A Fixed-Point Iterative Schema for Error Minimization in k-Sparse Decomposition

Alessandro Adamo and Giuliano Grossi

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico 39, I-20135 Milan, Italy
Email: adamo@dsi.unimi.it, grossi@dsi.unimi.it

Outline

- Sparse Representation
- Sparsity Promoting by LiMapS Algorithm
- The k-LiMapS Algorithm
- Numerical Results
- Conclusions and Future Works

Sparse Representation

- Let $\Phi = [\phi_1, \ldots, \phi_m]$ an overcomplete dictionary, with atoms in $\mathbb{R}^n$
- sparse representation for a given vector or signal $s = (s_1, \ldots, s_n)$ expressed as $s = \sum_i \alpha_i \phi_i$, is measured with $\| \alpha \|_0$
- It is simply to assume s as a noisy signal

$$s = \Phi\alpha + \varepsilon, \tag{1}$$

  where $\varepsilon \in \mathbb{R}^n$ is noise
- The minimization problem

$$\min_{\alpha \in \mathbb{R}^m} \| s - \Phi\alpha \|^2 \quad \text{subject to} \quad \| \alpha \|_0 \leq k, \qquad (\mathrm{P}_0)$$

  where $\| \cdot \|$ denotes the ℓ^2-norm

Sparsity Promoting by LiMapS Algorithm

- LIMAPS is a sparse approximation technique which consists on a fixed-point iteration schema based on a nonlinear mapping. (Do you know it? :-D)
- the method provides a parametric family of nonlinear functions $\mathcal{F} = \{f_\lambda : \mathbb{R}^m \to \mathbb{R}^m \mid \lambda \in \mathbb{R}^+\}$ where a component is defined as:

$$f_\lambda(\alpha) = \alpha \odot \left(1 - e^{-\lambda|\alpha|}\right), \qquad (2)$$

being $\odot$ the Hadamard (elementwise) product.

- Let $g_\lambda(|\alpha_i|) = 1 - e^{-\lambda|\alpha_i|}$ such that

$$\lim_{\lambda\to+\infty} g_\lambda(|\alpha_i|) = \begin{cases} 0, & \text{if } \alpha_i = 0 \\ 1, & \text{when } \alpha_i \neq 0 \end{cases}.$$

Sparsity Promoting by LiMapS Algorithm

- Combining nonlinear mappings belongings to the family (2) and orthogonal projections in the null space of matrix Φ, it can be showed that the sequence $\{\alpha^{(t)}\}$ generated by the following fixed-point schema always converges:

$$\alpha^{(t+1)} = T_{\lambda_t}(\alpha^{(t)}) = Pf_{\lambda_t}(\alpha^{(t)}) + \Phi^{\dagger} s, \tag{3}$$

where $P = I - Q$, with $Q = \Phi^{\dagger}\Phi$, is the orthogonal projector onto the null space of Φ, $\Phi^{\dagger} = (\Phi^T \Phi)^{-1}\Phi^T$ is the Moore-Penrose pseudo-inverse of Φ and $\{\lambda_t\}$ a suitable sequence satisfying $\sum_{t=0}^{\infty} 1/\lambda_t < +\infty$.

The k-LiMapS Algorithm: Definition of Parameter λ

- Fixed $1 \le k \le n$, a possible strategy for finding k-sparse solutions using LIMAPS consists on choosing $\lambda_t = \sigma_t^{-1}$ at time $t \ge 0$ satisfying

$$\sigma_t = \hat{\alpha}_{k+1}^{(t)}$$

  being $\hat{\alpha}^{(t)}$ the absolute values of $\alpha^{(t)}$ rearranged in descending order and $\hat{\alpha}_{k+1}^{(t)}$ its k-th element
- Goal:
  - to speed up the process aimed to drop the smallest coefficients, i.e., those corresponding to elements $\hat{\alpha}_j \le \sigma_t$, which have indexes in the set $\Lambda^{(t)} = \{j : |\alpha_j^{(t)}| \le \sigma_t\}$;
  - to minimize the solution error induced by $\alpha^{(t)}$ "adjusting" the not null coefficients, i.e., those corresponding to elements $\hat{\alpha}_j > \sigma_t$ which have indexes in the set $\Lambda_c^{(t)} = \{j : |\alpha_j^{(t)}| > \sigma_t\}$.

- Based upon this strategy, the method should ideally force the σ_t values in such a way to have

$$\lim_{t\to+\infty} g_{\sigma_t}(|\alpha_j|) = 1 - e^{-|\alpha_j^{(t)}|/\sigma_t} = \begin{cases} 1, & \text{if } j \in \Lambda^{(t)} \\ 0, & \text{if } j \in \Lambda_c^{(t)} \end{cases}.$$

The k-LiMapS Algorithm: Convergence Issue

- The LiMapS algorithm computes fixed-points of iterated nonlinear functions for sparse recovery that are solutions of the exact (without noise) model $\Phi\alpha = s$.
- The noisy case described by (1) can be recast as exact model for the noisy signal $\bar{s} = s - \varepsilon$.
- for $t > 0$ holds $Q\alpha^{(t)} = \Phi^{\dagger}\bar{s}$
- the present iterative system has dynamics described by motion equation

$$\alpha^{(t+1)} - \alpha^{(t)} = P\left[\alpha^{(t)} \odot e^{-|\alpha^{(t)}|/\sigma_t}\right]. \quad (4)$$

$$\lim_{t\to+\infty} \left\| \alpha^{(t+1)} - \alpha^{(t)} \right\| = 0$$

$$\left\| P\left[\alpha^{(t)} \odot e^{-|\alpha^{(t)}|/\sigma_t}\right] \right\| \to 0 \quad \text{as } t \to +\infty. \quad (5)$$

The k-LiMapS Algorithm: Convergence Issue

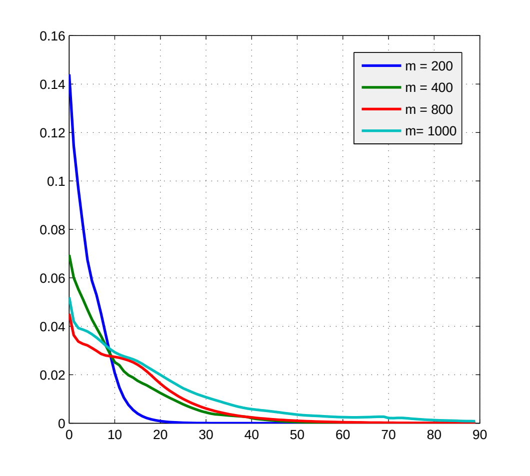

Figure: Plotting of the norm in (5) with sparsity $k = 10$, size $n = 100$ and $m = 200, 400, 800, 1000$.

The k-LiMapS Algorithm: Pseudocode

Algorithm 1 k-LiMapS

Require:
- a dictionary $\Phi \in \mathbb{R}^{n \times m}$
- its pseudo-inverse $\Phi^\dagger$
- a signal $s \in \mathbb{R}^n$
- a sparsity level k

$\alpha \leftarrow \Phi^\dagger s$
while [cond] **do**
  $\sigma \leftarrow \text{sort}\,(|\alpha|)$ *<descending order coefficients>*
  $\lambda \leftarrow 1/\sigma_k$ *<sparsity ratio update>*
  $\beta \leftarrow f_\lambda(\alpha)$ *<increase sparsity>*
  $\alpha \leftarrow \beta - \Phi^\dagger(\Phi\beta - s)$ *<orthogonal projection>*
end while
$\alpha_j \leftarrow 0 \quad \forall\, j \text{ s.t. } |\alpha_j| \leq \sigma_k$ *<thresholding>*

Ensure: an approx. solution of $s = \Phi\alpha$ s.t. $\|\,\alpha\,\|_0 \leq k$

The k-LiMapS Algorithm: The α Coefficients

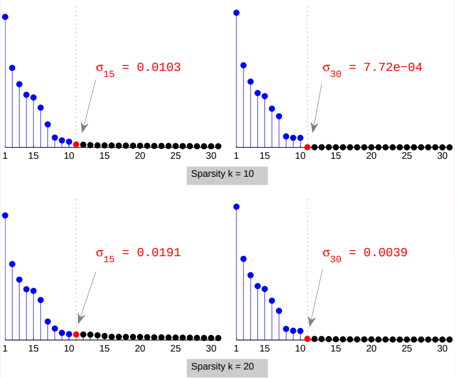

Figure: Sorted absolute values of the α coefficients

Empirical Results

- $\Phi \in \mathbb{R}^{n \times m}$ matrices have been sampled from the uniform spherical ensemble
- signals s was a single realization of a random variable having k nonzeros sampled from a standard iid $\mathcal{N}(0,1)$ distribution
- performances of the algorithms on each realization are measured by mean square error:

$$\text{MSE} = \frac{\| \Phi\alpha - s \|^2}{n}$$

Empirical Results: Comparison on Random Instances

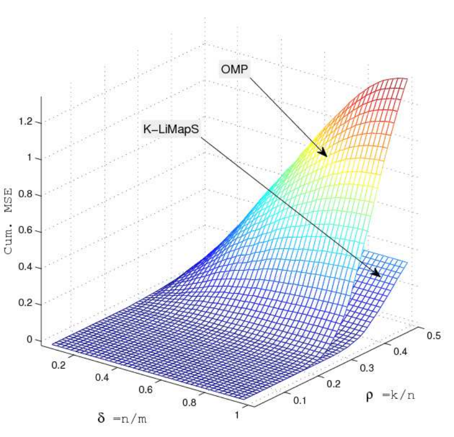

Figure: MSE between the original s and reconstructed $\Phi\alpha$ signals, over 100 trials. $n = 100$, $m = [101 - 1000]$ and $k = [1 - 50]$

Empirical Results: Comparison on ECG Dictionary Learning

- we focus on the dictionary learning task for sparse representation applied to ECG signals
- Instances are taken from the Physionet bank PhysioNet, specifically in the class of normal sinus rhythm
- the signals s was taken of length $n = 128$, each one corresponding to a second of the entire signal registration
- to perform the dictionary learning task we use KSVD and MOD techniques working in conjunction with both the pursuit algorithm OMP and our nonlinear method k-LIMAPS as sparsity recovery algorithms

Empirical Results: Measures

- To evaluate the accuracy of the signal reconstruction, one of the most used performance measure in the ECG signal processing field is the root mean square difference or PRD, together with its normalized version PRDN (which does not depend on the signal mean), defined respectively as:

$$\text{PRD} = 100 * \frac{\|s - \hat{s}\|_2}{\|s\|_2} \quad \text{and} \quad \text{PRDN} = 100 * \frac{\|s - \hat{s}\|_2}{\|s - \bar{s}\|_2},$$

Table: PRD over 5000 test signals.

	PRD mean (%)	PRD std. dev.
KSVD-LiMapS	15.86	5.26
MOD-LiMapS	16.16	5.05
KSVD-OMP	17.92	5.13
MOD-OMP	17.41	4.93

Table: PRDN over 5000 test signals.

	PRDN mean (%)	PRDN std. dev.
KSVD-LiMapS	16.17	5.26
MOD-LiMapS	15.86	5.05
KSVD-OMP	17.92	5.13
MOD-OMP	17.42	4.92

Empirical Results: Comparison on ECG Dictionary Learning

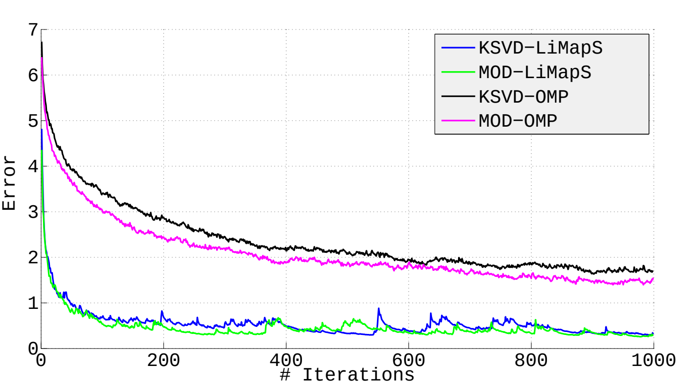

Figure: Mean square error over the training set during each iteration of the learning process.

Empirical Results: Comparison on ECG Dictionary Learning

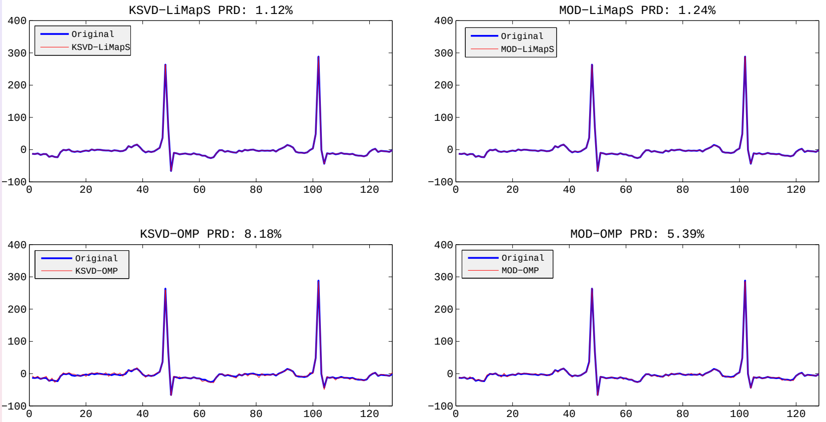

Figure: Original and reconstructed signal from learned dictionaries.

Conclusions and Future Works

- The proposed heuristic, derived by a version devised for ideal noiseless signal which consists on a fixed-point iteration scheme which alternates the application of a suitable nonlinear mapping to the points of the affine space associated to the undetermined system.
- In the experimental section we showed the ability of our algorithm to approximately solve sparse recovery problem when the level of sparsity required is fixed in advance.
- This method can be used efficently in data compression, classification and dictionary learning problems (Work in progress in our laboratory)

Thanks