Grouped Coordinate Descent Algorithms for Robust Edge-Preserving Image Restoration

Jeffrey A. Fessler

EECS Department
The University of Michigan

SPIE’97: Image Reconstruction and Restoration

July 29, 1997
OUTLINE

- Problem Description
- Huber Algorithm
- Optimization Transfer
- Convex Algorithm (ala Lange / De Pierro)
- Grouped Coordinate Descent (GCD) Algorithm
- Anecdotal Results
- Summary
“LINEAR” INVERSE PROBLEM

\[ \underline{y} = \underline{A} \underline{x} + \text{noise} \]

- \( \underline{y} \): noisy measurements (blurred image or sinogram)
- \( \underline{x} \): unknown object (true image)
- \( \underline{A} \): known system model
  (each column is a point response function)
- Errors in \( \underline{A} \) partially motivate robust methods

Goal: recover an estimate \( \hat{\underline{x}} \) of \( \underline{x} \) from \( \underline{y} \).
DATA-FIT COST FUNCTION

Want $\hat{x}$ to “fit the data,” i.e. $y \approx A\hat{x}$

Natural cost function for independent measurement errors:

$$\Phi^{\text{data}}(x) = \sum_{i=1}^{m_1} \psi_i^{\text{data}}([y - Ax]_i)$$

- $[y - Ax]_i = y_i - \sum_{j=1}^{p} a_{ij} x_j$
- $m_1$: length of $y$
- $\psi_i$: convex function.

Traditional choice: $\psi_i(t) = t^2/2$, which is appropriate for Gaussian noise, but is not robust to noise with heavy-tailed distributions.
Example - Huber function:

\[ \psi(t) = \begin{cases} 
\frac{t^2}{2}, & |t| \leq \delta, \\
\delta |t| - \frac{\delta^2}{2}, & |t| > \delta 
\end{cases} \]
ROBUST ESTIMATORS

Generalized-Gaussian family of pdfs with unit variance:
\[ f_X(x; \mu, p) = \frac{p}{2 \Gamma(1/p)} \sqrt{r_p} \exp\left( -|x - \mu|^p r_p^{p/2} \right) \] where \( r_p = \frac{\Gamma(3/p)}{\Gamma(1/p)} \).

Asymptotic variance of the sample median estimator for \( \mu \) is:
\[ \frac{1}{4n f^2(\mu)} = \frac{1}{n} \frac{\Gamma^2(1/p)}{p^2 r_p} \] (cf \( 1/n \) for the sample mean).

CR bound for estimating \( \mu \): \( \sigma_{\mu}^2 \geq \frac{1}{n p^2 r_p} \frac{\Gamma(1/p)}{\Gamma(2 - 1/p)} \).
Minimizing $\Phi^{\text{data}}$ is inadequate for ill-conditioned inverse problems.

Prior “knowledge” of piece-wise smoothness:

- $x_j - x_{j-1} \approx 0$ (piece-wise constant)
- $x_{j-1} - 2x_j + x_{j+1} \approx 0$ (piece-wise linear)
- $x_j \approx 0$ (support constraints)
- ... Combining: $Cx \approx z$

Regularized cost function:

$$\Phi(x) = \Phi^{\text{data}}(x) + \Phi^{\text{penalty}}(x),$$

$$\Phi^{\text{penalty}}(x) = \sum_{i=1}^{m_2} \psi_i^{\text{penalty}}([Cx - z]_i)$$
EXAMPLE: ROUGHNESS PENALTY
(AKA GIBBS PRIOR)

\[ D_n = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad C = \begin{bmatrix} I_{ny} \otimes D_{nx} \\ D_{ny} \otimes I_{nx} \end{bmatrix} \]

where \( \otimes \) denotes the Kronecker matrix product.

If \( z = 0 \) and \( N_j \) is the four pixel neighborhood of pixel \( j \), then

\[ \Phi^\text{penalty}(x) = \sum_j \sum_{k \in N_j} \psi_{j,k}(x_j - x_k) \]

Conventional (Tikhonov-Miller) regularization: \( \psi(t) = t^2/2 \).

(Gaussian prior)

For edge-preserving image recovery, need non-quadratic \( \psi(\cdot) \), such as Huber function.
UNIFIED COST FUNCTION

\[
\Phi(x) = \sum_{i=1}^{m} \psi_i([Bx - c_i])
\]

Regularized edge-preserving cost function is a special case:

\[
\Phi(x) = \Phi^{\text{data}}(x) + \Phi^{\text{penalty}}(x), \quad B = \begin{bmatrix} A \\ C \end{bmatrix}, \quad c = \begin{bmatrix} y \\ z \end{bmatrix}
\]

Optimization problem:

\[
\hat{x} = \arg\min_{x} \Phi(x) \quad \text{or} \quad \hat{x} = \arg\min_{x \geq 0} \Phi(x).
\]
OPTIMIZATION

Simple in quadratic case where \( \psi_i(t) = t^2/2 \ \forall i \)

\[
\hat{\mathbf{x}} = (\mathbf{B}'\mathbf{B})^{-1} \mathbf{B}'\mathbf{c}
\]

Good algorithms:
- Preconditioned conjugate gradients
- Coordinate descent (Gauss-Siedel)

Challenging for non-quadratic \( \psi_i \)'s
Very challenging for non-convex \( \psi_i \)'s

Proposition: algorithms tailored to structure of \( \Phi \) can outperform general purpose optimization methods.

but cannot solve it all...
ASSUMPTIONS

\( B \) has full column rank, so \( M > 0 \Rightarrow B'MB > 0 \)
(Easily achieved with sensible regularization design)

- \( \psi \) is symmetric
- \( \psi \) is everywhere differentiable (and therefore continuous)
- \( \dot{\psi}(t) = \frac{d}{dt} \psi(t) \) is non-decreasing (and hence \( \psi \) is convex)
- \( \omega_\psi(t) = \frac{\dot{\psi}(t)}{t} \) is non-increasing for \( t \geq 0 \)
- \( \omega_\psi(0) = \lim_{t \to 0} \frac{\dot{\psi}(t)}{t} \) is finite and nonzero, i.e. \( 0 < \omega_\psi(0) < \infty \)

\( \Phi \) has a unique minimizer
(Easily ensured with perturbation of regularizer)
UNCONSTRAINED SOLUTION

\[ \Phi(x) = \sum_{i=1}^{m} \psi_i([B x - \xi_i]) \]

Column gradient:

\[ \nabla \Phi(x) = B' \Omega(x)(B x - c), \quad \nabla \Phi(x) \bigg|_{x=\hat{x}} = 0 \]

where \( \Omega(x) = \text{diag}\{\omega_i([B x - \xi_i])\} \)

Unconstrained solution:

\[ \hat{x} = [B' \Omega(\hat{x}) B]^{-1} B' \Omega(\hat{x}) c \]

\[ = \arg \min_{\hat{x}} \frac{1}{2} (c - B \hat{x})' \Omega(\hat{x})(c - B \hat{x}) \]

(ala WLS, but weights depend on estimate \( \hat{x} \), hence nonlinear)

Therefore need iterative algorithm...
WEIGHTING FUNCTIONS $\omega_\psi$

![Graph of Weighting Functions](image)

- Quadratic
- Robust / Edge-preserving
NEWTON-RAPHSON ALGORITHM

\[
\mathbf{x}^{n+1} = \mathbf{x}^n - [B' \Lambda(\mathbf{x}^n) B]^{-1} \nabla \Phi(\mathbf{x}^n)
\]

where

\[
\Lambda(\mathbf{x}^n) = \text{diag}\{\dot{\psi}_i([B \mathbf{x} - \mathbf{c}]_i)\}
\]

Advantage:
- Super-linear convergence rate (if convergent)

Disadvantages:
- Requires twice-differentiable \( \psi_i \)'s
- Not guaranteed to converge
- Not guaranteed to monotonically decrease \( \Phi \)
- Does not enforce nonnegativity constraint
- Impractical for image recovery due to matrix inverse

General purpose remedy: bound-constrained Quasi-Newton algorithms
HUBER ALGORITHM (1981)

Recall \( \hat{x} = [B'\Omega(\hat{x})B]^{-1}B'\Omega(\hat{x})c = \hat{x} - [B'\Omega(\hat{x})B]^{-1}\nabla \Phi(\hat{x}) \)

Successive Substitutions:

\[
x^{n+1} = x^n - [B'\Omega(x^n)B]^{-1}\nabla \Phi(x^n)
\]

Advantages:
- Monotonically decreases \( \Phi \)
- Converges globally to unique minimizer (not shown by Huber)

Disadvantages:
- Does not enforce nonnegativity constraint
- Impractical for image recovery due to matrix inverse

Successive substitutions is often not convergent. Why here?
OPTIMIZATION TRANSFER

\[ x^{n+1} = \arg \min_{\bar{x}} \phi^{\text{Huber}}(\bar{x}; x^n) \]

\[ \phi^{\text{Huber}}(\bar{x}; x^n) = \frac{1}{2}(\bar{c} - B\bar{x})^T \Omega(x^n)(\bar{c} - B\bar{x}) \]

Minimizing surrogate function \( \phi \) ensures a monotone decrease in \( \Phi \) if:

- \( \phi(x^n; x^n) = \Phi(x^n) \)
- \( \nabla_x \phi(x; x^n)|_{x=x^n} = \nabla \Phi(x)|_{x=x^n} \)
- \( \Phi(x) \leq \phi(x; x^n) \).

These 3 (sufficient) conditions are satisfied by \( \phi^{\text{Huber}} \).
OPTIMIZATION TRANSFER IN 2D
GENERALIZED HUBER ALGORITHM

\[ x^{n+1} = x^n - M_n^{-1} \nabla \Phi(x^n) \]

where

\[ M_n \geq B' \Omega(x^n) B \]

Advantages:
- Monotonically decreases \( \Phi \)
- Converges globally to unique minimizer
- Can choose \( M_n \) to be easily invertible, e.g. diagonal.
  (Or splitting matrices more generally)

Disadvantages:
- Does not enforce nonnegativity constraint
- Converges slower than Huber algorithm
CONVERGENCE RATE

Low Curvature
Large Steps
Fast Convergence

High Curvature
Small Steps
Slow Convergence

Old   New

x

φ

φ

φ

φ

can we beat this tradeoff?
USING THE STRUCTURE OF $\Phi$

De Pierro’s decomposition (uses form of argument of $\psi_i$):

$$B\mathbf{x} - c = \sum_{j=1}^{p} \alpha_{ij} \left[ \frac{b_{ij}}{\alpha_{ij}} (x_j - x^n_j) + B\mathbf{x}^n - c \right]$$

provided $\alpha_{ij} \geq 0$ and $\sum_{j=1}^{p} \alpha_{ij} = 1$, $\forall i$.

The $\alpha_{ij}$’s are algorithm design factors.
Natural choice is $\alpha_{ij} = |b_{ij}|/\sum_{j=1}^{p} |b_{ik}|$.

By convexity of $\psi_i$:

$$\psi_i([B\mathbf{x} - c]_i) \leq \sum_{j=1}^{p} \alpha_{ij} \psi_i \left( \frac{b_{ij}}{\alpha_{ij}} (x_j - x^n_j) + B\mathbf{x}^n - c \right)$$

Construct surrogate function:

$$\Phi(\mathbf{x}) = \sum_{i=1}^{m} \psi_i([B\mathbf{x} - c]_i) \leq \phi^{LDC}(\mathbf{x}; \mathbf{x}^n)$$

$$\phi^{LDC}(\mathbf{x}; \mathbf{x}^n) = \sum_{j=1}^{p} \phi_j(x_j; \mathbf{x}^n),$$

$$\phi_j(x_j; \mathbf{x}^n) = \sum_{i=1}^{m} \alpha_{ij} \psi_i \left( \frac{b_{ij}}{\alpha_{ij}} (x_j - x^n_j) + B\mathbf{x}^n - c \right)$$

$\phi^{LDC}$ satisfies the 3 conditions for monotonicity.
LANGE / DE PIERRO CONVEX ALGORITHM

\[
\mathbf{x}^{n+1} = \arg \min_{\mathbf{x}} \phi^{\text{LDC}}(\mathbf{x}; \mathbf{x}^n)
\]

\[
x_j^{n+1} = \arg \min_{x_j \geq 0} \phi_j(x_j; \mathbf{x}^n)
= \arg \min_{x_j \geq 0} \sum_{i=1}^{m} \alpha_{ij} \psi_i \left( \frac{b_{ij}}{\alpha_{ij}} (x_j - x_j^n) + B x^n - c \right)
\]

Advantages:
1. Monotonically decreases $\Phi$
2. Converges globally to unique minimizer
3. No matrix inversion required
4. Can enforce nonnegativity constraint
5. Parallelizable (all pixels updated simultaneously)

Disadvantages:
1. Requires subiteration for minimization
   - Solution: use 1-D Huber algorithm
2. Very slow convergence (ala EM algorithm)
   - Solution: update only a subset of the pixels simultaneously
GROUPED COORDINATE DESCENT ALGORITHM

Construct surrogate function using Lange / De Pierro convexity method but for only a (large) subset of the pixels.

<table>
<thead>
<tr>
<th>Pixel Groups (2x3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

Pixels separated => decoupled => fast convergence
Many pixels per subiteration => parallelizable

Retains advantages of Convex Algorithm, but converges faster.

Disadvantages:
- Slightly less parallelizable.
- Slightly more complicated implementation
- Difficult to exploit structure of $B$
  (e.g. FFTs for shift-invariant PSF, separable blur in PET)
SIMULATION EXAMPLE

True object $x$:

With 5 pixel horizontal motion blur and Gaussian noise, $y$ is:
RESTORED IMAGE

Wiener filter:

Edge-preserving restoration \( \hat{x} \):

Huber function used for \( \psi_i \)'s for piece-wise smoothness. 15 iterations of Grouped Coordinate Descent.
CONVERGENCE RATES

NORMALIZED RMS DISTANCE

\[
\frac{\|x^n - x^\infty\|}{\|x^\infty\|}
\]

Normalized RMS Distance

- \(\|x^n - x^\infty\| / \|x^\infty\|: 400\) iterations of single-coordinate descent

(Thanks to Web Stayman for interfacing LBFGS with ASPIRE.)
SUMMARY

Grouped Coordinate Descent Algorithm

- Accommodates non-quadratic cost function (for noise robustness and preserving edges)
- Monotonically decreases $\Phi$
- Converges globally to unique minimizer
- Easily accommodates nonnegativity constraint
- Parallelizable
- Converges faster than a general-purpose optimization method

Future Work:
Extend convergence proofs for multiple global minimizers:

Slides and paper available from:
http://www.eecs.umich.edu/~fessler/