Sparse & Redundant Representations by Iterated-Shrinkage Algorithms

Michael Elad*
The Computer Science Department
The Technion – Israel Institute of technology
Haifa 32000, Israel

26 - 30 August 2007
San Diego Convention Center
San Diego, CA USA

* Joint work with: Boaz Matalon, Joseph Shtok, Michael Zibulevsky
Today’s Talk is About

the Minimization of the Function

\[ f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) \]

by Iterated-Shrinkage Algorithms

Today we will discuss:

- Why this minimization task is important?
- Which applications could benefit from this minimization?
- How can it be minimized effectively?
- What iterated shrinkage methods are there? and...
1. **Motivating the Minimization of** $f(\alpha)$
   Describing various applications that need this minimization

2. **Some Motivating Facts**
   General purpose optim.

3. **Iterated-Shrinkage**
   We describe five versions.

4. **Some Results**
   Image deblurring results

5. **Conclusions**

---

\[ \hat{\alpha} = \text{ArgMin}_{\alpha} \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) \]
Many of the existing **image denoising** algorithms are related to the minimization of an energy function of the form

\[
    f(x) = \frac{1}{2} \left\| x - y \right\|_2^2 + \Pr(x)
\]

- \( y \): Given measurements
- \( x \): Unknown to be recovered

We will use a **Sparse & Redundant Representation** prior.
Our MAP Energy Function

- We assume that $x$ is created by $M$:
  where $\alpha$ is a **sparse** & **redundant** representation and $D$ is a known dictionary.

- This leads to:
  $$\hat{\alpha} = \text{ArgMin}_{\alpha} \frac{1}{2} \| x - y \|_2^2$$

$$\hat{\alpha} = \text{ArgMin}_{\alpha} \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha)$$

This is Our Problem !!!

- The $L_p$ norm ($\| \cdot \|_p$) with $0 < p \leq 1$ is often found to be equivalent.

- Many other **ADDITIVE** sparsity measures are possible.
General (linear) Inverse Problems

- Assume that $x$ is known to emerge from $\mathcal{M}$, as before.
- Suppose we observe $y = Hx + v$, a “blurred” and noisy version of $x$. How could we recover $x$?
- A MAP estimator leads to:

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{2} \| H D \alpha - y \|_2^2 + \lambda \rho(\alpha)$$

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{2} \| D \alpha - x \|_2^2 + \lambda \rho(\alpha)$$

This is Our Problem !!!
Inverse Problems of Interest

- De-Noising
- De-Blurring
- In-Painting
- De-Mosaicing
- Tomography
- Image Scale-Up & super-resolution
- And more ...

$$\hat{\alpha} = \text{ArgMin}_{\alpha} \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha)$$

This is Our Problem !!!
Signal Separation

Given a mixture \( z = x_1 + x_2 + v \) — two sources, \( M_1 \) and \( M_2 \), and white Gaussian noise \( v \), we desire to separate it to its ingredients.

Written differently:

Thus, solving this problem using MAP leads to the Morphological Component Analysis (MCA) [Starck, Elad, Donoho, 2005]:

\[
\hat{\alpha} = \text{ArgMin}_{\alpha} \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha)
\]

This is Our Problem !!!
In compressed-sensing we compress the signal \( y \) by exploiting its origin. This is done by \( p \ll n \) random projections.

The core idea: \( y \approx P \alpha \) holds all the information about the original signal \( x \), even though \( p \ll n \).

Reconstruction? Use MAP again and solve:

\[
\hat{\alpha} = \text{ArgMin}_{\alpha} \frac{1}{2} \| P \alpha - x \|_2^2 + \lambda \rho(\alpha)
\]

This is Our Problem !!!

\[
\hat{x} = D \hat{\alpha}
\]
The minimization of the function

\[ f(\alpha) = \frac{1}{2} \| D \alpha - x \|_2^2 + \lambda \rho(\alpha) \]

is a worthy task, serving many & various applications.

So, How This Should be Done?
A Wide-Angle View Of Iterated-Shrinkage Algorithms
By: Michael Elad, Technion, Israel

1. Motivating the Minimization of $f(\alpha)$
   Describing various applications that need this minimization

2. Some Motivating facts
   General purpose optimization tools, and the unitary case

3. Iterated-Shrinkage Algorithms
   We describe five versions of those in detail

4. Some Results
   Image deblurring results

5. Conclusions
Is there a Problem?

\[ f(\alpha) = \frac{1}{2} \left\| D\alpha - x \right\|_2^2 + \lambda \rho(\alpha) \]

- The first thought: With all the existing knowledge in optimization, we could find a solution.

- Methods to consider:
  - Normalized Steepest Descent: compute the gradient and follow its path.
  - Conjugate Gradient: use the gradient and the previous update direction, combined by a preset formula.
  - Pre-Conditioned SD: weight the gradient by the Hessian's diagonal inverse.
  - Truncated Newton: Use the gradient and Hessian to define a linear system, and solve it approximately by a set of CG steps.
  - Interior-Point Algorithms: Separate to positive and negative entries, and use both the primal and the dual problems + barrier for forcing positivity.
General-Purpose Software?

- **A Problem:** General purpose software-packages (algorithms) are typically performing poorly on our task.
  - The fact that the solution is expected to be sparse (or nearly so) in our problem is not exploited in such algorithms.
  - The Hessian of \( f(\alpha) \) tends to be highly ill-conditioned near the (sparse) solution.

- **So, are we stuck? Is this problem really that complicated?**

\[
\nabla f(\alpha) = D^H(D\alpha - x) + \lambda \rho'(\alpha)
\]

\[
\nabla^2 f(\alpha) = D^HD + \lambda \rho''(\alpha)
\]
Consider the Unitary Case \((DD^H = I)\)

Given a separable set of \(m\) identical 1D optimization problems:

\[
f(\alpha) = \frac{1}{2} \|D\alpha - x\|_2^2 + \lambda \rho(\alpha)
\]

Use \(DD^H = I\)

Define \(D^H x = \beta\)

\[
f(\alpha) = \frac{1}{2} \|D\alpha - D \beta\|_2^2 + \lambda \rho(\alpha)
\]

\(L_2\) is unitarily invariant

\[
f(\alpha) = \frac{1}{2} \|\alpha - \beta\|_2^2 + \lambda \rho(\alpha) = m \sum_{j=1}^{m} \left[ \frac{1}{2} (\alpha_j - \beta_j)^2 + \lambda \rho(\alpha_j) \right]
\]
We need to solve the following 1D problem:

$$\alpha_{\text{opt}} = \underset{\alpha}{\text{ArgMin}} \left\{ \frac{1}{2} (\alpha - \beta)^2 + \lambda \rho(\alpha) \right\}$$

Such a Look-Up-Table (LUT) $\alpha_{\text{opt}} = S_{\rho, \lambda}(\beta)$ can be built for ANY sparsity measure function $\rho(\alpha)$, including non-convex ones and non-smooth ones (e.g., $L_0$ norm), giving in all cases the GLOBAL minimizer of $g(\alpha)$. 
The Unitary Case: A Summary

Minimizing \( f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) = \frac{1}{2} \| \alpha - \beta \|_2^2 + \lambda \rho(\alpha) \)

is done by:

\[ D^H x = \beta \]

The obtained solution is the GLOBAL minimizer of \( f(\alpha) \), even if \( f(\alpha) \) is non-convex.
The minimization of

\[ f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) \]

Leads to two very **Contradicting Observations:**

1. The problem is **quite hard** – classic optimization find it hard.
2. The problem is **trivial** for the case of unitary \( D \).

**How Can We Enjoy This Simplicity in the General Case?**
1. Motivating the Minimization of $f(\alpha)$
Describing various applications that need this minimization.

2. Some Motivating Facts
General purpose optimization tools, and the unitary case.

3. Iterated-Shrinkage Algorithms
We describe five versions of those in detail.

4. Some Results
Image deblurring results.

5. Conclusions
Iterated-Shrinkage Algorithms?

We will present THE PRINCIPLES of several leading methods:

- Bound-Optimization and EM [Figueiredo & Nowak, `03],
- Surrogate-Separable-Function (SSF) [Daubechies, Defrise, & De-Mol, `04],
- Parallel-Coordinate-Descent (PCD) algorithm [Elad `05], [Matalon, et.al. `06],
- IRLS-based algorithm [Adeyemi & Davies, `06], and
- Stepwise-Ortho-Matching Pursuit (StOMP) [Donoho et.al. `07].

Common to all is a set of operations in every iteration that includes:
(i) Multiplication by $D$,
(ii) Multiplication by $D^H$, and
(iii) A Scalar shrinkage on the solution $S_{\rho,\lambda}(\alpha)$.

Some of these algorithms pose a direct generalization of the unitary case, their 1\textsuperscript{st} iteration is the solver we have seen.
1. The Proximal-Point Method

- Aim: minimize $f(\alpha)$ – Suppose it is found to be too hard.
- Define a surrogate-function $g(\alpha, \alpha_0) = f(\alpha) + \text{dist}(\alpha - \alpha_0)$, using a general (uni-modal, non-negative) distance function.
- Then, the following algorithm necessarily converges to a local minima of $f(\alpha)$ [Rokafellar, ’76]:

  $\alpha_0 \rightarrow \text{Minimize } g(\alpha, \alpha_0) \rightarrow \alpha_1 \rightarrow \text{Minimize } g(\alpha, \alpha_1) \rightarrow \alpha_2 \rightarrow \ldots \rightarrow \alpha_k \rightarrow \text{Minimize } g(\alpha, \alpha_k) \rightarrow \alpha_{k+1}$

- Comments: (i) Is the minimization of $g(\alpha, \alpha_0)$ easier? It better be!
  (ii) Looks like it will slow-down convergence. Really?
The Proposed Surrogate-Functions

- Our original function is: \( f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) \)

- The distance to use: \( \text{dist}(\alpha, \alpha_0) = \frac{c}{2} \cdot \| \alpha - \alpha_0 \|_2^2 - \frac{1}{2} \| D\alpha - D\alpha_0 \|_2^2 \)

  Proposed by [Daubechies, Defrise, & De-Mol `04]. Require \( c > r(D^H D) \).

- The beauty in this choice: the term \( \| D\alpha \|_2^2 \) vanishes.

\[
g(\alpha, \alpha_0) = \lambda \rho(\alpha) + \frac{c}{2} \| \alpha \|_2^2 - \alpha^H \beta_0 \quad \text{where} \quad \beta_0 = D^H (x - D\alpha_0) + c\alpha_0
\]

- Minimization of \( g(\alpha, \alpha_0) \) is done in a closed form by shrinkage, done on the vector \( \beta_0 \) and this generates the solution \( \alpha_k \) of the next iteration.

- It is a separable sum of m 1D problems. Thus, we have a closed form solution by THE SAME SHRINKAGE!!
The Resulting SSF Algorithm

While the Unitary case solution is given by

\[ \hat{\alpha} = S_{\rho, \lambda}(D^H x) \; ; \; \hat{x} = D\hat{\alpha} \]

the general case, by SSF requires:

\[ \alpha_{k+1} = S_{\rho, \lambda} \left( \frac{1}{c} D^H (x - D\alpha_k) + \alpha_k \right) \]

\[ \hat{x} \]

A Wide-Angle View Of Iterated-Shrinkage Algorithms
By: Michael Elad, Technion, Israel
2. Bound-Optimization Technique

- Aim: minimize $f(\alpha)$ – Suppose it is found to be too hard.

- Define a function $Q(\alpha,\alpha_0)$ that satisfies the following conditions:
  - $Q(\alpha_0,\alpha_0)=f(\alpha_0)$,
  - $Q(\alpha,\alpha_0) \geq f(\alpha)$ for all $\alpha$, and
  - $\nabla Q(\alpha,\alpha_0) = \nabla f(\alpha)$ at $\alpha_0$.

- Then, the following algorithm necessarily converges to a local minima of $f(\alpha)$ [Hunter & Lange, (Review)`04]:

\[
\begin{align*}
\alpha_0 & \quad \text{Minimize} \quad Q(\alpha,\alpha_0) \\
\alpha_1 & \quad \text{Minimize} \quad Q(\alpha,\alpha_1) \\
\alpha_2 & \quad \text{Minimize} \quad Q(\alpha,\alpha_k) \\
\vdots & \\
\alpha_k & \quad \text{Minimize} \quad Q(\alpha,\alpha_{k+1})
\end{align*}
\]
3. Start With Coordinate Descent

- We aim to minimize \( f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) \).

- First, consider the Coordinate Descent (CD) algorithm.

- This is a 1D minimization problem:

- It has a closed form solution, using a simple **Shrinkage** as before, applied on the scalar \(<e_j, d_j>\).

- \( \alpha_{\text{opt}} = \text{ArgMin}_{\alpha} \left( \frac{1}{2} \| \alpha + d_j \|_2^2 \right) + \lambda \rho(\alpha_j) \).
Parallel Coordinate Descent (PCD)

Current solution for minimization of $f(\alpha)$

$$\{v_j\}_{j=1}^{m} : \text{Descent directions obtained by the previous CD algorithm}$$

- We will take the sum of these $m$ descent directions for the update step.
- Line search is mandatory.
- This leads to \ldots\ldots\ldots
The PCD Algorithm

\[ \alpha_{k+1} = \alpha_k + \mu \left[ S_{\rho, \lambda} \left( QD^H(x - D\alpha_k) + \alpha_k \right) - \alpha_k \right] \]

Where \( Q = \text{diag}^{-1}(D^HD) \) and \( \mu \) represents a line search (LS).

Note: \( Q \) can be computed quite easily off-line. Its storage is just like storing the vector \( \alpha_k \).
Surprising as it may sound, these very effective acceleration methods can be implemented with no additional "cost" (i.e., multiplications by $D$ or $D^T$).

Option 1 – Use $v$ as is:
\[
\alpha_{k+1} = \alpha_k + \begin{bmatrix}
\alpha_{k-M+1} - \alpha_k & \cdots & \alpha_{k-2} - \alpha_k & \alpha_k - \alpha_{k-1} & v - \alpha_k \\
\mu_M \\
\vdots \\
\mu_0
\end{bmatrix}
\]

Option 2 – Use line-search with $v$:
\[
x = (x - D\alpha_k) + \alpha_k
\]

Option 3 – Use Sequential Subspace Optimization (SESOP):
\[
\begin{bmatrix}
\mu_M \\
\vdots \\
\mu_0
\end{bmatrix}
\]

[Zibulevsky & Narkis, '05]
[Elad, Matalon, & Zibulevsky, '07]
For an effective minimization of the function

$$f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha)$$

we saw several iterated-shrinkage algorithms, built using

1. Proximal Point Method
2. Bound Optimization
3. Parallel Coordinate Descent
4. Iterative Reweighed LS
5. Fixed Point Iteration
6. Greedy Algorithms

How Are They Performing?
1. Motivating the Minimization of $f(\alpha)$
   Describing various applications that need this minimization

2. Some Motivating Facts
   General purpose optimizers

3. Iterated-Shrinkage Algorithms
   We describe five versions.

4. Some Results
   Image deblurring results

5. Conclusions
A Deblurring Experiment

15×15 kernel
\((n^2 + m^2 + 1)^{1/2}\)
\(-7 \leq n, m \leq 7\)

White Gaussian Noise \(\sigma^2=2\)

\[
\hat{x} = \arg\min_{x} \frac{1}{2} \| Hx - y \|_2^2 + \lambda \rho(x)
\]

\[
\hat{x} = D\hat{\alpha}
\]
A given blurred and noisy image of size 256×256

2D un-decimated Haar wavelet transform, 3 resolution layers, 7:1 redundancy

The blur operator

\[ f(\alpha) = \frac{1}{\|HD\alpha - V\|^2} + \lambda \rho(\alpha) \]

\[ \rho(\alpha) = |\alpha| - S \cdot \log \left( 1 + \frac{|\alpha|}{S} \right) \]

\[ \lambda = 0.075 \]

Note: This experiment is similar (but not equivalent) to one of tests done in [Figueiredo & Nowak `05], that leads to state-of-the-art results.
So, The Results: The Function Value

The graph shows the function value $f(\alpha) - f_{\text{min}}$ over iterations/computations. The y-axis represents the function value on a logarithmic scale ranging from $10^2$ to $10^9$, and the x-axis represents the iterations/computations ranging from 0 to 50.

Three lines are plotted:
- Blue line: SSF
- Red line: SSF-LS
- Black line: SSF-SESOP-5

The graph indicates the convergence of the algorithms over iterations, with SSF-SESOP-5 showing the fastest convergence to the minimum function value.
So, The Results: The Function Value

Comment:

Both SSF and PCD (and their accelerated versions) are provably converging to the minima of $f(\alpha)$. 
So, The Results: The Function Value

\[ f(\alpha) - f_{\text{min}} \]

Iterations/Computations

A Wide-Angle View Of Iterated-Shrinkage Algorithms
By: Michael Elad, Technion, Israel
So, The Results: ISNR

\[ \text{ISNR} = 10 \log_{10} \left( \frac{\| y - x_0 \|_2^2}{\| D^\alpha_k - x_0 \|_2^2} \right) \]

- SSF
- SSF-LS
- SSF-SESOP-5

6.41 dB
So, The Results: ISNR

7.03dB
So, The Results: ISNR

Comments:

StOMP is inferior in speed and final quality (ISNR=5.91dB) due to over-estimated support.

PDCO is very slow due to the numerous inner Least-Squares iterations done by CG. It is not competitive with the Iterated-Shrinkage methods.
Visual Results

PCD-SESOP-5 Results:

original (left), Measured (middle), and Restored (right): Iteration 0, ISNR = -16.7728 dB

original (left), Measured (middle), and Restored (right): Iteration 1, ISNR = 0.069583 dB

original (left), Measured (middle), and Restored (right): Iteration 2, ISNR = 2.46924 dB

original (left), Measured (middle), and Restored (right): Iteration 3, ISNR = 4.1824 dB

original (left), Measured (middle), and Restored (right): Iteration 4, ISNR = 4.9726 dB

original (left), Measured (middle), and Restored (right): Iteration 5, ISNR = 5.5875 dB

original (left), Measured (middle), and Restored (right): Iteration 6, ISNR = 6.2188 dB

original (left), Measured (middle), and Restored (right): Iteration 7, ISNR = 6.6479 dB

original (left), Measured (middle), and Restored (right): Iteration 8, ISNR = 6.6789 dB

original (left), Measured (middle), and Restored (right): Iteration 12, ISNR = 6.9416 dB

original (left), Measured (middle), and Restored (right): Iteration 19, ISNR = 7.0322 dB
1. Motivating the Minimization of $f(\alpha)$
   Describing various applications that need this minimization

2. Some Motivating Facts
   General purpose optimization tools, and the unitary case

3. Iterated-Shrinkage Algorithms
   We describe five versions of those in detail

4. Some Results
   Image deblurring results

5. Conclusions
Conclusions - The Bottom Line

If your work leads you to the need to minimize the problem:

\[ f(\alpha) = \frac{1}{2} \| D\alpha - x \|_2^2 + \lambda \rho(\alpha) \]

Then:

- We recommend you use an **Iterated-Shrinkage** algorithm.
- **SSF and PCD are Preferred**: both are provably converging to the (local) minima of \( f(\alpha) \), and their performance is very good, getting a reasonable result in few iterations.
- Use **SESOP Acceleration** – it is very effective, and with hardly any cost.
- There is **Room** for more work on various aspects of these algorithms – see the accompanying paper.
Thank You for Your Time & Attention

This field of research is very hot ...

More information, including these slides and the accompanying paper, can be found on my web-page
http://www.cs.technion.ac.il/~elad

THE END !!
3. The IRLS-Based Algorithm

- Use the following principles [Edeyemi & Davies `06]:

  1. Iterative Reweighed Least-Squares (IRLS)

  \[ f(\alpha) = \frac{1}{2} \| D\alpha - x \|^2 + \lambda \rho(\alpha) \]

  \[ \rho(\alpha) = \alpha^H W(\alpha) \alpha \]

  This is the IRLS algorithm, used in FOCUSS [Gorodinsky & Rao `06].

\[
\begin{bmatrix}
\frac{\rho(\alpha_1)}{\alpha_1^2} \\
\frac{\rho(\alpha_j)}{\alpha_j^2} \\
\frac{\rho(\alpha_m)}{\alpha_m^2}
\end{bmatrix} = 0
\]
The IRLS-Based Algorithm

- Use the following principles [Edeyemi & Davies `06]:
  - (2) Fixed-Point Iteration

\[- \mathbf{D}^H (\mathbf{x} - \mathbf{D}\alpha_k) + 2\lambda \mathbf{W}(\alpha_k)\alpha + c\alpha - c\alpha = 0\]

Task: solve the system \(\Phi(\alpha) - \alpha = 0\)

\[\alpha_{k+1} = \left(\frac{2\lambda}{c} \mathbf{W}(\alpha_k) + \mathbf{I}\right)^{-1} \left(\frac{1}{c} \mathbf{D}^H (\mathbf{x} - \mathbf{D}\alpha_k) + \alpha_k\right)\]

Notes: (1) For convergence, we should require \(c > \frac{\text{tr}(\mathbf{D}^H\mathbf{D})}{2}\).
(2) This algorithm cannot guarantee local-minimum.
4. Stagewise-OMP [Donoho, Drori, Starck, & Tsaig, `07]

- StOMP is originally designed to solve
  \[ \min_{\alpha} \frac{1}{2} \| D\alpha - x \|^2 + \lambda \| \alpha \|_0 \]
  and especially so for random dictionary (Compressed-Sensing).

- Nevertheless, it is used elsewhere (restoration) [Fadili & Starck, `06].

- If S grows by one item at each iteration, this becomes OMP.

- LS uses \( K_0 \) CG steps, each equivalent to 1 iterated-shrinkage step.
So, The Results: The Function Value

The results show the function value $f(\alpha) - f_{\text{min}}$ as a function of iterations. The graph compares different algorithms:

- SSF
- SSF-LS
- SSF-SESOP-5
- IRLS
- IRLS-LS

The graph displays the convergence of these algorithms over iterations, with the $y$-axis representing the function value difference and the $x$-axis representing the number of iterations.
So, The Results: ISNR

ISNR [dB]

Iteration

SSF
SSF-LS
SSF-SESOP-5
IRLS
IRLS-LS

A Wide-Angle View Of
Iterated-Shrinkage Algorithms
By: Michael Elad, Technion, Israel