Sparse Source Separation from Orthogonal Mixtures

Moshe Mishali and Yonina C. Eldar
Department of Electrical Engineering, Technion – Israel Institute of Technology

- **Problem Formulation**
  \[ \mathbf{X} = \mathbf{\Psi} \mathbf{S} \]
  Given \( \mathbf{X} \) find an orthogonal \( \mathbf{\Psi} \) and sparse \( \mathbf{S} \)

  - **Assumptions:**
    * No noise
    * \( K \) active sources (exactly) at each snapshot
    * The nonzero values are drawn from some continuous distribution (or positive).
    * See extensions for the general case

- **Stage 1 – Source Recovery**
  Orthogonal mixture preserves inner products
  \[ \mathbf{C} = \mathbf{X}^T \mathbf{X} = \mathbf{S}^T \mathbf{S} \]
  
  \(-\text{R1}\) \( C_{ik} = 0 \) implies that supp(\( \mathbf{S}_k \)) and supp(\( \mathbf{S}_j \)) are disjoint
  
  \(-\text{R2}\) \( C_{ik} \neq 0 \) implies supp(\( \mathbf{S}_k \)) \( \cap \) supp(\( \mathbf{S}_j \)) is not empty
  
  \(-\text{R3}\) Every column \( \mathbf{S}_k \) contains exactly \( K \) non-zeros
  
  \(-\text{R4}\) Row permutations of \( \mathbf{S} \) are allowed

  Construct the support by iteratively applying (R1) – (R4)

- **Stage 2 - Source and Mixture Recovery**
  
  \[ \begin{align*}
  (\hat{\mathbf{\Psi}}, \hat{\mathbf{S}}) &= \arg \min \| \mathbf{X} - \mathbf{\Psi} \mathbf{S} \|_F \\
  & \quad \text{s.t. } \mathbf{\Psi}^T \mathbf{\Psi} = \mathbf{I}, \\
  & \quad \text{supp}(\mathbf{S}_k) = \emptyset, \quad (i, k) \in \Omega_0
  \end{align*} \]

  \(-\text{Optimal } \mathbf{S} \) given \( \mathbf{\Psi} \)
  \(-\text{Optimal } \mathbf{\Psi} \) given \( \mathbf{S} \)

  Alternate minimization based on \( \nabla \mathcal{L}(\mathbf{S}, \mathbf{\Psi}) = 0 \)

- **Motivation**
  \[ (\hat{\mathbf{\Psi}}, \hat{\mathbf{S}}) = \arg \min \| \mathbf{X} - \mathbf{\Psi} \mathbf{S} \|_F \]
  
  s.t. \( \mathbf{\Psi}^T \mathbf{\Psi} = \mathbf{I}, \quad 1 \leq i \leq T \)

  \(-\text{Known methods:}
  - In [1]: \( \min_{\mathbf{\Phi}} \sum_{\alpha} \| \mathbf{\Phi}^T \mathbf{X}_{\alpha} \|_1 \)
  - In [2-4]:
    \(-\text{Combinatorial}
    \(-\text{Nonconvex}
    \(-\text{Many local minima in practice}
    \(-\text{Nonorthogonal } \mathbf{\Psi} \)

- **Algorithm**
  Sketch (full pseudo-code in paper)
  
  - Maintain a dynamic list of rules
  - Sweep through \( \mathbf{S} \) columns and update estimated support and rules list
  - Resolve singleton rules
  - Resolve rules when symmetry allows
  - Repeat till no further changes

  \(-\text{# Iterations is bounded by } nKT \)
  
  (Typically much smaller)

  Partial support recovery is usually “good enough”

- **Extensions**
  - Support recovery in the presence of noise
  - Preprocessing to determine the sparsity levels \( \| \mathbf{S}_i \|_0 \)
  - Additional update rules based on support blockness

- **Numerical Experiments**
  \(-100 \text{ trials on random } \mathbf{S}, \mathbf{\Psi} \)
  
  \(-\text{Success: } \mathbf{S} = \hat{\mathbf{S}} \) and \( \mathbf{\Psi} = \hat{\mathbf{\Psi}} \) (up to row/column permutation, resp.)

  \(-\text{K-SVD} [3]: \text{Learn an } n \times n \text{ dictionary } \mathbf{\Psi} \)

- **References (short list)**