A multilevel approach for $L_1$ penalized least-squares minimization

Eran Treister and Irad Yavneh

Department of computer science, Technion, Israel.

**Introduction**

Sparse modeling of signals:
An emerging area of research that is drawing vast interest and finding use in numerous applications.
The key observation:
Natural signals, such as images, admit sparse decompositions over specific spatial transforms.

**Applications**

- Denoising
- Super resolution
- Deblurring
- Inpainting
- Also - compression, signal separation, compressed sensing & many more...

**Problem Formulation**

$L_1$ penalized least-squares minimization:

$$
\min_{x \in \mathbb{R}^m} F(x) = \min_{x \in \mathbb{R}^m} \frac{1}{2} ||Ax - y||_2^2 + \mu ||x||_1,
$$

with $\mu$ balancing between sparsity and adherence to the data.

**Iterated Shrinkage “Relaxations”**

Standard optimization tools struggle.
“Iterated shrinkage” methods are used.

- **Coordinate descent (CD):**
  Optimizes each scalar $x_i$ in turn w.r.t $F$.
- **Parallel coordinate descent (PCD):**
  Applies the CD update simultaneously to all $x_i$'s.

Accelerations to PCD: SESOP, non-linear CG.

**Numerical Results**

Synthetic denoising experiment:

- $A$ — random and highly ill-conditioned.
- $y$ — combination of several columns + noise.

Convergence history: $|F(x) - F_{\text{min}}|$

Each work unit stands for $mn$ floating point operations (cost of computing $A^T y$).

Conclusions:
ML — very promising.
Complements existing acceleration methods.

**Motivation**

Shrinkage methods
Used intensively when processing videos, images or training new dictionaries.

The drawback:
These methods tend to be slow for some real-world ill-conditioned dictionaries.

The need:
Sparse modelling must be accelerated.

Our multilevel approach:
(1) Relax — Iterated Shrinkage.
(2) Restric — reduce column dimension.
(3) Solve smaller problem (recursively).
(4) Prolong — restore column dimension.
(5) Relax — Iterated Shrinkage.

**Theoretical Results**

Monotonicity:
By definition, reducing $F^c$ implies reduction of $F$.

Direct solution (two-level):
x$^*$ — the solution. If $C \supseteq \text{supp}(x^*)$, then the solution is achieved in one cycle.

No stagnation (two-level):
If $C \not\supseteq \text{supp}(x^*)$, then a single post-relaxation inserts at least one atom to supp(x).

Complementary roles:
Relaxation — finds the true support while the reduced-level finds the non-zero values in x.

$A$ — selection Guarantee:
Let $i$ be s.t $i \in \text{supp}(x^*)$ and $i \not\in \text{supp}(x)$. Then $i$ is chosen to $C$ if

$$
|\tilde{x}_i^*| \geq \frac{2\delta}{1 + \delta} ||x^* - x||_1; \quad \delta = \max_{i \neq j} \{|a_i^T a_j|\}.
$$

**Reducing Dimension**

The main idea: $A_c$, submatrix of $A$
$x$ is sparse — many columns are unnecessary.

Choice of $m_c$ columns:
$\text{supp}(x)+$ likely to enter the support.

**Initialization**

We use sequential dictionary expansion, starting from $x = 0$.

Support remains small throughout the iterations.

**Multilevel Algorithm - The V cycle**

Reduced level problem:

$$
\min_{x \in \mathbb{R}^m} F^c(x) = \min_{x \in \mathbb{R}^m} \frac{1}{2} ||A_c x - y||_2^2 + \mu ||x||_1.
$$

**Contact Information**

eran@cs.technion.ac.il