (1)

$y_k = \Phi x_k + e_k$ where $e_k \sim \mathcal{N}(0, \sigma_k^2 I_{m \times m})$

- Upper bound on the probability of error for $|S|$ iterations: $nC_s \exp \left[ -\frac{1}{8} K \xi^2 \right]$

- Minimum value of $K$ to achieve probability of error $p_{err}$ for $|S|$ iterations:
  
  $$K_{\text{min}}(p_{err}) \leq \frac{8}{\xi^2} \log \left( \frac{nC_{|S|-1}}{p_{err}} \right) \quad \text{with} \quad \xi = \alpha \text{SNR}_m - \beta \omega_{\sigma}$$

- $\alpha$: to what extent is $\Phi$ appropriately designed ($0 < \alpha \leq 1$)?
- $\text{SNR}_m$: signal-to-noise ratio for all the $K$ channels
- $\alpha \text{SNR}_m$: term related to SNR and quality of meas. matrix $\Phi$
- $\omega_{\sigma}$: penalty depending on noise std. dev. uniformity ($\rightarrow$ sparsity of $\sigma$)
- $\beta$: theoretical constant ($\beta \leq \sqrt{2/\pi}$)
- $\beta \omega(\sigma_1, \ldots, \sigma_K)$: noise-related penalty on robustness without noise
- $nC_{|S|-1}$: increases with # of atoms $n$ and support size $|S|$
  - Theoretical expression is not sharp
Min. value of \( K \) for given probability of error \( p_{err} \):

\[
K_{\text{min}}(p_{err}) := \frac{8}{\xi^2} \log \left( \frac{nC_{|S|-1}}{p_{err}} \right) \quad \xi = \alpha \text{ SNR}_m - \beta \omega \alpha
\]

- Minimum value of \( K \) to achieve probability of error \( p_{err} \):

- Rewrites

\[
K_{\text{min}}(p_{err}) := \frac{8}{(\alpha \text{ SNR}_{\text{min}} - \omega \sigma \beta)^2} (\gamma - \log p_{err})
\]

with \( \gamma := \log (nC_{|S|-1}) \)

- Useful for simulations
Simulation framework

- **Goal:** Validate theoretical analysis
- **Method:** Carry out simulations and compare results with formula

\[
K_{\text{min}}(p_{\text{err}}) := \frac{8}{(\alpha \text{ SNR}_{\text{min}} - \omega \sigma \beta)^2} (\gamma - \log p_{\text{err}})
\]

- Identify the values of \( \alpha \), \( \beta \), and \( \gamma \) on the basis of simulations.
- Assess whether theoretical curve fits simulation curves
- Assess whether identified values are coherent with theory

- Detailed signal model is not described here
- Identification procedure not discussed either
Simulations - Results (1)

\[ K_{\min}(p_{err}) := \frac{8}{(\alpha \text{ SNR}_{\min} - \omega \alpha \beta)^2 (\gamma - \log p_{err})} \]

Probability of error of SOMP

(a) \(|S| = 10 - \) Identified parameters: 
\[ \alpha = 1.0535, \, \beta = 0.54045, \text{ and } \gamma = 2.0741. \]

(b) \(|S| = 20 - \) Identified parameters: 
\[ \alpha = 1.0535, \, \beta = 0.58682, \text{ and } \gamma = 2.5451. \]

Question 1: Do theoretical curves fit empirical ones?

😊 YES 😊
Simulations - Results (2)

\[ K_{\text{min}}(p_{\text{err}}) := \frac{8}{(\alpha \text{ SNR}_{\text{min}} - \omega_\sigma \beta)^2} (\gamma - \log p_{\text{err}}) \]

**Question 2:** Are the identified values coherent with the theory?

| \(|S| = 10\) – Identified parameters: \(\alpha = 1.0535\), \(\beta = 0.54045\), and \(\gamma = 2.0741\) | \(|S| = 20\) – Identified parameters: \(\alpha = 1.0535\), \(\beta = 0.58682\), and \(\gamma = 2.5451\) |
| --- | --- |

- \(\alpha\) should be \(\leq 1\) in theory but discrepancy is OK 😞
- \(\beta\) is lower than \(\sqrt{2/\pi} \approx 0.7979\) 😊
- It can be shown that \(\gamma\) is way too low wrt the theory 😞
  - but proof method explains why 😊
  - and \(\gamma := \log (nC_{|S|-1})\) increases with support cardinality \(|S|\) 😊
- See « Future work » in the thesis

**Analysis mostly OK for \(\alpha\) and \(\beta\)**
The contributions so far

- Contribution:
  - Thorough analysis of noiseless and noisy SOMP (theory + simulations)

- Related publications:
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
SOMP with noise stabilization (SOMP-NS)

\[ y_k = \Phi x_k + e_k \text{ where } e_k \sim \mathcal{N}(0, \sigma_k^2 I) \]

Require: \( Y \in \mathbb{R}^{m \times K}, \Phi \in \mathbb{R}^{m \times n}, s \geq 1, \{q_k\}_{1 \leq k \leq K} \)

1: Initialization: \( R^{(0)} \leftarrow Y \) and \( S_0 \leftarrow \emptyset \)
2: \( t \leftarrow 0 \)
3: while \( t < s \) do
4: \( j_t \leftarrow \arg\max_{1 \leq j \leq n} \sum_{k=1}^K |\langle \phi_j, r_k^{(t)} \rangle| q_k \)
5: Update the support: \( S_{t+1} \leftarrow S_t \cup \{j_t\} \)
6: Projection of each measurement vector onto \( \mathcal{R}(\Phi S_{t+1})^\perp \):
   \( R^{(t+1)} \leftarrow (I - \Phi S_{t+1} \Phi_{S_{t+1}}^+ \Phi_{S_{t+1}}) Y \)
7: \( t \leftarrow t + 1 \)
8: end while
9: return \( S_s \) {Support at last step}

Idea: The SNRs of the measurement vectors are unequal \( \rightarrow \) weight the impact of each measurement vector according to its reliability.

\[
\sum_{k=1}^K |\langle \phi_j, r_k^{(t)} \rangle| q_k \rightarrow \sum_{k=1}^K |\langle \phi_j, r_k^{(t)} \rangle| q_k
\]

Question: What are the optimal weights \( q_k \) ?

(Our) Answer: Resort to the theory and find how to minimize an upper bound on the probability of SOMP-NS failing to perform correct decisions.
Theoretical probability of failure
Optimal weights

- If we assume $|X_{j,k}| \simeq c_k|X_{j,1}|$ ($c_k > 0$) for $(j, k) \in [n] \times [K]$

$$q_k = \frac{c_k}{\sigma_k^2}$$

Formula stems from our analysis of SOMP with noise

**Question 1:** Does SOMP-NS yield improvements?

**Question 2:** Theoretically optimal weights = truly optimal weights?

⇒ Simulations
Theoretically opt. Weights vs. truly optimal ones

- $K = 2$, $\sigma := (\sigma_1, \sigma_2) = (\cos(\theta_\sigma), \sin(\theta_\sigma))$ and $q := (q_1, q_2) = (\cos(\theta_q), \sin(\theta_q))$

- Grid $\mathcal{G} := \mathcal{G}_\sigma \times \mathcal{G}_q$ of values for $\theta_\sigma$ and $\theta_q$ → evaluate the probability of SOMP-NS succeeding in recovering the support $S$ in exactly $|S|$ iterations for each 2-tuple $(\theta_\sigma, \theta_q)$

Figure 1: Simulation results for simulation setup 3.2 — The black line represents the analytically optimal weights given by $q_k = 1/\sigma_k^2$. 
Theoretically opt. Weights vs. truly optimal ones

- $K = 2$, $\sigma := (\sigma_1, \sigma_2) = (\cos(\theta_\sigma), \sin(\theta_\sigma))$ and $q := (q_1, q_2) = (\cos(\theta_q), \sin(\theta_q))$

- Grid $\mathcal{G} := \mathcal{G}_\sigma \times \mathcal{G}_q$ of values for $\theta_\sigma$ and $\theta_q$ → evaluate the probability of SOMP-NS succeeding in recovering the support $S$ in exactly $|S|$ iterations for each 2-tuple $(\theta_\sigma, \theta_q)$

- $n = 1000$, real and complex signal models (random sign or random phase), $\|\sigma\|_2^2 = 1$

- Two different signal patterns
Signal patterns

**SP1:** signal pattern with non-zero entries of \( X \) with equal moduli

\[
X = \begin{bmatrix}
1 & -1 & 1 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
-1 & -1 & -1 \\
1 & 1 & -1 \\
\vdots & \vdots & \vdots
\end{bmatrix}
\]

Similar non-zero moduli

\[ q_k = 1/\sigma_k^2 \]

**SP2:** signal pattern with Gaussian non-zero entries (N(0,1)), common for all columns, that are then normalized

\[
X = \begin{bmatrix}
3.36 & -3.36 & 3.36 \\
0 & 0 & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
-1.42 & -1.42 & -1.42 \\
1 & 1 & -1 \\
\vdots & \vdots & \vdots
\end{bmatrix}
\]

Dissimilar non-zero moduli
Theoretically opt. Weights vs. truly optimal ones (Results)

\[ q_k = \frac{1}{\sigma_k^2} \]

Figure 1: Optimal weighting angles — Simulation setups 1.1 to 1.3 — The black continuous line represents the analytically optimal weights (AOW) given by \( q_k = \frac{1}{\sigma_k^2} \).

Figure 1: Optimal weighting angles — Simulation setups 2.1 to 2.3 — The black, continuous line represents the analytically optimal weights (AOW) given by \( q_k = \frac{1}{\sigma_k^2} \).
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion
Conclusion

Contributions:
- Analysis of SOMP with and without noise
- Proposal and analysis of SOMP-NS
- Numerical validation for both contributions

Thank you for your attention!
Outline

- Introduction to compressive sensing
- Support recovery algorithms
- Multiple measurement vector signal models
- Analysis of SOMP with noise
- SOMP with noise stabilization
- Conclusion

- Backup slides
Orthogonal matching pursuit

\[ y = \Phi x = \sum_{j \in S} x_j \phi_j \]

**Require:** \( y \in \mathbb{R}^m, \Phi \in \mathbb{R}^{m \times n}, s \geq 1 \)

1. Initialization: \( r^{(0)} \leftarrow y \) and \( S_0 \leftarrow \emptyset \)
2. \( t \leftarrow 0 \)
3. **while** \( t < s \) **do**
   4. Determine the atom of \( \Phi \) to be included in the support:
      \[ j_t \leftarrow \text{argmax}_{j \in [n]} |\langle r^{(t)}, \phi_j \rangle| \]
   5. Update the support: \( S_{t+1} \leftarrow S_t \cup \{j_t\} \)
   6. Projection of the measurement vector onto \( \mathcal{R}(\Phi_{S_{t+1}}) \):
      \[ y^{(t+1)} \leftarrow \Phi_{S_{t+1}} \Phi_{S_{t+1}}^+ y \]
   7. Projection of the measurement vector onto \( \mathcal{R}(\Phi_{S_{t+1}})^{\perp} \):
      \[ r^{(t+1)} \leftarrow y - y^{(t+1)} \]
   8. \( t \leftarrow t + 1 \)
4. **end while**
5. **return** \( S_s \) \{Support at last step\}
Simulations - Results

\[ \sigma = (\sigma_{\text{odd}}, \sigma_{\text{even}}, \sigma_{\text{odd}}, \sigma_{\text{even}}, \ldots) \text{ where } r_{\sigma} = \sigma_{\text{even}} / \sigma_{\text{odd}} \]

\[ \omega_{\sigma} = \frac{1}{\sqrt{2}} \left( r_{\sigma} + 1 \right) / \sqrt{r_{\sigma}^2 + 1} \]

Figure 1: Levels sets of the probability of SOMP committing at least one error when performing the joint full support recovery — |S| = 10