Dec. 19, 1997

The University of Michigan
EECS Department

Jeffrey A. Fessler and Hakan Erdogan

Robust Edge-Preserving Algorithms for PET Image Reconstruction
Outline

Motivation:
PET Image Reconstruction

Cost Function Description (for a simpler problem)

Minimization Algorithms
- Huber Algorithm
- Generalized Huber Algorithm
- Optimization Transfer
- Huber Algorithm
- Minimization Algorithms

Anecdotal Preliminary Results

Summary and Future Work

- Grouped Coordinate Descent (GCD) Algorithm
Almost all statistical methods for PET image reconstruction are based on the following Poisson statistical model:

\[ Y_i \sim \text{Poisson} \left( \frac{\gamma j \cdot g_{ij} \cdot \theta_i}{\varepsilon_i} \right) \]

where:
- \( Y_i \): measured counts in sinogram bins
- \( \gamma j \): unknown radiotracer concentration in the \( j \)th voxel
- \( \gamma j \cdot g_{ij} \cdot \theta_i \): photon survival probability along the ray (attenuation)
- \( \varepsilon_i \): \( i \)th detector efficiency
- \( g_{ij} \): projection matrix
- \( \theta_i \): random coincidences

\( \left\{ \theta_i + \gamma j \cdot g_{ij} \cdot \theta_i \right\} \text{Poisson} \sim Y_i \)
If the Poisson model is valid, it is natural to estimate the emission image by finding the estimate to the sinogram data, as measured by the log-likelihood:

\[
\chi = \arg \max_{\chi} L(\chi) = (\chi)^{\text{MLE}}
\]

where

\[
(\chi)^{\text{MLE}} = (\chi)^{T}
\]

By the log-likelihood:

- By finding the “best fit” to the sinogram data measured image \( \chi \) by maximizing \( L(\chi) \).
- If the Poisson model is valid, it is natural to estimate the emission.
Possible Solution: Robust Log-Likelihood

Proposed Robust Log-likelihood Function

- Poisson Log-likelihood
- Robust Form

to be continued...
Goal: recover an estimate $\hat{x}$ of $x$ from $\overline{y}$.

Errors in $A$ partially motivate robust methods.

Each column in $A$ is a point response function.

$A$: known system model

$\overline{A}$: unknown object (true image)

$x$: unknown object (true image or sinogram)

$\overline{y}$: noisy measurements (blurred image or sinogram)

\[
\overline{y} + xA = \overline{h}
\]
One wants to fit the data, i.e., $\bar{x} \approx \bar{y}$. Therefore, the data fit cost function is:

$$0 \approx \bar{x}V - \bar{y}$$
Robust Data-Fit Cost Function

Huber function: $\psi(t) = \begin{cases} 
  \frac{t^2}{2}, & |t| \leq \delta, \\
  \delta|t| - \frac{\delta^2}{2}, & |t| > \delta
\end{cases}$
The sample mean is well known to be very sensitive to outliers.

$$\begin{align*}
\bar{X} &= \frac{1}{n} \sum_{i=1}^{n} X_i \\
\text{Sample Mean:} \\
\bar{X} &= \frac{1}{n} \sum_{i=1}^{n} X_i \\
\text{Sample Median:} \\
\tilde{X} &= \text{median} \left( \sum_{i=1}^{n} X_i \right)
\end{align*}$$
Mean vs Median

Generalized-Gaussian family of pdfs with unit variance:

$$f_X(x; \mu, p) = \frac{p}{2 \Gamma(1/p)} \sqrt{r_p} e^{-|x-\mu|^{p} r_p^{p/2}} \quad \text{where} \ r_p = \frac{\Gamma(3/p)}{\Gamma(1/p)}.$$
Minimizing \( \phi \) data is inadequate for ill-conditioned inverse problems.

Prior knowledge of piece-wise smoothness:

\[ \bullet \quad x_i - x_{i-1} \geq 0 \]
\[ \bullet \quad x_{i+1} - 2x_i + x_{i-1} \geq 0 \]
\[ \bullet \quad x_i \approx 0 \]

... 

Combining: \( Cz \approx z \)

Regularized cost function:

\[ \Phi(z) = \phi_{\text{data}}(z) + \sum_{i=1}^{m^2} \psi_{\text{penalty}}(Cz_i) \]

\[ (\text{piece-wise constant}) \]

\[ (\text{piece-wise linear}) \]

\[ (\text{support constraints}) \]
For edge-preserving image recovery, need non-quadratic constraint on\( \mathcal{F} \).

Conventional (Tikhonov-Miller) regularization:

\[
\begin{bmatrix}
    x_u I \otimes h_u A \\
    x_u A \otimes h_u I
\end{bmatrix} = C
\begin{bmatrix}
    1 & 1 & 0 & 0 & 0 & 0 \\
    1 & 1 & 0 & 0 & 0 & 0 \\
    0 & 0 & 1 & 1 & 0 & 0 \\
    0 & 0 & 0 & 1 & 1 & 0 \\
    0 & 0 & 0 & 0 & 1 & 1 \\
    0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix} = u \mathcal{A}
\]

Example: Roughness penalty (Gibbs prior)
Optimization problem:
\[
\hat{x} = \text{arg min}_x \Phi(x)
\]
or
\[
\hat{x} = \text{arg min}_x \hat{\Phi}(x).
\]

Regularized edge-preserving cost function is a special case:
\[
\Phi(x) = \Phi_{\text{data}}(x) + \Phi_{\text{penalty}}(x),
\]
\[
B = \begin{bmatrix} A \\ C \end{bmatrix}, \quad c = \begin{bmatrix} y \\ \bar{z} \end{bmatrix}
\]

Unified Cost Function

\[
\Phi(x) = \sum_{i=1}^{m} \psi_i(\|B\bar{x} - c_i\|)
\]
Optimization

Simple in quadratic case where

\[
\frac{i}{t} = \frac{t}{2} = \frac{1}{8}
\]

Proposition: algorithms tailored to structure of \( \phi \) can outperform

Good (fast converging) iterative algorithms:

- Preconditioned conjugate gradients
- Coordinate descent (Gauss-Seidel)

Challenging for non-quadratic \( \phi \)'s

Very challenging for non-convex \( \phi \)'s

But cannot solve all...
Assumptions

\( \frac{\partial f}{\partial x} \) rules out entropy, look at...

Easily ensured with perturbation of regularizer

\( M \) has a unique minimizer \( \Phi \)

\[ \infty > (0)^{\Phi \mathcal{M}} > 0 \]

\( \lim \frac{t}{(t)}\Phi \mapsto \lim = (0)^{\Phi \mathcal{M}} \)

\[ 0 < \frac{t}{(t)\Phi} \]

Each individual cost-function satisfies

Easily achieved with sensible regularization design

\[ 0 < MB, B \Leftrightarrow 0 < \mathcal{M} \] has full column rank, so \( B \)
$\omega_{\psi}(t_0)$ is the curvature of the parabola that is tangent at $t_0$
Therefore need iterative algorithm.

Unconstrained solution:

\[
\frac{\partial}{\partial \bar{x}} \left( \bar{x}^T B - \bar{y} \right) = \bar{y}^T \bar{A}^T \bar{X} = 0
\]

subject to

\[
\bar{x} \in \mathbb{R}_+^n
\]

where

\[
\left( \frac{\partial}{\partial \bar{x}} \left( \bar{x}^T B - \bar{y} \right) \right)_{\phi} \phi \Delta = \left( \bar{x}^T B - \bar{y} \right) \phi \Delta
\]

Column Gradient:

\[
\left( \frac{\partial}{\partial \bar{x}} \left( \bar{x}^T B - \bar{y} \right) \right)_{\phi} \phi \Delta \bar{X} = \left( \bar{x}^T B - \bar{y} \right) \phi \Delta
\]
Weighting Functions $\omega_{\psi}$

The graph shows two types of weighting functions:

- **Quadratic**: Represented by a dashed line.
- **Robust / Edge-preserving**: Represented by a solid line.

The x-axis represents the measurement minus the prediction, ranging from -4 to 4. The y-axis represents the weighting function $\omega(\cdot)$, ranging from 0 to 1.2.
Generic remedy: bound-constrained Quasi-Newton algorithms

- Impractical for image recovery due to matrix inversion
- Does not enforce nonnegative constraint
- Not guaranteed to monotonically decrease
- Not guaranteed to converge
- Requires twice-differentiable function

Disadvantages:

- Super-linear convergence rate (if convergent)

Advantages:

\[
\left\{ \left( \phi - \bar{x} \mathcal{B} \right)_i \right\}_{i=1}^{n} = (\bar{x} \mathcal{B}) \mathcal{V}
\]

where

\[
(\bar{x} \mathcal{B}) \Phi \Delta_1 \left[ \mathcal{B} (\bar{x} \mathcal{B}) \mathcal{V}, \mathcal{B} \right] - \bar{x} = \frac{1}{1+\bar{x}}
\]
Successive substitutions is often not convergent. Why here?

- Impractical for image recovery due to matrix inversion
- Does not enforce nonnegativity constraint

Disadvantages:

- Converges globally to unique minimizer (not shown by Huber)
- Monotonically decreases

Advantages:

\[
(wx)\Phi \Delta_{1-}[B(wx)u,B] - wx = \frac{1}{1+wx}
\]

Successive Substitutions:

\[
(x)\Phi \Delta_{1-}[B(x)u,B] - x = \tilde{c}(x)u,B_{1-}[B(x)u,B] = \tilde{x}
\]

Recall

\[
\text{Huber Algorithm (1981)}
\]
Optimization Transfer
MonotoneDecreaseIn

Minimizing surrogate function
ensures a monotone decrease in
the original cost function if:

The above sufficient conditions are satisfied by

\[
(u \bar{x})^\phi \geq \frac{(\bar{x} B - \bar{y})(u \bar{x}) U_i (\bar{x} B - \bar{y})}{2} = (u \bar{x} : \bar{x})_i^\phi
\]

\[
(u \bar{x} : \bar{x})_i^\phi \min \arg \min_{\text{Huber}} \bar{x} = 1 + u \bar{x}
\]

Huber's Algorithm:

\[
(u \bar{x} : \bar{x})_i^\phi \geq \frac{(\bar{x})^\phi}{u \bar{x} = \bar{x}} = \frac{(u \bar{x} : \bar{x})_i^\phi \Delta}{(\bar{x})^\phi}
\]

\[
(u \bar{x})^\phi = (u \bar{x} : u \bar{x})^\phi
\]

the original cost function.

\[\text{Minimizing surrogate function } \phi \text{ ensures a monotone decrease in } \phi.\]
Optimization Transfer in 2D
Generalized Huber Algorithm

Converges slower than Huber Algorithm

Does not enforce nonnegativity constraint in general

Disadvantages:

Can choose $M_n$ to be easily invertible, e.g., diagonal.

Or splitting matrices more generally

Converges globally to unique minimizer

Monotonically decreases

Advantages:

$B(u \bar{v}) U B \preceq u W$

where

$$(u \bar{v}) \Phi \Delta_{I - uW} - u \bar{v} = 1 + u \bar{v}$$
One can use the convexity of the Huber surrogate function to define a second surrogate function that is separable:}

\[ \nabla (u \bar{x} : \bar{x} - \bar{x}) \preceq (u \bar{x} : \bar{x}) \quad \text{descent direction} \]

Minimizing the separable paraboloid is trivial, especially compared to minimizing a paraboloid.

\[
\begin{align*}
&\min_{x} \frac{1}{2} \sum_{i=1}^{n} \left\| x - y_{i} \right\|_d^2 \\
&\text{where}
\end{align*}
\]
Separable Paraboloid Algorithm

\begin{align*}
\text{Advantages:} & \\
& + \left[ (u\overline{x}) \Phi \Delta \left\{ \left( \frac{(u\overline{x}:0)^T b}{1} \right) \text{diag} - u\overline{x} \right\} \right]
= + \left[ \frac{(u\overline{x}:0)^T b}{(u\overline{x}:0)^T b - u\overline{x}} \right] = 1 + u\overline{x}
\end{align*}

\begin{itemize}
\item Very slow convergence (ala EM algorithm)
\item Parallelizable (all pixels updated simultaneously)
\item Can enforce nonnegativity constraint
\item No matrix inversion required
\item Converges globally to unique minimizer
\item Monotonically decreases
\item Solution: update only a subset of the pixels simultaneously
\end{itemize}
**Grouped Coordinate Descent Algorithm**

Construct a separable paraboloidal surrogate function but for only a (large) subset of the pixels.

**Pixel Groups (2x3)**

```
1  5  3  1  5  3  1  5
4  2  6  4  2  6  4  2
1  5  3  1  5  3  1  5
4  2  6  4  2  6  4  2
1  5  3  1  5  3  1  5
4  2  6  4  2  6  4  2
```

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

Advantages/

• Monotonically decreases
• Converges globally to unique minimizer
• No matrix inversion required
• Can enforce nonnegativity constraint
• Parallelizable (all pixels updated simultaneously)
• Can enforce nonnegativity constraint
• Parallelizable (all pixels updated simultaneously)
• Fast convergence

Disadvantages:

• Slightly less parallelizable.
• Slightly more complicated implementation
• Slightly more complicated implementation
• Slightly less parallelizable.
• More complicated to explain.
• More complicated to explain.

(π). RFTs for shift-invariant PSF separable blur in PET

Grouped Coordinate Descent Algorithm
128 x 128 attenuation map with 4.5 mm pixel size

- 160 radial by 192 angular samples
- 0.921M prompt coincidences

PET Transmission Example

12 minute transmission scan from ECAT EXACT (single slice)
Normalized RMS Distance

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

where \( x^\infty \): 400 iterations of single-coordinate descent

LBFGS: Limited Memory Bound Constrained Quasi-Newton Method  
(Thanks to Web Stayman for interfacing LBFGS with ASPIRE.)
Grouped Coordinate Descent Algorithm

- Monotonically decreases \( \phi \)
- For noise robustness and preserving edges
- Accommodates non-quadratic cost function
- Easily accommodates non-negativity constraint
- Converges globally to unique minimizer
- Converges faster than a general-purpose optimization method
- Parallelizable
- Easily accommodates non-negativity constraint
- Converges globally to unique minimizer
- PET emission reconstruction problem
- Convergence proof for multiple minima

Future Work:
Convergence


Let $M$ be a point to set mapping such that on $C$, $W$ is uniformly compact, upper semi-continuous, and strictly monotonic with respect to $M$.

If $\{x_n\}$ is any sequence generated by the algorithm corresponding to $M$, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.

Conversely, let $\{x_n\}$ be any accumulation point of $\{x_n\}$.

Then $x_n$ is a fixed point of $M$. If $\{x_n\}$ is a continuum, then

$\forall \varepsilon > 0, \exists k \in \mathbb{N}$ such that

$\|x_{n+k} - x_n\| < \varepsilon$.