Using Neumann Series to Solve Inverse Problems in Imaging

Christopher Kumar Anand
Inverse Problem

\[ T f = m \]

Solve for \( f \).

- given
- measurements \((m)\)
- model \((T)\)
Seismic Imaging

1. Bang
2. Listen
3. Solve Acoustic Equations

Magnetic Resonance Imaging

0. Tissue Density

1. Phase Modulation

2. Sample Fourier Transform

3. Invert Linear System
Challenging when...

- model is big
  - 1,000,000,000 variables
- model is nonlinear
- data is inexact

(usually know error probabilistically)
Imaging

- discretize continuous model
- regular volume/area elements
- sparse structure

\( \rho_{i,j} \)
Solutions: Noise

- filter noisy solution
  1. convolution filter
  2. bilateral filter
  3. Anisotropic Diffusion (uses pde)
- regularize via penalty
  1. energy
  2. Total Variation
  3. something new
Solutions: Problem Size

I. use a fast method (i.e. based on FFT)
II. use (parallelizable) iterative method
   a. Conjugate Gradient
   b. Neumann series
III. use sparsity
    a. choose penalties with sparse Hessians
IV. use fast hardware
   a. 1000-way parallelizable
   b. single precision
Cell BE

- 25 GFlops DP
- 200 GFlops SP
- need 384-way ||ism
  - 4-way SIMD
  - 8-way cores
  - 6-times unrolling
  - double buffering
Solutions: Nonlinearity

use iterative method

A. sequential projection onto convex sets

B. trust region

C. sequential quadratic approximations
Plan of Talk

A. example/benchmark
B. optimization
   1. fit to data
   2. regularization
      i. new penalty (with optimized gradient)
      ii. nonlinear penalties
C. solution
   1. operator decomposition
   2. Neumann series
D. proof of convergence
E. numerical example
   1. noise reduction
   2. linear convergence
Example/Benchmark

\[ \rho : \Omega \rightarrow \mathbb{C} \quad \text{complex image} \]

\[ \mu_1, \mu_2, \mu_3, \mu_4 : \Omega/2 \rightarrow \mathbb{C} \quad \text{complex data} \]

\[ \mu_{m;i,j} = S_{m;i,j} \rho_{i,j} + S_{m;i+128,j} \rho_{i,j} + \epsilon_{m;i,j} \]

model
**Example/Benchmark**

\[ \mu_{m;i,j} = S_{m;i,j} \rho_{i,j} + S_{m;i+128,j} \rho_{i,j} + \epsilon_{m;i,j} \]

\[ T = \begin{pmatrix}
S_{1;\text{top}} & S_{1;\text{bottom}} \\
S_{2;\text{top}} & S_{2;\text{bottom}} \\
S_{3;\text{top}} & S_{3;\text{bottom}} \\
S_{4;\text{top}} & S_{4;\text{bottom}}
\end{pmatrix} \]  

\[ T^T T \]  

diagonal blocks

easily invertible
Example/Benchmark

\[(T^T T)^{-1} T^T \mu\]
Optimization

fit-to-data

\[
\min_{\rho : \Omega \to \mathbb{C}} \lambda \phi_{\text{data}} + \lambda_{\text{bi2}} \phi_{\text{bi2}} + \lambda_{\text{biTv}} \phi_{\text{biTv}} + \lambda_{\text{mask}} \phi_{\text{mask}} + \lambda_{\text{magnet}} \phi_{\text{magnet}} + \lambda_{\text{seg}} \phi_{\text{seg}}
\]

quadratic penalties  nonlinear penalties
Fit to Data

\[ \phi_{\text{data}}(f) = \| T f - m \|^2 \]

\[ \nabla \phi_{\text{data}} = 2 T^T T f - 2 T^T m \]

\[ \mathcal{H} \phi_{\text{data}} = 2 T^T T. \]

- typically dense transformation
- use fast matrix-vector products
  - e.g. FFT-based forward/adjoint problem
- linear forward problem gives quadratic objective
Bilateral Filter

\[ \hat{f}(x) = \sum_{y \in \mathbb{R} \setminus \{x\}} c(y - x) s(f(y) - f(x)) f(y) \]

Bilateral Regularizer

\[ \phi_{bi}(f) = \sum_{y \neq x} c(y - x) s(f(y) - f(x)) \]
Quadtratic Case

\[ \phi_{bi2} = \sum_{y \neq x} c(y - x) \| f(y) - f(x) \|^2 \]

- optimize direction of gradient
- LP problem like FIR filter design
- heuristic choice of “stop band”

\[ \frac{\partial}{\partial f_i(x)} \phi_{bi2} = 2 \sum_{y \neq x} (f_i(x) - f_i(y))c(x - y) = 2 \left( f_i(x) - \sum_{y \neq x} f_i(y)c(y - x) \right) \]
Optimal Spatial Kernel

- $-\nabla \phi_{bi}(f_{\text{zero}}) = 0$
- $-\nabla \phi_{bi}(f_{\text{high}}) = -f_{\text{high}}$

**discrete kernel**
**not rotational symmetric**

**exact in quadratic case**
**use in all cases**
TV-like

\[ \phi_{\text{biTV}} = \sum_{y \neq x} c(y - x) \| f(y) - f(x) \| \]

- similar properties to Total Variation
- won’t smooth edges
- not differentiable
Sequential Quadratic TV-like

$$\phi_{\text{biTV}} = \sum_{y \neq x} \frac{\epsilon c(y - x)}{\sqrt{\| \tilde{f}(y) - \tilde{f}(x) \|^2 + \epsilon}} \| f(y) - f(x) \|^2$$

- sequential quadratic approximation
- tangent to TV-like
Mask

\[
\phi_{\text{mask}} = \sum_{\{x \mid x \text{ is air}\}} \| f(x) \|^2
\]

- penalize pixel values outside object
Segmentation

\[
\phi_{\text{seg}}(f) = \sum_{\{g \mid \text{mean values of components}\}} e^{-\|f - g\|^2 / \sigma^2}
\]

- probability of observing pixels
- assumes discrete pixel values
- equal likelihood
- equal normal error
Solve: Operator Decomposition

\[ \mathcal{H}(\phi_i) + \alpha I = (A + B) \]

- block diagonal with sparse banded blocks
- linear-time Cholesky decomposition

\[ A = LL^T \]

- leads to formal expression

\[ (A + B)^{-1} = (LL^T + B)^{-1} \]

\[ = L^{T-1}(I + L^{-1}BL^{T-1})^{-1}L^{-1} \]
Calculate Using Fast Matrix-Vector Ops

(1) calculate $L^{-1}(-\nabla \phi_i)$ by back-solving in $O(nN)$ operations
(2) calculate $L^{T-1}L^{-1}(-\nabla \phi_i)$ by back-solving in $O(nN)$ more operations
(3) save result
(4) calculate $BL^{T-1}L^{-1}(-\nabla \phi_i)$ using the fast computation for $B$
(5) calculate $L^{-1}BL^{T-1}L^{-1}(-\nabla \phi_i)$ by back-solving in $O(nN)$ more operations
(6) calculate $L^{T-1}L^{-1}BL^{T-1}L^{-1}(-\nabla \phi_i)$ by back-solving in $O(nN)$ more operations
(7) subtract from result
   ... continue to the order of truncation.

linear in problem size  plus  (poly-order)(fast algorithm)
Row by Row

• each block corresponds to a row
• each block can be calculated in parallel
• (number-rows)-way parallelism
Even Better: Use Minimax Polynomial

\[
\min_{p \text{ polynomial}} \max_{x \in \text{spec}(L^{-1}BLT^{-1})} \left\| \frac{1}{1 + x} - p(x) \right\| = \epsilon
\]

- calculate

\[
L^{T-1} p(L^{-1}BLT^{-1}) L^{-1}(-\nabla \phi_i)
\]
Convergence

\[
\|x + \Delta x - x_0\| = \|x + p(A^{-1}B)A^{-1}(-\mathcal{H}(x - x_0)) - x_0\|
\]

\[
\leq \|p(A^{-1}B)A^{-1}(-\mathcal{H}(x - x_0)) - (\mathcal{H} + \alpha I)^{-1}(-\mathcal{H}(x - x_0))\|
\]

\[
+ \| (\mathcal{H} + \alpha I)^{-1}(-\mathcal{H}(x - x_0)) - (x - x_0) \|
\]

\[
\leq (\epsilon \|\mathcal{H}\| + \| (\mathcal{H} + \alpha I)^{-1} \| \alpha) \|x - x_0\|
\]
Safe in Single-Precision

• recalculate gradient at each outer iteration

• numerical error only builds up during polynomial evaluation

• coefficients well-behaved (and in our control)
Numerical Tests

• start with quadratic penalties
• add nonlinear penalties and change weights
• with and without time fixed budget for computation
10 iterations

I. 10 iterations with bi2 regularization
100 iterations

I. 10 iterations with bi2 regularization
II. introduce other penalties
   1. masking
   2. magnet
   3. segmentation
15 iterations

optimized c

simple c
Figure 7. Convergence using by stages the simplest to the most complex terms. Normalized to make the error in the direction of the graph. On the top we plot the total $L_2$ error (versus the true image). On the bottom we plot the difference between the error of the current rate and the error in the limit, to show that the linear convergence continues up to the 100th iteration.
Figure 7. Convergence using stages the simplest to the most complex alternative terms. Normalized to make their error in the direct inverse 1. On the top we plot the total $L_2$ error (versus the true image). On the bottom we plot the difference between the error of the current iterate and the error in the limit, to show that the linear convergence continues up to the 100th iteration.
Conclusion

• highly-parallel
• safe in single precision
• robust with respect to noise
• accommodates nonlinear penalties
Thanks to:

students and colleagues in the