Statistical Methods for Image Reconstruction

Jeffrey A. Fessler

EECS Department
The University of Michigan

Johns Hopkins University: “Short” Course

May 11, 2007
Image Reconstruction Methods
(Simplified View)

Analytical
(FBP)
(MR: iFFT)

Iterative
(OSEM?)
(MR: CG?)
Image Reconstruction Methods / Algorithms

ANALYTICAL

- FBP
- BPF
- Gridding
  ...

ITERATIVE

ALGEBRAIC

- Algebraic
  \( y = Ax \)
  - ART
  - MART
  - SMART
    ...

STOCHASTIC

- Least Squares
  - (Weighted)
  - CG
  - CD
  - ISRA
    ...

LIKELIHOOD

- Likelihood
  - (e.g., Poisson)
  - EM (etc.)
  - OSEM
  - SAGE
  - CG
  - Int. Point
  - GCA
  - PSCD
  - FSCD
    ...

Outline

Part 0: Introduction / Overview / Examples

Part 1: Problem Statements
○ Continuous-discrete vs continuous-continuous vs discrete-discrete

Part 2: Four of Five Choices for Statistical Image Reconstruction
○ Object parameterization
○ System physical modeling
○ Statistical modeling of measurements
○ Cost functions and regularization

Part 3: Fifth Choice: Iterative algorithms
○ Classical optimization methods
○ Considerations: nonnegativity, convergence rate, ...
○ Optimization transfer: EM etc.
○ Ordered subsets / block iterative / incremental gradient methods

Part 4: Performance Analysis
○ Spatial resolution properties
○ Noise properties
○ Detection performance
History

- Successive substitution method vs direct Fourier (Bracewell, 1956)
- Iterative method for X-ray CT (Hounsfield, 1968)
- ART for tomography (Gordon, Bender, Herman, JTB, 1970)
- Richardson/Lucy iteration for image restoration (1972, 1974)
- Weighted least squares for 3D SPECT (Goitein, NIM, 1972)
- Proposals to use Poisson likelihood for emission and transmission tomography
  Emission: (Rockmore and Macovski, TNS, 1976)
  Transmission: (Rockmore and Macovski, TNS, 1977)
- Expectation-maximization (EM) algorithms for Poisson model
  Emission: (Shepp and Vardi, TMI, 1982)
  Transmission: (Lange and Carson, JCAT, 1984)
- Regularized (aka Bayesian) Poisson emission reconstruction
  (Geman and McClure, ASA, 1985)
- Ordered-subsets EM algorithm
  (Hudson and Larkin, TMI, 1994)
- Commercial introduction of OSEM for PET scanners circa 1997
Why Statistical Methods?

- Object constraints (e.g., nonnegativity, object support)
- Accurate physical models (less bias $\implies$ improved quantitative accuracy) (e.g., nonuniform attenuation in SPECT)
- Improved spatial resolution?
- Appropriate statistical models (less variance $\implies$ lower image noise) (FBP treats all rays equally)
- Side information (e.g., MRI or CT boundaries)
- Nonstandard geometries (e.g., irregular sampling or “missing” data)

Disadvantages?

- Computation time
- Model complexity
- Software complexity

Analytical methods (a different short course!)

- Idealized mathematical model
  - Usually geometry only, greatly over-simplified physics
  - Continuum measurements (discretize/sample after solving)
- No statistical model
- Easier analysis of properties (due to linearity)
  e.g., Huesman (1984) FBP ROI variance for kinetic fitting
What about Moore’s Law?
**Benefit Example: Statistical Models**

<table>
<thead>
<tr>
<th>Method</th>
<th>Soft Tissue NRMS Error</th>
<th>Cortical Bone NRMS Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBP</td>
<td>22.7%</td>
<td>29.6%</td>
</tr>
<tr>
<td>PWLS</td>
<td>13.6%</td>
<td>16.2%</td>
</tr>
<tr>
<td>PL</td>
<td>11.8%</td>
<td>15.8%</td>
</tr>
</tbody>
</table>
Benefit Example: Physical Models

a. True object

b. Unorrected FBP

c. Monoenergetic statistical reconstruction

a. Soft−tissue corrected FBP

b. JS corrected FBP

c. Polyenergetic Statistical Reconstruction
Benefit Example: Nonstandard Geometries

Photon Source

Detector Bins
Truncated Fan-Beam SPECT Transmission Scan
One Final Advertisement: Iterative MR Reconstruction
Part 1: From Physics to Statistics

or

“What quantity is reconstructed?”
(in emission tomography)

Outline

• Decay phenomena and fundamental assumptions
• Idealized detectors
• Random phenomena
• Poisson measurement statistics
• State emission tomography reconstruction problem
What Object is Reconstructed?

In emission imaging, our aim is to image the radiotracer distribution.

The what?

At time $t = 0$ we inject the patient with some radiotracer, containing a “large” number $N$ of metastable atoms of some radionuclide.

Let $\vec{X}_k(t) \in \mathbb{R}^3$ denote the position of the $k$th tracer atom at time $t$. These positions are influenced by blood flow, patient physiology, and other unpredictable phenomena such as Brownian motion.

The ultimate imaging device would provide an exact list of the spatial locations $\vec{X}_1(t), \ldots, \vec{X}_N(t)$ of all tracer atoms for the entire scan.

Would this be enough?
Atom Positions or Statistical Distribution?

Repeating a scan would yield different tracer atom sample paths \( \left\{ \vec{X}_k(t) \right\}_{k=1}^N \).

\[ \therefore \text{statistical formulation} \]

**Assumption 1.** The spatial locations of individual tracer atoms at any time \( t \geq 0 \) are *independent* random variables that are all *identically distributed* according to a common probability density function (pdf) \( p_t(\vec{x}) \).

This pdf is determined by patient physiology and tracer properties.

Larger values of \( p_t(\vec{x}) \) correspond to “hot spots” where the tracer atoms tend to be located at time \( t \). Units: inverse volume, *e.g.*, atoms per cubic centimeter.

The *radiotracer distribution* \( p_t(\vec{x}) \) is the quantity of interest.

(Not \( \left\{ \vec{X}_k(t) \right\}_{k=1}^N \))
Example: Perfect Detector

True radiotracer distribution $p_t(\vec{x})$ at some time $t$.

A realization of $N = 2000$ i.i.d. atom positions (dots) recorded "exactly."

Little similarity!
Estimate of $p_r(\vec{x})$ formed by histogram binning of $N = 2000$ points. Ramp remains difficult to visualize.
Gaussian kernel density estimator for $p_t(\bar{x})$ from $N = 2000$ points.

Horizontal profiles at $x_2 = 3$ through density estimates.
Poisson Spatial Point Process

Assumption 2. The number of administered tracer atoms $N$ has a Poisson distribution with some mean

$$
\mu_N \triangleq \mathbb{E}[N] = \sum_{n=0}^{\infty} n \mathbb{P}\{N = n\}.
$$

Let $N_t(B)$ denote the number of tracer atoms that have spatial locations in any set $B \subset \mathbb{R}^3$ (VOI) at time $t$ after injection.

$N_t(\cdot)$ is called a Poisson spatial point process.

Fact. For any set $B$, $N_t(B)$ is Poisson distributed with mean:

$$
\mathbb{E}[N_t(B)] = \mathbb{E}[N] \mathbb{P}\{\vec{X}_k(t) \in B\} = \mu_N \int_B p_t(\vec{x}) \, d\vec{x}.
$$

Poisson $N$ injected atoms + i.i.d. locations $\implies$ Poisson point process
Illustration of Point Process \((\mu_N = 200)\)
Radionuclide Decay

preceding quantities are all unobservable. we “observe” a tracer atom only when it decays and emits photon(s).

the time that the $k$th tracer atom decays is a random variable $T_k$.

**assumption 3.** the $T_k$’s are statistically *independent* random variables, and are independent of the (random) spatial location.

**assumption 4.** each $T_k$ has an exponential distribution with mean $\mu_T = t_{1/2}/\ln 2$.

cumulative distribution function: $P\{T_k \leq t\} = 1 - \exp(-t/\mu_T)$
Statistics of an Ideal Decay Counter

Let $K_t(\mathcal{B})$ denote the number of tracer atoms that decay by time $t$, and that were located in the VOI $\mathcal{B} \subset \mathbb{R}^3$ at the time of decay.

Fact. $K_t(\mathcal{B})$ is a **Poisson counting process** with mean

$$E[K_t(\mathcal{B})] = \int_0^t \int_{\mathcal{B}} \lambda(\vec{x}, \tau) \, d\vec{x} \, d\tau,$$

where the (nonuniform) **emission rate density** is given by

$$\lambda(\vec{x}, t) \triangleq \mu_N \frac{e^{-t/\mu_T}}{\mu_T} \cdot p_t(\vec{x}).$$

**Ingredients:** “dose,” “decay,” “distribution”

**Units:** “counts” per unit time per unit volume, *e.g.*, $\mu$Ci/cc.

“Photon emission is a Poisson process”

What about the actual measurement statistics?
Idealized Detector Units

A nuclear imaging system consists of $n_d$ conceptual *detector units*.

**Assumption 5.** Each decay of a tracer atom produces a recorded count in at most one detector unit.

Let $S_k \in \{0, 1, \ldots, n_d\}$ denote the index of the incremented detector unit for decay of $k$th tracer atom. ($S_k = 0$ if decay is undetected.)

**Assumption 6.** The $S_k$’s satisfy the following conditional independence:

$$P\left\{S_1, \ldots, S_N \mid N, T_1, \ldots, T_N, \tilde{X}_1(\cdot), \ldots, \tilde{X}_N(\cdot)\right\} = \prod_{k=1}^N P\left\{S_k \mid \tilde{X}_k(T_k)\right\}.$$

The recorded bin for the $k$th tracer atom’s decay depends only on its position when it decays, and is independent of all other tracer atoms.

(No event pileup; no deadtime losses.)
PET Example

\[ n_d \leq (n_{\text{crystals}} - 1) \cdot n_{\text{crystals}} / 2 \]
SPECT Example

\[ n_d = n_{\text{radial bins}} \cdot n_{\text{angular steps}} \]
Detector Unit Sensitivity Patterns

Spatial localization:

\[ s_i(\vec{x}) \triangleq \text{probability that decay at } \vec{x} \text{ is recorded by } i\text{th detector unit.} \]

**Idealized Example. Shift-invariant PSF:**

\[ s_i(\vec{x}) = h(\vec{k}_i \cdot \vec{x} - r_i) \]

- \( r_i \) is the radial position of \( i\text{th ray} \)
- \( \vec{k}_i \) is the unit vector orthogonal to \( i\text{th parallel ray} \)
- \( h(\cdot) \) is the shift-invariant radial PSF (e.g., Gaussian bell or rectangular function)
Two representative $s_i(x)$ functions for a collimated Anger camera.
Example: PET Detector-Unit Sensitivity Patterns
Detector Unit Sensitivity Patterns

$s_i(\vec{x})$ can include the effects of
- geometry / solid angle
- collimation
- scatter
- attenuation
- detector response / scan geometry
- duty cycle (dwell time at each angle)
- detector efficiency
- positron range, noncollinearity
- ...

System sensitivity pattern:

$$s(\vec{x}) \triangleq \sum_{i=1}^{n_d} s_i(\vec{x}) = 1 - s_0(\vec{x}) \leq 1$$

(probability that decay at location $\vec{x}$ will be detected at all by system)
(Overall) System Sensitivity Pattern: $s(\bar{x}) = \sum_{i=1}^{n_d} s_i(\bar{x})$

Example: collimated $180^\circ$ SPECT system with uniform attenuation.
Detection Probabilities $s_i(\vec{x}_0)$ (vs det. unit index $i$)

Image domain

Sinogram domain
Summary of Random Phenomena

• Number of tracer atoms injected $N$
• Spatial locations of tracer atoms $\left\{ \vec{X}_k(t) \right\}_{k=1}^N$
• Time of decay of tracer atoms $\left\{ T_k \right\}_{k=1}^N$
• Detection of photon $[S_k \neq 0]$
• Recording detector unit $\left\{ S_k \right\}_{i=1}^{n_d}$
Emission Scan

Record events in each detector unit for \( t_1 \leq t \leq t_2 \).

\( Y_i \triangleq \text{number of events recorded by } i\text{th detector unit during scan, for } i = 1, \ldots, n_d. \)

\[
Y_i \triangleq \sum_{k=1}^{N} 1\{S_k=i, \ T_k\in[t_1,t_2]\}.
\]

The collection \( \{Y_i : i = 1, \ldots, n_d\} \) is our sinogram.

Note \( 0 \leq Y_i \leq N \).

Fact. Under Assumptions 1-6 above,

\[
Y_i \sim \text{Poisson}\left\{ \int s_i(\vec{x}) \lambda(\vec{x}) \, d\vec{x} \right\} \quad \text{(cf “line integral”)}
\]

and \( Y_i \)'s are statistically independent random variables, where the emission density is given by

\[
\lambda(\vec{x}) = \mu_N \int_{t_1}^{t_2} \frac{1}{\mu_T} e^{-t/\mu_T} p_t(\vec{x}) \, dt.
\]

(Local number of decays per unit volume during scan.)

Ingredients:
- dose (injected)
- duration of scan
- decay of radionuclide
- distribution of radiotracer
Poisson Statistical Model (Emission)

Actual measured counts = “foreground” counts + “background” counts.

Sources of background counts:
- cosmic radiation / room background
- random coincidences (PET)
- scatter not accounted for in $s_i(\vec{x})$
- “crosstalk” from transmission sources in simultaneous T/E scans
- anything else not accounted for by $\int s_i(\vec{x}) \lambda(\vec{x}) \, d\vec{x}$

Assumption 7.
The background counts also have independent Poisson distributions.

Statistical model (continuous to discrete)

$$Y_i \sim \text{Poisson} \left\{ \int s_i(\vec{x}) \lambda(\vec{x}) \, d\vec{x} + r_i \right\}, \quad i = 1, \ldots, n_d$$

$r_i$ : mean number of “background” counts recorded by $i$th detector unit.
Emission Reconstruction Problem

Estimate the emission density $\lambda(\vec{x})$ using (something like) this model:

$$Y_i \sim \text{Poisson} \left\{ \int s_i(\vec{x}) \lambda(\vec{x}) \, d\vec{x} + r_i \right\}, \quad i = 1, \ldots, n_d.$$  

Knowns:

- $\{Y_i = y_i\}_{i=1}^{n_d}$: observed counts from each detector unit
- $s_i(\vec{x})$ sensitivity patterns (determined by system models)
- $r_i$'s: background contributions (determined separately)

Unknown: $\lambda(\vec{x})$
List-mode acquisitions

Recall that conventional sinogram is temporally binned:

\[ Y_i \triangleq \sum_{k=1}^{N} 1\{s_k=i, T_k \in [t_1, t_2]\}. \]

This binning discards temporal information.

List-mode measurements: record all (detector,time) pairs in a list, \textit{i.e.},

\( \{(S_k, T_k) : k = 1, \ldots, N\} \).

List-mode dynamic reconstruction problem:

Estimate \( \lambda (\bar{x}, t) \) given \( \{(S_k, T_k)\} \).
Emission Reconstruction Problem - Illustration

\[ \lambda(\vec{x}) \]

\[ \{Y_i\} \]

\[ x_1 \]

\[ x_2 \]

\[ \theta \]

\[ r \]
Part 1: Problem Statement(s)

Example: in PET, the goal is to reconstruct radiotracer distribution \( \lambda(\vec{x}) \) from photon pair coincidence measurements \( \{y_i\}_{i=1}^{n_d} \), given the detector sensitivity patterns \( s_i(\vec{x}) \), \( i = 1, \ldots, n_d \), for each “line of response.”

Statistical model: \( y_i \sim \text{Poisson} \left\{ \int \lambda(\vec{x}) s_i(\vec{x}) \, d\vec{x} + r_i \right\} \)
Each “k-space sample” involves the transverse magnetization $f(\vec{x})$ weighted by:

- sinusoidal (complex exponential) pattern corresponding to k-space location $\vec{k}$
- RF receive coil sensitivity pattern
- phase effects of field inhomogeneity
- spin relaxation effects.

$$y_i = \int f(\vec{x}) s_i(\vec{x}) \, d\vec{x} + \epsilon_i, \quad i = 1, \ldots, n_d,$$

$$s_i(\vec{x}) = c_{RF}(\vec{x}) e^{-i\omega(\vec{x})t_i} e^{-t_i/T_2(\vec{x})} e^{-i2\pi\vec{k}(t_i) \cdot \vec{x}}$$
Continuous-Discrete Models

Emission tomography: $y_i \sim \text{Poisson}\left\{ \int \lambda(\vec{x}) s_i(\vec{x}) \, d\vec{x} + r_i \right\}$

Transmission tomography (monoenergetic): $y_i \sim \text{Poisson}\left\{ b_i \exp\left( - \int_{L_i} \mu(\vec{x}) \, d\ell \right) + r_i \right\}$

Transmission (polyenergetic): $y_i \sim \text{Poisson}\left\{ \int I_i(E) \exp\left( - \int_{L_i} \mu(\vec{x}, E) \, d\ell \right) \, dE + r_i \right\}$

Magnetic resonance imaging: $y_i = \int f(\vec{x}) s_i(\vec{x}) \, d\vec{x} + \epsilon_i$

Discrete measurements $y = (y_1, \ldots, y_{n_d})$
Continuous-space unknowns: $\lambda(\vec{x}), \mu(\vec{x}), f(\vec{x})$
Goal: estimate $f(\vec{x})$ given $y$

Solution options:

- Continuous-continuous formulations ("analytical")
- Continuous-discrete formulations usually $\hat{f}(\vec{x}) = \sum_{i=1}^{n_d} c_i s_i(\vec{x})$
- Discrete-discrete formulations $f(\vec{x}) \approx \sum_{j=1}^{n_p} x_j b_j(\vec{x})$
Part 2: Five Categories of Choices

- Object parameterization: function $f(\vec{r})$ vs finite coefficient vector $\mathbf{x}$
- System physical model: $\{s_i(\vec{r})\}$
- Measurement statistical model $y_i \sim ?$
- Cost function: data-mismatch and regularization
- Algorithm / initialization

No perfect choices - one can critique all approaches!
Choice 1. Object Parameterization

Finite measurements: \( \{y_i\}_{i=1}^{n_d} \). Continuous object: \( f(\vec{r}) \).

“All models are wrong but some models are useful.”

Linear \textit{series expansion} approach. Replace \( f(\vec{r}) \) by \( x = (x_1, \ldots, x_{n_p}) \) where

\[
  f(\vec{r}) \approx \tilde{f}(\vec{r}) = \sum_{j=1}^{n_p} x_j b_j(\vec{r}) \quad \text{← “basis functions”}
\]

Forward projection:

\[
  \int s_i(\vec{r}) f(\vec{r}) \, d\vec{r} = \int s_i(\vec{r}) \left[ \sum_{j=1}^{n_p} x_j b_j(\vec{r}) \right] \, d\vec{r} = \sum_{j=1}^{n_p} \left[ \int s_i(\vec{r}) b_j(\vec{r}) \, d\vec{r} \right] x_j
\]

\[
  = \sum_{j=1}^{n_p} a_{ij} x_j = [Ax]_i, \quad \text{where } a_{ij} \triangleq \int s_i(\vec{r}) b_j(\vec{r}) \, d\vec{r}
\]

- Projection integrals become finite summations.
- \( a_{ij} \) is contribution of \( j \)th basis function (e.g., voxel) to \( i \)th measurement.
- The units of \( a_{ij} \) and \( x_j \) depend on the user-selected units of \( b_j(\vec{r}) \).
- The \( n_d \times n_p \) matrix \( A = \{a_{ij}\} \) is called the \textit{system matrix}. 

2.2
(Linear) Basis Function Choices

- Fourier series (complex / not sparse)
- Circular harmonics (complex / not sparse)
- Wavelets (negative values / not sparse)
- Kaiser-Bessel window functions (blobs)
- Overlapping circles (disks) or spheres (balls)
- Polar grids, logarithmic polar grids
- “Natural pixels” \( \{s_i(r)\} \)
- B-splines (pyramids)
- Rectangular **pixels** / **voxels** (rect functions)
- Point masses / bed-of-nails / lattice of points / “comb” function
- Organ-based voxels (e.g., from CT in PET/CT systems)
- ...
Basis Function Considerations

Mathematical
- Represent \( f(\vec{r}) \) “well” with moderate \( n_p \) (approximation accuracy)
- \textit{e.g.}, represent a constant (uniform) function
- Orthogonality? (not essential)
- Linear independence (ensures uniqueness of expansion)
- Insensitivity to shift of basis-function grid (approximate shift invariance)
- Rotation invariance

Computational
- “Easy” to compute \( a_{ij} \) values and/or \( Ax \)
- If stored, the system matrix \( A \) should be sparse (mostly zeros).
- Easy to represent nonnegative functions \textit{e.g.}, if \( x_j \geq 0 \), then \( f(\vec{r}) \geq 0 \).
  A sufficient condition is \( b_j(\vec{r}) \geq 0 \).
Nonlinear Object Parameterizations

Estimation of intensity and shape (e.g., location, radius, etc.)

Surface-based (homogeneous) models
- Circles / spheres
- Ellipses / ellipsoids
- Superquadrics
- Polygons
- Bi-quadratic triangular Bezier patches, ...

Other models
- Generalized series \( f(\vec{r}) = \sum_j x_j b_j(\vec{r}, \theta) \)
- Deformable templates \( f(\vec{r}) = b(T_\theta(\vec{r})) \)
- ...

Considerations
- Can be considerably more parsimonious
- If correct, yield greatly reduced estimation error
- Particularly compelling in limited-data problems
- Often oversimplified (all models are wrong but...)
- Nonlinear dependence on location induces non-convex cost functions, complicating optimization
Example Basis Functions - 1D

1. Continuous object
2. Piecewise Constant Approximation
3. Quadratic B-Spline Approximation
Pixel Basis Functions - 2D

Continuous image \( f(\vec{r}) \)

Pixel basis approximation

\[ \sum_{j=1}^{n_p} x_j b_j(\vec{r}) \]
Blobs in SPECT: Qualitative

(2D SPECT thorax phantom simulations)
Blobs in SPECT: Quantitative

Standard deviation vs. bias in reconstructed phantom images

- Per iteration, rotation–based
- Per iteration, blob–based $\alpha=10.4$
- Per iteration, blob–based $\alpha=0$
- Per FWHM, rotation–based
- Per FWHM, blob–based $\alpha=10.4$
- Per FWHM, blob–based $\alpha=0$
- FBP
Discrete-Discrete Emission Reconstruction Problem

Having chosen a basis and \textit{linearly} parameterized the emission density...

Estimate the emission density coefficient vector \( \mathbf{x} = (x_1, \ldots, x_{np}) \) (aka “image”) using (something like) this statistical model:

\[
y_i \sim \text{Poisson} \left\{ \sum_{j=1}^{np} a_{ij} x_j + r_i \right\}, \quad i = 1, \ldots, n_d.
\]

- \( \{y_i\}_{i=1}^{n_d} \): observed counts from each detector unit
- \( \mathbf{A} = \{a_{ij}\} \): system matrix (determined by system models)
- \( r_i \)'s: background contributions (determined separately)

Many image reconstruction problems are “find \( \mathbf{x} \) given \( \mathbf{y} \)” where

\[
y_i = g_i(\lfloor \mathbf{A} \mathbf{x} \rfloor_i) + \varepsilon_i, \quad i = 1, \ldots, n_d.
\]
Choice 2. System Model, aka Physics

System matrix elements:  \[ a_{ij} = \int s_i(\vec{r}) b_j(\vec{r}) \, d\vec{r} \]

- scan geometry
- collimator/detector response
- attenuation
- scatter (object, collimator, scintillator)
- duty cycle (dwell time at each angle)
- detector efficiency / dead-time losses
- positron range, noncollinearity, crystal penetration, ...
- ...

Considerations
- Improving system model can improve
  - Quantitative accuracy
  - Spatial resolution
  - Contrast, SNR, detectability
- Computation time (and storage vs compute-on-fly)
- Model uncertainties
  (e.g., calculated scatter probabilities based on noisy attenuation map)
- Artifacts due to over-simplifications
Measured System Model?

Determine $a_{ij}$’s by scanning a voxel-sized cube source over the imaging volume and recording counts in all detector units (separately for each voxel).

- Avoids mathematical model approximations
- Scatter / attenuation added later (object dependent), approximately
- Small probabilities $\Rightarrow$ long scan times
- Storage
- Repeat for every voxel size of interest
- Repeat if detectors change
“Line Length” System Model for Tomography

\[ a_{ij} \triangleq \text{length of intersection} \]

\[ x_1 \quad x_2 \]

\[ \text{ith ray} \]
“Strip Area” System Model for Tomography

\[ a_{ij} \triangleq \text{area} \]
(Implicit) System Sensitivity Patterns

\[\sum_{i=1}^{n_d} a_{ij} \approx s(\vec{r}_j) = \sum_{i=1}^{n_d} s_i(\vec{r}_j)\]

- Line Length
- Strip Area
Point-Lattice Projector/Backprojector

\[ a_{ij} \text{'s determined by linear interpolation} \]
Point-Lattice Artifacts

Projections (sinograms) of uniform disk object:

Point Lattice

Strip Area
Forward- / Back-projector “Pairs”

Forward projection (image domain to projection domain):

\[ \tilde{y}_i = \int s_i(\vec{r}) f(\vec{r}) \, d\vec{r} = \sum_{j=1}^{n_p} a_{ij} x_j = [Ax]_i, \quad \text{or} \quad \tilde{y} = Ax \]

Backprojection (projection domain to image domain):

\[
A'y = \left\{ \sum_{i=1}^{n_d} a_{ij} y_i \right\}_{j=1}^{n_p}
\]

The term “forward/backprojection pair” often corresponds to an implicit choice for the object basis and the system model.

Sometimes \( A'y \) is implemented as \( By \) for some “backprojector” \( B \neq A' \)

Least-squares solutions (for example):

\[ \hat{x} = [A'A]^{-1}A'y \neq [BA]^{-1}By \]
Mismatched Backprojector $B \neq A'$

\[
\begin{align*}
x & \quad \hat{x}(PWLS - CG) & \quad \hat{x}(PWLS - CG) \\
\text{Matched} & \quad \text{Matched} & \quad \text{Mismatched}
\end{align*}
\]
Horizontal Profiles

\[ \hat{f}(x_1, x_32) \]
System Model Tricks

- **Factorize** (e.g., PET Gaussian detector response)
  \[ A \approx SG \]
  (geometric projection followed by Gaussian smoothing)

- **Symmetry**

- **Rotate and Sum**

- **Gaussian diffusion**
  for SPECT Gaussian detector response

- **Correlated Monte Carlo** (Beekman *et al.*)

In all cases, consistency of backprojector with $A'$ requires care.
Complications: nonuniform attenuation, depth-dependent PSF, Compton scatter

(MR system models discussed in Part II)
Choice 3. Statistical Models

After modeling the system physics, we have a deterministic “model:”

\[ y_i \approx g_i([Ax]_i) \]

for some functions \( g_i \), e.g., \( g_i(l) = l + r_i \) for emission tomography.

Statistical modeling is concerned with the “\( \approx \)” aspect.

**Considerations**

- More accurate models:
  - can lead to lower variance images,
  - may incur additional computation,
  - may involve additional algorithm complexity (e.g., proper transmission Poisson model has nonconcave log-likelihood)
- Statistical model errors (e.g., deadtime)
- Incorrect models (e.g., log-processed transmission data)
Statistical Model Choices for Emission Tomography

- “None.” Assume \( y - r = Ax \). “Solve algebraically” to find \( x \).
- White Gaussian noise. Ordinary least squares: minimize \( \|y - Ax\|^2 \)
  (This is the appropriate statistical model for MR.)
- Non-white Gaussian noise. Weighted least squares: minimize
  \[
  \|y - Ax\|^2_W = \sum_{i=1}^{n_d} w_i (y_i - [Ax]_i)^2, \text{ where } [Ax]_i \triangleq \sum_{j=1}^{n_p} a_{ij} x_j
  \]
  (e.g., for Fourier rebinned (FORE) PET data)
- Ordinary Poisson model (ignoring or precorrecting for background)
  \[y_i \sim \text{Poisson}\{[Ax]_i\}\]
- Poisson model
  \[y_i \sim \text{Poisson}\{[Ax]_i + r_i\}\]
- Shifted Poisson model (for randoms precorrected PET)
  \[y_i = y_i^{\text{prompt}} - y_i^{\text{delay}} \sim \text{Poisson}\{[Ax]_i + 2r_i\} - 2r_i\]
Shifted Poisson model for PET

Precorrected random coincidences: \( y_i = y_i^{\text{prompt}} - y_i^{\text{delay}} \)

\[
\begin{align*}
    y_i^{\text{prompt}} &\sim \text{Poisson}\{[Ax]_i + r_i\} \\
y_i^{\text{delay}} &\sim \text{Poisson}\{r_i\} \\
E[y_i] &= [Ax]_i \\
\text{Var}\{y_i\} &= [Ax]_i + 2r_i
\end{align*}
\]

Mean \(\neq\) Variance \(\implies\) not Poisson!

Statistical model choices

- Ordinary Poisson model: ignore randoms
  \( [y_i]_+ \sim \text{Poisson}\{[Ax]_i\} \)
  Causes bias due to truncated negatives
- Data-weighted least-squares (Gaussian model):
  \( y_i \sim \mathcal{N}([Ax]_i, \hat{\sigma}^2_i) \), \( \hat{\sigma}^2_i = \max(y_i + 2\hat{r}_i, \sigma_{\text{min}}^2) \)
  Causes bias due to data-weighting
- Shifted Poisson model (matches 2 moments):
  \( [y_i + 2\hat{r}_i]_+ \sim \text{Poisson}\{[Ax]_i + 2\hat{r}_i\} \)
  Insensitive to inaccuracies in \( \hat{r}_i \).
  One can further reduce bias by retaining negative values of \( y_i + 2\hat{r}_i \).
A model that includes both photon variability and electronic readout noise:

\[ y_i \sim \text{Poisson}\{\bar{y}_i(\mu)\} + \mathcal{N}(0, \sigma^2) \]

Shifted Poisson approximation

\[ [y_i + \sigma^2]^+ \sim \text{Poisson}\{\bar{y}_i(\mu) + \sigma^2\} \]

or just use WLS...

Complications:
- Intractability of likelihood for Poisson+Gaussian
- Compound Poisson distribution due to photon-energy-dependent detector signal.

X-ray statistical modeling is a current research area in several groups!
Choice 4. Cost Functions

Components:
- *Data-mismatch* term
- *Regularization* term (and regularization parameter $\beta$)
- Constraints (e.g., nonnegativity)

Cost function:

$$
\Psi(x) = \text{DataMismatch}(y, Ax) + \beta \text{Roughness}(x)
$$

Reconstruct image $\hat{x}$ by minimization:

$$
\hat{x} \triangleq \arg \min_{x \geq 0} \Psi(x)
$$

Actually *several* sub-choices to make for Choice 4 ...

Distinguishes “statistical methods” from “algebraic methods” for “$y = Ax$."

Why Cost Functions?

(vs “procedure” e.g., adaptive neural net with wavelet denoising)

**Theoretical reasons**
ML is based on minimizing a cost function: the negative log-likelihood
- ML is asymptotically consistent
- ML is asymptotically unbiased
- ML is asymptotically efficient (under true statistical model...)
- **Estimation**: Penalized-likelihood achieves uniform CR bound asymptotically
- **Detection**: Qi and Huesman showed analytically that MAP reconstruction out-performs FBP for SKE/BKE lesion detection (T-MI, Aug. 2001)

**Practical reasons**
- Stability of estimates (if $\Psi$ and algorithm chosen properly)
- Predictability of properties (despite nonlinearities)
- Empirical evidence (?)
Bayesian Framework

Given a prior distribution $p(x)$ for image vectors $x$, by Bayes’ rule:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

so

$$\log p(x|y) = \log p(y|x) + \log p(x) - \log p(y)$$

- $-\log p(y|x)$ corresponds to data mismatch term (negative log-likelihood)
- $-\log p(x)$ corresponds to regularizing penalty function

**Maximum a posteriori (MAP) estimator:**

$$\hat{x} = \arg\max_x \log p(x|y) = \arg\max_x \log p(y|x) + \log p(x)$$

- Has certain optimality properties (provided $p(y|x)$ and $p(x)$ are correct).
- Same form as $\Psi$
Choice 4.1: Data-Mismatch Term

Options (for emission tomography):

- Negative log-likelihood of statistical model. Poisson emission case:
  \[ -L(\mathbf{x}; \mathbf{y}) = - \log p(\mathbf{y} | \mathbf{x}) = \sum_{i=1}^{n_d} ([Ax]_i + r_i) - y_i \log([Ax]_i + r_i) + \log y_i! \]

- Ordinary (unweighted) least squares: \[ \sum_{i=1}^{n_d} \frac{1}{2} (y_i - \hat{r}_i - [Ax]_i)^2 \]
- Data-weighted least squares: \[ \sum_{i=1}^{n_d} \frac{1}{2} (y_i - \hat{r}_i - [Ax]_i)^2 / \hat{\sigma}^2_i, \quad \hat{\sigma}^2_i = \max \left(y_i + \hat{r}_i, \sigma_{\text{min}}^2\right), \] (causes bias due to data-weighting).
- Reweighted least-squares: \[ \hat{\sigma}^2_i = [Ax]_i + \hat{r}_i \]
- Model-weighted least-squares (nonquadratic, but convex!)
  \[ \sum_{i=1}^{n_d} \frac{1}{2} (y_i - \hat{r}_i - [Ax]_i)^2 / ([Ax]_i + \hat{r}_i)^2 \]

- Nonquadratic cost-functions that are robust to outliers
- ...

Considerations

- Faithfulness to statistical model vs computation
- Ease of optimization (convex?, quadratic?)
- Effect of statistical modeling errors
Choice 4.2: Regularization

Forcing too much “data fit” gives noisy images
Ill-conditioned problems: small data noise causes large image noise

Solutions:
• Noise-reduction methods
• True regularization methods

Noise-reduction methods
• Modify the data
  ○ Prefilter or “denoise” the sinogram measurements
  ○ Extrapolate missing (e.g., truncated) data
• Modify an algorithm derived for an ill-conditioned problem
  ○ Stop algorithm before convergence
  ○ Run to convergence, post-filter
  ○ Toss in a filtering step every iteration or couple iterations
  ○ Modify update to “dampen” high-spatial frequencies
Noise-Reduction vs True Regularization

Advantages of noise-reduction methods
- Simplicity (?)
- Familiarity
- Appear less subjective than using penalty functions or priors
- Only fiddle factors are # of iterations, or amount of smoothing
- Resolution/noise tradeoff usually varies with iteration (stop when image looks good - in principle)
- Changing post-smoothing does not require re-iterating

Advantages of true regularization methods
- Stability (unique minimizer & convergence $\Rightarrow$ initialization independence)
- Faster convergence
- Predictability
- Resolution can be made object independent
- Controlled resolution (e.g., spatially uniform, edge preserving)
- Start with reasonable image (e.g., FBP) $\Rightarrow$ reach solution faster.
True Regularization Methods

Redefine the *problem* to eliminate ill-conditioning, rather than patching the data or algorithm!

**Options**

- Use bigger pixels (fewer basis functions)
  - Visually unappealing
  - Can only preserve edges coincident with pixel edges
  - Results become even less invariant to translations

- Method of sieves (constrain image roughness)
  - Condition number for “pre-emission space” can be even worse
  - Lots of iterations
  - Commutability condition rarely holds exactly in practice
  - Degenerates to post-filtering in some cases

- Change cost function by adding a roughness penalty / prior
  \[ \hat{x} = \arg\min_x \Psi(x), \quad \Psi(x) = \ell(x) + \beta R(x) \]
  - Disadvantage: apparently subjective choice of penalty
  - Apparent difficulty in choosing penalty parameter(s), e.g., $\beta$
    (cf. apodizing filter / cutoff frequency in FBP)
Penalty Function Considerations

- Computation
- Algorithm complexity
- Uniqueness of minimizer of $\Psi(x)$
- Resolution properties (edge preserving?)
- # of adjustable parameters
- Predictability of properties (resolution and noise)

Choices
- separable vs nonseparable
- quadratic vs nonquadratic
- convex vs nonconvex
Penalty Functions: Separable vs Nonseparable

Separable

- Identity norm: \( R(x) = \frac{1}{2}x' L x = \sum_{j=1}^{np} x_j^2 / 2 \)
  penalizes large values of \( x \), but causes “squashing bias”

- Entropy: \( R(x) = \sum_{j=1}^{np} x_j \log x_j \)

- Gaussian prior with mean \( \mu_j \), variance \( \sigma_j^2 \): \( R(x) = \sum_{j=1}^{np} \frac{(x_j - \mu_j)^2}{2\sigma_j^2} \)

- Gamma prior \( R(x) = \sum_{j=1}^{np} p(x_j, \mu_j, \sigma_j) \) where \( p(x, \mu, \sigma) \) is Gamma pdf

The first two basically keep pixel values from “blowing up.”
The last two encourage pixels values to be close to prior means \( \mu_j \).

General separable form: \( R(x) = \sum_{j=1}^{np} f_j(x_j) \)

Slightly simpler for minimization, but these do not explicitly enforce smoothness.
The simplicity advantage has been overcome in newer algorithms.
Penalty Functions: Separable vs Nonseparable

Nonseparable (partially couple pixel values) to penalize roughness

Example

\[ R(x) = (x_2 - x_1)^2 + (x_3 - x_2)^2 + (x_5 - x_4)^2 \\
+ (x_4 - x_1)^2 + (x_5 - x_2)^2 \]

Rougher images \(\Rightarrow\) larger \(R(x)\) values
Roughness Penalty Functions

First-order neighborhood and pairwise pixel differences:

\[ R(x) = \sum_{j=1}^{n_p} \frac{1}{2} \sum_{k \in \mathcal{N}_j} \psi(x_j - x_k) \]

\(\mathcal{N}_j \triangleq \text{neighborhood of } j\text{th pixel (e.g., left, right, up, down)}\)
\(\psi\) called the \textit{potential function}

Finite-difference approximation to continuous roughness measure:

\[ R(f(\cdot)) = \int \|\nabla f(\vec{r})\|^2 d\vec{r} = \int \left\\bigg| \frac{\partial}{\partial x} f(\vec{r}) \right\\bigg|^2 + \left\\bigg| \frac{\partial}{\partial y} f(\vec{r}) \right\\bigg|^2 + \left\\bigg| \frac{\partial}{\partial z} f(\vec{r}) \right\\bigg|^2 d\vec{r}. \]

Second derivatives also useful:
(More choices!)

\[ \frac{\partial^2}{\partial x^2} f(\vec{r}) \bigg|_{\vec{r} = \vec{r}_j} \approx f(\vec{r}_{j+1}) - 2f(\vec{r}_j) + f(\vec{r}_{j-1}) \]

\[ R(x) = \sum_{j=1}^{n_p} \psi(x_{j+1} - 2x_j + x_{j-1}) + \cdots \]
Penalty Functions: General Form

\[ R(x) = \sum_k \psi_k([Cx]_k) \]

where \[ [Cx]_k = \sum_{j=1}^{n_p} c_{kj} x_j \]

Example:

\[
C = \begin{bmatrix}
-1 & 1 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 \\
-1 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 0 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \\ x_5 - x_4 \\ x_4 - x_1 \\ x_5 - x_2 \end{bmatrix}
\]

\[
R(x) = \sum_{k=1}^5 \psi_k([Cx]_k)
\]

\[ = \psi_1(x_2 - x_1) + \psi_2(x_3 - x_2) + \psi_3(x_5 - x_4) + \psi_4(x_4 - x_1) + \psi_5(x_5 - x_2) \]
Penalty Functions: Quadratic vs Nonquadratic

\[ R(x) = \sum_k \psi_k([Cx]_k) \]

**Quadratic** \( \psi_k \)

If \( \psi_k(t) = \frac{t^2}{2} \), then \( R(x) = \frac{1}{2}x'C'Cx \), a quadratic form.
- Simpler optimization
- Global smoothing

**Nonquadratic** \( \psi_k \)

- Edge preserving
- More complicated optimization. (This is essentially solved in convex case.)
- Unusual noise properties
- Analysis/prediction of resolution and noise properties is difficult
- More adjustable parameters (e.g., \( \delta \))

Example: Huber function. \( \psi(t) \triangleq \begin{cases} t^2/2, & |t| \leq \delta \\ \delta |t| - \delta^2/2, & |t| > \delta \end{cases} \)

Example: Hyperbola function. \( \psi(t) \triangleq \delta^2 \left( \sqrt{1+(t/\delta)^2} - 1 \right) \)
Quadratic vs Non-quadratic Potential Functions

- **Parabola (quadratic)**
- **Huber, $\delta=1$**
- **Hyperbola, $\delta=1$**

Lower cost for large differences $\implies$ edge preservation
Edge-Preserving Reconstruction Example

Phantom

Quadratic Penalty

Huber Penalty
More “Edge Preserving” Regularization

Chlewicki et al., PMB, Oct. 2004: “Noise reduction and convergence of Bayesian algorithms with blobs based on the Huber function and median root prior”
Piecewise Constant “Cartoon” Objects

400 k-space samples

\[ |x| \text{ true} \]

\[ \angle x \text{ true} \]

\[ |x| \text{ conj phase} \]

\[ \angle x \text{ conj phase} \]

\[ |x| \text{ pcg quad} \]

\[ \angle x \text{ pcg quad} \]

\[ |x| \text{ pcg edge} \]

\[ \angle x \text{ pcg edge} \]
Total Variation Regularization

Non-quadratic roughness penalty:

$$\int \| \nabla f(\vec{r}) \| \, d\vec{r} \approx \sum_k |[C\mathbf{x}]_k|$$

Uses \textit{magnitude} instead of \textit{squared magnitude} of gradient.

Problem: $| \cdot |$ is not differentiable.

Practical solution:

$$|t| \approx \delta \left( \sqrt{1 + \left( \frac{t}{\delta} \right)^2} - 1 \right)$$

(hyperbola!)

\[ \text{Potential functions} \]

\[ \text{Total Variation} \]

\[ \text{Hyperbola, } \delta=0.2 \]

\[ \text{Hyperbola, } \delta=1 \]
Penalty Functions: Convex vs Nonconvex

**Convex**
- Easier to optimize
- Guaranteed unique minimizer of $\Psi$ (for convex negative log-likelihood)

**Nonconvex**
- Greater degree of edge preservation
- Nice images for piecewise-constant phantoms!
- Even more unusual noise properties
- Multiple extrema
- More complicated optimization (simulated / deterministic annealing)
- Estimator $\hat{x}$ becomes a discontinuous function of data $Y$

**Nonconvex examples**
- “broken parabola”
  $$\psi(t) = \min(t^2, t_{\text{max}}^2)$$
- true median root prior:
  $$R(x) = \sum_{j=1}^{n_p} \frac{(x_j - \text{median}_j(x))^2}{\text{median}_j(x)}$$
  where $\text{median}_j(x)$ is local median

Exception: orthonormal wavelet threshold *denoising* via nonconvex potentials!
Potential Functions

\[ t = x_j - x_k \]

- Parabola (quadratic)
- Huber (convex)
- Broken parabola (non-convex)

Potential Function \( \psi(t) \)
Local Extrema and Discontinuous Estimators

Small change in data $\iff$ large change in minimizer $\hat{x}$. Using convex penalty functions obviates this problem.
Augmented Regularization Functions

Replace roughness penalty $R(x)$ with $R(x|b) + \alpha R(b)$, where the elements of $b$ (often binary) indicate boundary locations.

- Line-site methods
- Level-set methods

Joint estimation problem:

$$(\hat{x}, \hat{b}) = \arg \min_{x,b} \Psi(x, b), \quad \Psi(x, b) = \ell(x)[x; y] + \beta R(x|b) + \alpha R(b).$$

Example: $b_{jk}$ indicates the presence of edge between pixels $j$ and $k$:

$$R(x|b) = \sum_{j=1}^{n_p} \sum_{k \in \mathcal{N}_j} (1 - b_{jk}) \frac{1}{2} (x_j - x_k)^2$$

Penalty to discourage too many edges (e.g.):

$$R(b) = \sum_{jk} b_{jk}.$$

- Can encourage local edge continuity
- May require annealing methods for minimization
Modified Penalty Functions

\[ R(x) = \sum_{j=1}^{n_p} \sum_{k \in \mathcal{N}_j} \frac{1}{2} w_{jk} \psi(x_j - x_k) \]

Adjust weights \( \{w_{jk}\} \) to
- Control resolution properties
- Incorporate anatomical side information (MR/CT)
  (avoid smoothing across anatomical boundaries)

**Recommendations**
- Emission tomography:
  - Begin with quadratic (nonseparable) penalty functions
  - Consider modified penalty for resolution control and choice of \( \beta \)
  - Use modest regularization and post-filter more if desired
- Transmission tomography (attenuation maps), X-ray CT
  - consider convex nonquadratic (e.g., Huber) penalty functions
  - choose \( \delta \) based on attenuation map units (water, bone, etc.)
  - choice of regularization parameter \( \beta \) remains nontrivial,
    learn appropriate values by experience for given study type
Choice 4.3: Constraints

- Nonnegativity
- Known support
- Count preserving
- Upper bounds on values
  *e.g.*, maximum $\mu$ of attenuation map in transmission case

Considerations

- Algorithm complexity
- Computation
- Convergence rate
- Bias (in low-count regions)
- ...
Open Problems

Modeling

- Noise in $a_{ij}$ values (system model errors)
- Noise in $\hat{r}_i$ values (estimates of scatter / randoms)
- Statistics of corrected measurements
- Statistics of measurements with deadtime losses

For PL or MAP reconstruction, Qi (MIC 2004) has derived a bound on system model errors relative to data noise.

Cost functions

- Performance prediction for nonquadratic penalties
- Effect of nonquadratic penalties on detection tasks
- Choice of regularization parameters for nonquadratic regularization
Summary

- 1. Object parameterization: function $f(\vec{r})$ vs vector $x$
- 2. System physical model: $s_i(\vec{r})$
- 3. Measurement statistical model $Y_i \sim ?$
- 4. Cost function: data-mismatch / regularization / constraints

Reconstruction Method $\triangleq$ Cost Function + Algorithm

Naming convention: “criterion”-“algorithm”:
- ML-EM, MAP-OSL, PL-SAGE, PWLS+SOR, PWLS-CG, ...
Part 3. Algorithms

Method = Cost Function + Algorithm

Outline

• Ideal algorithm
• Classical general-purpose algorithms
• Considerations:
  ◦ nonnegativity
  ◦ parallelization
  ◦ convergence rate
  ◦ monotonicity
• Algorithms tailored to cost functions for imaging
  ◦ Optimization transfer
  ◦ EM-type methods
  ◦ Poisson emission problem
  ◦ Poisson transmission problem
• Ordered-subsets / block-iterative algorithms
  ◦ Recent convergent versions (relaxation, incrementalism)
Why iterative algorithms?

- For nonquadratic $\Psi$, no closed-form solution for minimizer.
- For quadratic $\Psi$ with nonnegativity constraints, no closed-form solution.
- For quadratic $\Psi$ without constraints, closed-form solutions:

  PWLS:  
  \[ \hat{x} = \arg \min_x \| y - Ax \|^2_{W^{1/2}} + x'Rx = [A'WA + R]^{-1}A'Wy \]

  OLS:  
  \[ \hat{x} = \arg \min_x \| y - Ax \|^2 = [A'A]^{-1}A'y \]

  Impractical (memory and computation) for realistic problem sizes. $A$ is sparse, but $A'A$ is not.

All algorithms are imperfect. No single best solution.
General Iteration

Deterministic iterative mapping: \( x^{(n+1)} = \mathcal{M}(x^{(n)}) \)
Ideal Algorithm

\[ x^* \triangleq \arg \min_{x \geq 0} \Psi(x) \quad \text{(global minimizer)} \]

**Properties**

- stable and convergent
- converges quickly
- globally convergent
- fast
- robust
- user friendly
- parallelizable
- simple
- flexible

\( \{x^{(n)}\} \) converges to \( x^* \) if run indefinitely
\( \{x^{(n)}\} \) gets “close” to \( x^* \) in just a few iterations
\( \lim_{n} x^{(n)} \) independent of starting image \( x^{(0)} \)

requires minimal computation per iteration
insensitive to finite numerical precision
nothing to adjust (e.g., acceleration factors)

(when necessary)
easy to program and debug
accommodates any type of system model
(matrix stored by row or column, or factored, or projector/backprojector)

Choices: forgo one or more of the above
Classic Algorithms

Non-gradient based
- Exhaustive search
- Nelder-Mead simplex (amoeba)
Converge very slowly, but work with nondifferentiable cost functions.

Gradient based
- Gradient descent
  \[ x^{(n+1)} \triangleq x^{(n)} - \alpha \nabla \Psi(x^{(n)}) \]
  Choosing \( \alpha \) to ensure convergence is nontrivial.
- Steepest descent
  \[ x^{(n+1)} \triangleq x^{(n)} - \alpha_n \nabla \Psi(x^{(n)}) \quad \text{where} \quad \alpha_n \triangleq \arg\min_{\alpha} \Psi(x^{(n)} - \alpha \nabla \Psi(x^{(n)})) \]
  Computing stepsize \( \alpha_n \) can be expensive or inconvenient.

Limitations
- Converge slowly.
- Do not easily accommodate nonnegativity constraint.
**Gradients & Nonnegativity - A Mixed Blessing**

**Unconstrained optimization** of differentiable cost functions:

\[ \nabla \Psi(x) = 0 \text{ when } x = x^* \]

- A necessary condition always.
- A sufficient condition for strictly convex cost functions.
- Iterations search for zero of gradient.

**Nonnegativity-constrained minimization**:

Karush-Kuhn-Tucker conditions

\[
\frac{\partial}{\partial x_j} \Psi(x) \bigg|_{x = x^*} \quad \text{is} \quad \begin{cases} 
= 0, & x_j^* > 0 \\
\geq 0, & x_j^* = 0
\end{cases}
\]

- A necessary condition always.
- A sufficient condition for strictly convex cost functions.
- Iterations search for ???
- \( 0 = x_j^* \frac{\partial}{\partial x_j} \Psi(x^*) \) is a necessary condition, but never sufficient condition.
Karush-Kuhn-Tucker Illustrated

Inactive constraint

Active constraint

\[ \psi(x) \]
Why Not Clip Negatives?

Newton-Raphson with negatives set to zero each iteration. Fixed-point of iteration is not the constrained minimizer!
Newton-Raphson Algorithm

\[ x^{(n+1)} = x^{(n)} - [\nabla^2 \Psi(x^{(n)})]^{-1} \nabla \Psi(x^{(n)}) \]

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

**Disadvantages:**
- Requires twice-differentiable \( \Psi \)
- Not guaranteed to converge
- Not guaranteed to monotonically decrease \( \Psi \)
- Does not enforce nonnegativity constraint
- Computing Hessian \( \nabla^2 \Psi \) often expensive
- Impractical for image recovery due to matrix inverse

General purpose remedy: bound-constrained Quasi-Newton algorithms
Newton’s Quadratic Approximation

2nd-order Taylor series:

\[ \Psi(x) \approx \phi(x; x^{(n)}) \triangleq \Psi(x^{(n)}) + \nabla \Psi(x^{(n)})(x - x^{(n)}) + \frac{1}{2}(x - x^{(n)})^T \nabla^2 \Psi(x^{(n)})(x - x^{(n)}) \]

Set \( x^{(n+1)} \) to the (“easily” found) minimizer of this quadratic approximation:

\[
\begin{align*}
x^{(n+1)} & \triangleq \arg \min_x \phi(x; x^{(n)}) \\
& = x^{(n)} - \left[ \nabla^2 \Psi(x^{(n)}) \right]^{-1} \nabla \Psi(x^{(n)})
\end{align*}
\]

Can be nonmonotone for Poisson emission tomography log-likelihood, even for a single pixel and single ray:

\[ \Psi(x) = (x + r) - y \log(x + r). \]
Consideration: Monotonicity

An algorithm is monotonic if

$$\Psi(x^{(n+1)}) \leq \Psi(x^{(n)}), \quad \forall x^{(n)}.$$  

Three categories of algorithms:

- Nonmonotonic (or unknown)
- Forced monotonic (e.g., by line search)
- Intrinsically monotonic (by design, simplest to implement)

Forced monotonicity

Most nonmonotonic algorithms can be converted to forced monotonic algorithms by adding a line-search step:

$$x^{\text{temp}} \triangleq M(x^{(n)}), \quad d = x^{\text{temp}} - x^{(n)}$$

$$x^{(n+1)} \triangleq x^{(n)} - \alpha_n d^{(n)} \quad \text{where} \quad \alpha_n \triangleq \operatorname{arg\,min}_\alpha \Psi(x^{(n)} - \alpha d^{(n)})$$

Inconvenient, sometimes expensive, nonnegativity problematic.
Conjugate Gradient (CG) Algorithm

Advantages:
- Fast converging (if suitably preconditioned) (in unconstrained case)
- Monotonic (forced by line search in nonquadratic case)
- Global convergence (unconstrained case)
- Flexible use of system matrix $A$ and tricks
- Easy to implement in unconstrained quadratic case
- Highly parallelizable

Disadvantages:
- Nonnegativity constraint awkward (slows convergence?)
- Line-search somewhat awkward in nonquadratic cases
- Possible need to “restart” after many iterations

Highly recommended for unconstrained quadratic problems (e.g., PWLS without nonnegativity). Useful (but perhaps not ideal) for Poisson case too.
Consideration: Parallelization

**Simultaneous** (fully parallelizable)
update all pixels simultaneously using all data
EM, Conjugate gradient, ISRA, OSL, SIRT, MART, ...

**Block iterative** (ordered subsets)
update (nearly) all pixels using one subset of the data at a time
OSEM, RBBI, ...

**Row action**
update many pixels using a single ray at a time
ART, RAMLA

**Pixel grouped** (multiple column action)
update some (but not all) pixels simultaneously a time, using all data
Grouped coordinate descent, multi-pixel SAGE
Perhaps the most nontrivial to implement

**Sequential** (column action)
update one pixel at a time, using all (relevant) data
Coordinate descent, SAGE
Coordinate Descent Algorithm

aka Gauss-Siedel, successive over-relaxation (SOR), iterated conditional modes (ICM)

Update one pixel at a time, holding others fixed to their most recent values:

\[ x_j^{\text{new}} = \arg \min_{x_j \geq 0} \Psi(x_1^{\text{new}}, \ldots, x_j^{\text{new}}, x_{j+1}^{\text{old}}, \ldots, x_{n_p}^{\text{old}}), \quad j = 1, \ldots, n_p \]

**Advantages:**
- Intrinsically monotonic
- Fast converging (from good initial image)
- Global convergence
- Nonnegativity constraint trivial

**Disadvantages:**
- Requires column access of system matrix \( A \)
- Cannot exploit some “tricks” for \( A \), e.g., factorizations
- Expensive “arg min” for nonquadratic problems
- Poorly parallelizable
Constrained Coordinate Descent Illustrated

Clipped Coordinate–Descent Algorithm

\[ x_1 \]
\[ x_2 \]
Coordinate Descent - Unconstrained

Unconstrained Coordinate–Descent Algorithm
Coordinate-Descent Algorithm Summary

Recommended when all of the following apply:
- quadratic or nearly-quadratic convex cost function
- nonnegativity constraint desired
- precomputed and stored system matrix $A$ with column access
- parallelization not needed (standard workstation)

Cautions:
- Good initialization (e.g., properly scaled FBP) essential. (Uniform image or zero image cause slow initial convergence.)
- Must be programmed carefully to be efficient. (Standard Gauss-Siedel implementation is suboptimal.)
- Updates high-frequencies fastest $\Rightarrow$ poorly suited to unregularized case

Used daily in UM clinic for 2D SPECT / PWLS / nonuniform attenuation
Summary of General-Purpose Algorithms

**Gradient-based**
- Fully parallelizable
- Inconvenient line-searches for nonquadratic cost functions
- Fast converging in unconstrained case
- Nonnegativity constraint inconvenient

**Coordinate-descent**
- Very fast converging
- Nonnegativity constraint trivial
- Poorly parallelizable
- Requires precomputed/stored system matrix

CD is well-suited to moderate-sized 2D problem (e.g., 2D PET), but poorly suited to large 2D problems (X-ray CT) and fully 3D problems. Neither is ideal.

:: need *special-purpose algorithms* for image reconstruction!
Data-Mismatch Functions Revisited

For fast converging, intrinsically monotone algorithms, consider the form of $\Psi$.

WLS:

$$\ell(x) = \sum_{i=1}^{n_d} \frac{1}{2} w_i (y_i - [Ax]_i)^2 = \sum_{i=1}^{n_d} h_i([Ax]_i), \quad \text{where} \quad h_i(l) \equiv \frac{1}{2} w_i (y_i - l)^2.$$  

Emission Poisson (negative) log-likelihood:

$$\ell(x) = \sum_{i=1}^{n_d} ([Ax]_i + r_i) - y_i \log([Ax]_i + r_i) = \sum_{i=1}^{n_d} h_i([Ax]_i)$$

where $h_i(l) \equiv (l + r_i) - y_i \log(l + r_i).$

Transmission Poisson log-likelihood:

$$\ell(x) = \sum_{i=1}^{n_d} \left( b_i e^{-[Ax]_i} + r_i \right) - y_i \log \left( b_i e^{-[Ax]_i} + r_i \right) = \sum_{i=1}^{n_d} h_i([Ax]_i)$$

where $h_i(l) \equiv (b_i e^{-l} + r_i) - y_i \log(b_i e^{-l} + r_i).$

MRI, polyenergetic X-ray CT, confocal microscopy, image restoration, ...
All have same partially separable form.
General Imaging Cost Function

General form for data-mismatch function:

\[ \ell(x) = \sum_{i=1}^{n_d} h_i([Ax]_i) \]

General form for regularizing penalty function:

\[ R(x) = \sum_k \psi_k([Cx]_k) \]

General form for cost function:

\[ \Psi(x) = \ell(x) + \beta R(x) = \sum_{i=1}^{n_d} h_i([Ax]_i) + \beta \sum_k \psi_k([Cx]_k) \]

Properties of \( \Psi \) we can exploit:

- summation form (due to independence of measurements)
- convexity of \( h_i \) functions (usually)
- summation argument (inner product of \( x \) with \( i \)th row of \( A \))

Most methods that use these properties are forms of optimization transfer.
Optimization Transfer Illustrated

$\Psi(x) \text{ and } \phi^{(n)}(x)$

Surrogate function
Cost function

$x^{(n)} \quad x^{(n+1)}$
Optimization Transfer

General iteration:

\[ x^{(n+1)} = \arg \min_{x \geq 0} \phi(x; x^{(n)}) \]

Monotonicity conditions (cost function \( \Psi \) decreases provided these hold):

- \( \phi(x^{(n)}; x^{(n)}) = \Psi(x^{(n)}) \) (matched current value)
- \( \nabla_x \phi(x; x^{(n)}) \bigg|_{x=x^{(n)}} = \nabla \Psi(x) \bigg|_{x=x^{(n)}} \) (matched gradient)
- \( \phi(x; x^{(n)}) \geq \Psi(x) \quad \forall x \geq 0 \) (lies above)

These 3 (sufficient) conditions are satisfied by the \( Q \) function of the EM algorithm (and its relatives like SAGE).

The 3rd condition is \textit{not} satisfied by the Newton-Raphson quadratic approximation, which leads to its nonmonotonicity.
Optimization Transfer in 2d
Optimization Transfer cf EM Algorithm

E-step: choose surrogate function \( \phi(x; x^{(n)}) \)

M-step: minimize surrogate function

\[
x^{(n+1)} = \arg \min_{x \geq 0} \phi(x; x^{(n)})
\]

Designing surrogate functions
- Easy to “compute”
- Easy to minimize
- Fast convergence rate

Often mutually incompatible goals :: compromises
Convergence Rate: Slow

High Curvature
Small Steps
Slow Convergence
Convergence Rate: Fast

Low Curvature
Large Steps
Fast Convergence

Old  New

x

φ  Φ
Tool: Convexity Inequality

\[ g(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha g(x_1) + (1 - \alpha)g(x_2) \quad \text{for} \quad \alpha \in [0, 1] \]

More generally: \( \alpha_k \geq 0 \) and \( \sum_k \alpha_k = 1 \) \( \Rightarrow \quad g(\sum_k \alpha_k x_k) \leq \sum_k \alpha_k g(x_k). \quad \text{Sum outside!} \)
Example 1: Classical ML-EM Algorithm

Negative Poisson log-likelihood cost function (unregularized):
\[
\Psi(x) = \sum_{i=1}^{n_d} h_i([Ax]_i), \quad h_i(l) = (l + r_i) - y_i \log(l + r_i).
\]

Intractable to minimize directly due to summation within logarithm.

Clever trick due to De Pierro (let \( \bar{y}_i^{(n)} = [Ax]_i + r_i \)):
\[
[Ax]_i = \sum_{j=1}^{n_p} a_{ij} x_j = \sum_{j=1}^{n_p} \left[ \frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right] \left( \frac{x_j}{x_j^{(n)} \bar{y}_i^{(n)}} \right).
\]

Since the \( h_i \)'s are \textit{convex} in Poisson emission model:
\[
h_i([Ax]_i) = h_i \left( \sum_{j=1}^{n_p} \left[ \frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right] \left( \frac{x_j}{x_j^{(n)} \bar{y}_i^{(n)}} \right) \right) \leq \sum_{j=1}^{n_p} \left[ \frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right] h_i \left( \frac{x_j}{x_j^{(n)} \bar{y}_i^{(n)}} \right)
\]
\[
\Psi(x) = \sum_{i=1}^{n_d} h_i([Ax]_i) \leq \phi(x; x^{(n)}) \triangleq \sum_{i=1}^{n_d} \sum_{j=1}^{n_p} \left[ \frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right] h_i \left( \frac{x_j}{x_j^{(n)} \bar{y}_i^{(n)}} \right)
\]

Replace convex cost function \( \Psi(x) \) with \textit{separable} surrogate function \( \phi(x; x^{(n)}) \).
“ML-EM Algorithm” M-step

E-step gave separable surrogate function:

\[
\phi(x; x^{(n)}) = \sum_{j=1}^{n_p} \phi_j(x_j; x^{(n)}), \quad \text{where} \quad \phi_j(x_j; x^{(n)}) \triangleq \sum_{i=1}^{n_d} \left[ a_{ij} x_j^{(n)} \right] h_i \left( \frac{x_j}{\tilde{y}_i^{(n)}} \right).
\]

M-step separates:

\[
x^{(n+1)} = \arg \min_{x \geq 0} \phi(x; x^{(n)}) \implies x^{(n+1)}_j = \arg \min_{x_j \geq 0} \phi_j(x_j; x^{(n)}), \quad j = 1, \ldots, n_p
\]

Minimizing:

\[
\frac{\partial}{\partial x_j} \phi_j(x_j; x^{(n)}) = \sum_{i=1}^{n_d} a_{ij} \bar{h}_i \left( \frac{\tilde{y}_i^{(n)} x_j / x_j^{(n)}}{y_i^{(n)} x_j / x_j^{(n)}} \right) = \sum_{i=1}^{n_d} a_{ij} \left[ 1 - \frac{y_i^{(n)} x_j / x_j^{(n)}}{\tilde{y}_i^{(n)} x_j / x_j^{(n)}} \right] \bigg|_{x_j = x_j^{(n+1)}} = 0.
\]

Solving (in case \( r_i = 0 \)):

\[
x_j^{(n+1)} = x_j^{(n)} \left[ \sum_{i=1}^{n_d} a_{ij} \frac{y_i}{[Ax^{(n)}]_i} \right] / \left( \sum_{i=1}^{n_d} a_{ij} \right), \quad j = 1, \ldots, n_p
\]

- Derived without any statistical considerations, unlike classical EM formulation.
- Uses only convexity and algebra.
- Guaranteed monotonic: surrogate function \( \phi \) satisfies the 3 required properties.
- M-step trivial due to \textit{separable surrogate}.  

3.30
ML-EM is Scaled Gradient Descent

\[
x_j^{(n+1)} = x_j^{(n)} \left[ \sum_{i=1}^{n_d} a_{ij} \frac{y_i^{(n)}}{\bar{y}_i^{(n)}} \right] / \left( \sum_{i=1}^{n_d} a_{ij} \right)
\]

\[
= x_j^{(n)} + x_j^{(n)} \left[ \sum_{i=1}^{n_d} a_{ij} \left( \frac{y_i}{\bar{y}_i^{(m)}} - 1 \right) \right] / \left( \sum_{i=1}^{n_d} a_{ij} \right)
\]

\[
= x_j^{(n)} - \left( \frac{x_j^{(n)}}{\sum_{i=1}^{n_d} a_{ij}} \right) \frac{\partial}{\partial x_j} \Psi(x^{(n)}), \quad j = 1, \ldots, n_p
\]

\[
x^{(n+1)} = x^{(n)} + D(x^{(n)}) \nabla \Psi(x^{(n)})
\]

This particular diagonal scaling matrix remarkably

- ensures monotonicity,
- ensures nonnegativity.
Consideration: Separable vs Nonseparable

Contour plots: loci of equal function values.

Uncoupled vs coupled minimization.
Separable Surrogate Functions (Easy M-step)

The preceding EM derivation structure applies to any cost function of the form

$$\Psi(x) = \sum_{i=1}^{n_d} h_i([Ax]_i).$$

cf ISRA (for nonnegative LS), “convex algorithm” for transmission reconstruction

Derivation yields a separable surrogate function

$$\Psi(x) \leq \phi(x; x^{(n)}), \text{ where } \phi(x; x^{(n)}) = \sum_{j=1}^{n_p} \phi_j(x_j; x^{(n)})$$

M-step separates into 1D minimization problems (fully parallelizable):

$$x^{(n+1)} = \arg \min_{x \geq 0} \phi(x; x^{(n)}) \implies x_j^{(n+1)} = \arg \min_{x_j \geq 0} \phi_j(x_j; x^{(n)}), \quad j = 1, \ldots, n_p$$

Why do EM / ISRA / convex-algorithm / etc. converge so slowly?
Separable vs Nonseparable

Separable surrogates (e.g., EM) have high curvature \( \Rightarrow \) slow convergence. Nonseparable surrogates can have lower curvature \( \Rightarrow \) faster convergence. Harder to minimize? Use paraboloids (quadratic surrogates).
High Curvature of EM Surrogate

$h_i(l)$ and $Q(l;\mu^n)$
Find parabola $q_i^{(n)}(l)$ of the form:

$$q_i^{(n)}(l) = h_i\left(\ell_i^{(n)}\right) + \dot{h}_i\left(\ell_i^{(n)}\right)(l - \ell_i^{(n)}) + c_i^{(n)} \frac{1}{2}(l - \ell_i^{(n)})^2,$$

where $\ell_i^{(n)} \triangleq [A x^{(n)}]_i$.

Satisfies tangent condition. Choose curvature to ensure “lies above” condition:

$$c_i^{(n)} \triangleq \min \left\{ c \geq 0 : q_i^{(n)}(l) \geq h_i(l), \quad \forall l \geq 0 \right\}.$$
Paraboloidal Surrogate

Combining 1D parabola surrogates yields *paraboloidal surrogate*:

\[ \Psi(x) = \sum_{i=1}^{n_d} h_i([Ax]_i) \leq \phi(x; x^{(n)}) = \sum_{i=1}^{n_d} q_i^{(n)}([Ax]_i) \]

Rewriting: \( \phi(\delta + x^{(n)}; x^{(n)}) = \Psi(x^{(n)}) + \nabla \Psi(x^{(n)}) \delta + \frac{1}{2} \delta' A \text{diag}\{c_i^{(n)}\} A \delta \)

**Advantages**

- Surrogate \( \phi(x; x^{(n)}) \) is *quadratic*, unlike Poisson log-likelihood \( \implies \) easier to minimize
- Not separable (unlike EM surrogate)
- Not self-similar (unlike EM surrogate)
- Small curvatures \( \implies \) fast convergence
- Intrinsically monotone global convergence
- Fairly simple to derive / implement

**Quadratic minimization**

- Coordinate descent
  - fast converging
  - Nonnegativity easy
    - precomputed column-stored system matrix
- Gradient-based quadratic minimization methods
  - Nonnegativity inconvenient
Example: PSCD for PET Transmission Scans

- square-pixel basis
- strip-integral system model
- shifted-Poisson statistical model
- edge-preserving convex regularization (Huber)
- nonnegativity constraint
- inscribed circle support constraint
- paraboloidal surrogate coordinate descent (PSCD) algorithm
Separable Paraboloidal Surrogate

To derive a parallelizable algorithm apply another De Pierró trick:

$$\lfloor A x \rceil_i = \sum_{j=1}^{np} \pi_{ij} \left[ \frac{a_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + \ell_i^{(n)} \right], \quad \ell_i^{(n)} = [Ax^{(n)}]_i.$$ 

Provided $\pi_{ij} \geq 0$ and $\sum_{j=1}^{np} \pi_{ij} = 1$, since parabola $q_i$ is convex:

$$q_i^{(n)}(\lfloor A x \rceil_i) = q_i^{(n)} \left( \sum_{j=1}^{np} \pi_{ij} \left[ \frac{a_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + \ell_i^{(n)} \right] \right) \leq \sum_{j=1}^{np} \pi_{ij} q_i^{(n)} \left( \frac{a_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + \ell_i^{(n)} \right)$$

$$\phi(x; x^{(n)}) = \sum_{i=1}^{nd} q_i^{(n)}(\lfloor A x \rceil_i) \leq \tilde{\phi}(x; x^{(n)}) \triangleq \sum_{i=1}^{nd} \sum_{j=1}^{np} \pi_{ij} q_i^{(n)} \left( \frac{a_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + \ell_i^{(n)} \right)$$

Separable Paraboloidal Surrogate:

$$\tilde{\phi}(x; x^{(n)}) = \sum_{j=1}^{np} \phi_j(x_j; x^{(n)}), \quad \phi_j(x_j; x^{(n)}) \triangleq \sum_{i=1}^{nd} \pi_{ij} q_i^{(n)} \left( \frac{a_{ij}}{\pi_{ij}} (x_j - x_j^{(n)}) + \ell_i^{(n)} \right)$$

Parallelizable M-step (cf gradient descent!):

$$x_{j}^{(n+1)} = \arg \min_{x_j \geq 0} \phi_j(x_j; x^{(n)}) = \left[ x_j^{(n)} - \frac{1}{d_j^{(n)}} \frac{\partial}{\partial x_j} \Psi(x^{(n)}) \right]_+, \quad d_j^{(n)} = \sum_{i=1}^{nd} \frac{a_{ij}^2 c_i^{(n)}}{\pi_{ij}}$$

Natural choice is $\pi_{ij} = |a_{ij}|/|a|_i, |a|_i = \sum_{j=1}^{np} |a_{ij}|$
Example: Poisson ML Transmission Problem

Transmission negative log-likelihood (for $i$th ray):

$$h_i(l) = (b_i e^{-l} + r_i) - y_i \log(b_i e^{-l} + r_i).$$

Optimal (smallest) parabola surrogate curvature (Erdoğan, T-MI, Sep. 1999):

$$c_i^{(n)} = c(\ell^{(n)}_i, h_i), \quad c(l, h) = \begin{cases} 
\left[ \frac{2h(0) - h(l) + \dot{h}(l)l}{l^2} \right], & l > 0 \\
[\dot{h}(l)]_+, & l = 0.
\end{cases}$$

Separable Paraboloidal Surrogate (SPS) Algorithm:

Precompute $|a|_i = \sum_{j=1}^{n_p} a_{ij}$, \hspace{1cm} $i = 1, \ldots, n_d$

$$\ell^{(n)}_i = [A x^{(n)}]_i, \quad \text{(forward projection)}$$

$$\bar{y}^{(n)}_i = b_i e^{-\ell^{(n)}_i} + r_i \quad \text{(predicted means)}$$

$$\dot{h}^{(n)}_i = 1 - y_i/\bar{y}^{(n)}_i \quad \text{(slopes)}$$

$$c_i^{(n)} = c(\ell^{(n)}_i, h_i) \quad \text{(curvatures)}$$

$$x^{(n+1)}_j = \left[ x^{(n)}_j - \frac{1}{d^{(n)}_j} \frac{\partial}{\partial x_j} \Psi(x^{(n)}) \right]_+ = \left[ x^{(n)}_j - \frac{\sum_{i=1}^{n_d} a_{ij} \dot{h}^{(n)}_i}{\sum_{i=1}^{n_d} a_{ij} |a|_i c_i^{(n)}} \right]_+, \quad j = 1, \ldots, n_p$$

Monotonically decreases cost function each iteration. \hspace{1cm} No logarithm!
The MAP-EM M-step “Problem”

Add a penalty function to our surrogate for the negative log-likelihood:

\[ \Psi(x) = \mathcal{L}(x) + \beta \mathcal{R}(x) \]

\[ \phi(x; x^{(n)}) = \sum_{j=1}^{n_p} \phi_j(x_j; x^{(n)}) + \beta \mathcal{R}(x) \]

M-step: \[ x^{(n+1)} = \arg\min_{x \geq 0} \phi(x; x^{(n)}) = \arg\min_{x \geq 0} \sum_{j=1}^{n_p} \phi_j(x_j; x^{(n)}) + \beta \mathcal{R}(x) = ? \]

For nonseparable penalty functions, the M-step is coupled \( \therefore \) difficult.

Suboptimal solutions

- Generalized EM (GEM) algorithm (coordinate descent on \( \phi \))
  Monotonic, but inherits slow convergence of EM.
- One-step late (OSL) algorithm (use outdated gradients) (Green, T-MI, 1990)

\[ \frac{\partial}{\partial x_j} \phi(x; x^{(n)}) = \frac{\partial}{\partial x_j} \phi_j(x_j; x^{(n)}) + \beta \frac{\partial}{\partial x_j} \mathcal{R}(x) \approx \frac{\partial}{\partial x_j} \phi_j(x_j; x^{(n)}) + \beta \frac{\partial}{\partial x_j} \mathcal{R}(x^{(n)}) \]

Nonmonotonic. Known to diverge, depending on \( \beta \).
Temptingly simple, but avoid!

Contemporary solution

- Use separable surrogate for penalty function too (De Pierro, T-MI, Dec. 1995)
  Ensures monotonicity. Obviates all reasons for using OSL!
Apply separable paraboloidal surrogates to penalty function:

\[ R(x) \leq R_{SPS}(x; x^{(n)}) = \sum_{j=1}^{n_p} R_j(x_j; x^{(n)}) \]

Overall separable surrogate: \[ \phi(x; x^{(n)}) = \sum_{j=1}^{n_p} \phi_j(x_j; x^{(n)}) + \beta \sum_{j=1}^{n_p} R_j(x_j; x^{(n)}) \]

The M-step becomes fully parallelizable:

\[ x_j^{(n+1)} = \arg \min_{x_j \geq 0} \phi_j(x_j; x^{(n)}) - \beta R_j(x_j; x^{(n)}), \quad j = 1, \ldots, n_p. \]

Consider quadratic penalty \( R(x) = \sum_k \psi([C x]_k) \), where \( \psi(t) = \frac{t^2}{2} \).

If \( \gamma_{kj} \geq 0 \) and \( \sum_{j=1}^{n_p} \gamma_{kj} = 1 \) then

\[ [C x]_k = \sum_{j=1}^{n_p} \gamma_{kj} \left[ \frac{c_{kj}}{\gamma_{kj}} (x_j - x_j^{(n)}) + [C x^{(n)}]_k \right]. \]

Since \( \psi \) is convex:

\[ \psi([C x]_k) = \psi \left( \sum_{j=1}^{n_p} \gamma_{kj} \left[ \frac{c_{kj}}{\gamma_{kj}} (x_j - x_j^{(n)}) + [C x^{(n)}]_k \right] \right) \leq \sum_{j=1}^{n_p} \gamma_{kj} \psi \left( \frac{c_{kj}}{\gamma_{kj}} (x_j - x_j^{(n)}) + [C x^{(n)}]_k \right) \]
De Pierro’s Algorithm Continued

So \( R(\mathbf{x}) \leq R(\mathbf{x}; \mathbf{x}^{(n)}) \triangleq \sum_{j=1}^{np} R_j(x_j; \mathbf{x}^{(n)}) \) where

\[
R_j(x_j; \mathbf{x}^{(n)}) \triangleq \sum_k \gamma_{kj} \psi \left( \frac{c_{kj}}{\gamma_{kj}} (x_j - x_j^{(n)}) + [C \mathbf{x}^{(n)}]_k \right)
\]

M-step: Minimizing \( \phi_j(x_j; \mathbf{x}^{(n)}) + \beta R_j(x_j; \mathbf{x}^{(n)}) \) yields the iteration:

\[
x_j^{(n+1)} = \frac{x_j^{(n)} \sum_{i=1}^{nd} a_{ij} y_i^{(n)}}{B_j + \sqrt{B_j^2 + \left( x_j^{(n)} \sum_{i=1}^{nd} a_{ij} y_i^{(n)} \right) \left( \beta \sum_k c_{kj}^2 / \gamma_{kj} \right)}}
\]

where \( B_j \triangleq \frac{1}{2} \left[ \sum_{i=1}^{nd} a_{ij} + \beta \sum_k \left( c_{kj} [C \mathbf{x}^{(n)}]_k - \frac{c_{kj}^2}{\gamma_{kj}} x_j^{(n)} \right) \right] \), \( j = 1, \ldots, np \)

and \( \bar{y}_i^{(n)} = [A \mathbf{x}^{(n)}]_i + r_i \).

Advantages: Intrinsically monotone, nonnegativity, fully parallelizable. Requires only a couple % more computation per iteration than ML-EM

Disadvantages: Slow convergence (like EM) due to separable surrogate
Ordered Subsets Algorithms

aka block iterative or incremental gradient algorithms

The gradient appears in essentially every algorithm:

\[ L(x) = \sum_{i=1}^{n_d} h_i([Ax]_i) \implies \frac{\partial}{\partial x_j} L(x) = \sum_{i=1}^{n_d} a_{ij} \dot{h}_i([Ax]_i). \]

This is a backprojection of a sinogram of the derivatives \( \{\dot{h}_i([Ax]_i)\} \).

Intuition: with half the angular sampling, this backprojection would be fairly similar

\[
\frac{1}{n_d} \sum_{i=1}^{n_d} a_{ij} \dot{h}_i(\cdot) \approx \frac{1}{|S|} \sum_{i \in S} a_{ij} \dot{h}_i(\cdot),
\]

where \( S \) is a subset of the rays.

To “OS-ize” an algorithm, replace all backprojections with partial sums.

Recall typical iteration:

\[ x^{(n+1)} = x^{(n)} - D(x^{(n)}) \nabla \Psi(x^{(n)}). \]
Two subset case: $\Psi(x) = f_1(x) + f_2(x)$ (e.g., odd and even projection views).

For $x^{(n)}$ far from $x^*$, even partial gradients should point roughly towards $x^*$. For $x^{(n)}$ near $x^*$, however, $\nabla \Psi(x) \approx 0$, so $\nabla f_1(x) \approx -\nabla f_2(x) \implies$ cycles!

Issues. “Subset gradient balance”: $\nabla \Psi(x) \approx M \nabla f_k(x)$. Choice of ordering.
Incremental Gradients (WLS, 2 Subsets)

\[ \nabla f_{\text{WLS}}(x) \]
\[ 2 \cdot \nabla f_{\text{even}}(x) \]
\[ 2 \cdot \nabla f_{\text{odd}}(x) \]

\[ \chi^{\text{true}} \]
\[ \chi^0 \]

\[ \text{difference} \]
\[ M=2 \]

(full – subset)
Subset Gradient Imbalance

\[ \nabla f_{WLS}(x) \]

\[ 2 \cdot \nabla f_{0-90}(x) \]

\[ 2 \cdot \nabla f_{90-180}(x) \]

\[ x^0 \]

\[ \text{difference} \]

\[ \text{difference} \]

\[ \text{(full – subset)} \]
Problems with OS-EM

- Non-monotone
- Does not converge (may cycle)
- Byrne’s “rescaled block iterative” (RBI) approach converges only for consistent (noiseless) data
- \( \therefore \) unpredictable
  - What resolution after \( n \) iterations? Object-dependent, spatially nonuniform
  - What variance after \( n \) iterations?
  - ROI variance? (e.g., for Huesman’s WLS kinetics)
OSEM vs Penalized Likelihood

- 64 × 62 image
- 66 × 60 sinogram
- $10^6$ counts
- 15% randoms/scatter
- uniform attenuation
- contrast in cold region
- within-region $\sigma$ opposite side
Contrast-Noise Results

- OSEM 1 subset
- OSEM 4 subset
- OSEM 16 subset
- PL−PSCA

(64 angles)
Horizontal Profile

- OSEM 4 subsets, 5 iterations
- PL-PSCA 10 iterations
Making OS Methods Converge

- Relaxation
- Incrementalism

Relaxed block-iterative methods

$$\Psi(x) = \sum_{m=1}^{M} \Psi_m(x)$$

$$x^{(n+(m+1)/M)} = x^{(n+m/M)} - \alpha_n D(x^{(n+m/M)}) \nabla \Psi_m(x^{(n+m/M)}), \quad m = 0, \ldots, M - 1$$

Relaxation of step sizes:

$$\alpha_n \to 0 \text{ as } n \to \infty, \quad \sum_n \alpha_n = \infty, \quad \sum_n \alpha_n^2 < \infty$$

- ART
- RAMLA, BSREM (De Pierro, T-MI, 1997, 2001)
- Ahn and Fessler, NSS/MIC 2001, T-MI 2003

Considerations

- Proper relaxation can induce convergence, but still lacks monotonicity.
- Choice of relaxation schedule requires experimentation.
- $$\Psi_m(x) = \ell_m(x) + \frac{1}{M} R(x)$$, so each $$\Psi_m$$ includes part of the likelihood yet all of $$R$$
Relaxed OS-SPS

![Graph showing penalized likelihood increase over iterations for Original OS-SPS, Modified BSREM, and Relaxed OS-SPS.](image)
Incremental Methods

Incremental EM applied to emission tomography by Hsiao et al. as C-OSEM

Incremental optimization transfer (Ahn & Fessler, MIC 2004)

Find majorizing surrogate for each sub-objective function:

\[ \phi_m(x; x) = \Psi_m(x), \quad \forall x \]
\[ \phi_m(x; \bar{x}) \geq \Psi_m(x), \quad \forall x, \bar{x} \]

Define the following augmented cost function:

\[ F(x; \bar{x}_1, \ldots, \bar{x}_M) = \sum_{m=1}^{M} \phi_m(x; \bar{x}_m). \]

Fact: by construction \( \hat{x} = \arg \min_x \Psi(x) = \arg \min_x \min_{\bar{x}_1,\ldots,\bar{x}_M} F(x; \bar{x}_1, \ldots, \bar{x}_M). \)

Alternating minimization: for \( m = 1, \ldots, M \):

\[ x^{\text{new}} = \arg \min_x F\left(x; \bar{x}_1^{(n+1)}, \ldots, \bar{x}_{m-1}^{(n+1)}, \bar{x}_m^{(n)}, \bar{x}_{m+1}^{(n)}, \ldots, \bar{x}_M^{(n)}\right) \]
\[ \bar{x}_m^{(n+1)} = \arg \min_{\bar{x}_m} F\left(x^{\text{new}}, \bar{x}_1^{(n+1)}, \ldots, \bar{x}_{m-1}^{(n+1)}, \bar{x}_m, \bar{x}_{m+1}^{(n)}, \ldots, \bar{x}_M^{(n)}\right) = x^{\text{new}}. \]

- Use all current information, but increment the surrogate for only one subset.
- Monotone in \( F \), converges under reasonable assumptions on \( \Psi \)
- In contrast, OS-EM uses the information from only one subset at a time
Transmission incremental optimization transfer (TRIOT)

64 subsets, initialized with uniform image

2 iterations of OS–SPS included
TRIOT Example: Attenuation Map Images

OS-SPS: 64 subsets, 20 iterations, one point of the limit cycle
TRIOT-PC: 64 subsets, 20 iterations, after 2 iterations of OS-SPS)
Ordered subsets version of separable paraboloidal surrogates for PET transmission problem with nonquadratic convex regularization.

Matlab m-file [http://www.eecs.umich.edu/~fessler/code/transmission/tpl_osps.m](http://www.eecs.umich.edu/~fessler/code/transmission/tpl_osps.m)
Precomputed curvatures for OS-SPS

Separable Paraboloidal Surrogate (SPS) Algorithm:

\[
x_j^{(n+1)} = \left[ x_j^{(n)} - \sum_{i=1}^{n_d} a_{ij} \frac{\hat{h}_i \left( [Ax^{(n)}]_i \right)}{\sum_{i=1}^{n_d} a_{ij} |a| c_i^{(n)}} \right] + , \quad j = 1, \ldots, n_p
\]

Ordered-subsets abandons monotonicity, so why use optimal curvatures \( c_i^{(n)} \)?

Precomputed curvature:

\[
c_i = \ddot{h}_i (\hat{l}_i), \quad \hat{l}_i = \arg \min_l h_i (l)
\]

Precomputed denominator (saves one backprojection each iteration!):

\[
d_j = \sum_{i=1}^{n_d} a_{ij} |a| c_i, \quad j = 1, \ldots, n_p.
\]

OS-SPS algorithm with \( M \) subsets:

\[
x_j^{(n+1)} = \left[ x_j^{(n)} - \frac{\sum_{i \in S^{(n)}} a_{ij} \hat{h}_i \left( [Ax^{(n)}]_i \right)}{d_j / M} \right] + , \quad j = 1, \ldots, n_p
\]
Summary of Algorithms

- General-purpose optimization algorithms
- Optimization transfer for image reconstruction algorithms
- Separable surrogates $\implies$ high curvatures $\implies$ slow convergence
- Ordered subsets accelerate *initial* convergence
  require relaxation or incrementalism for true convergence
- Principles apply to emission and transmission reconstruction
- Still work to be done...

Matlab/Freemat “image reconstruction toolbox” online:
http://www.eecs.umich.edu/~fessler/code

An Open Problem

Still no algorithm with all of the following properties:
- Nonnegativity easy
- Fast converging
- Intrinsically monotone global convergence
- Accepts any type of system matrix
- Parallelizable
Part 4. Performance Characteristics

- Spatial resolution properties
- Noise properties
- Detection properties
Spatial Resolution Properties

Choosing $\beta$ can be painful, so ...

For true minimization methods:

$$\hat{x} = \arg \min_x \Psi(x)$$

the *local impulse response* is approximately (Fessler and Rogers, T-MI, 1996):

$$l_j(x) = \lim_{\delta \to 0} \frac{E[\hat{x} | x + \delta e_j] - E[\hat{x} | x]}{\delta} \approx \left[ -\nabla^2 \Psi \right]^{-1} \nabla^1 \Psi \frac{\partial}{\partial x_j} \tilde{y}(x).$$

Depends only on chosen *cost function* and *statistical model*. Independent of optimization algorithm (if iterated “to convergence”).

- Enables prediction of resolution properties (provided $\Psi$ is minimized)
- Useful for designing regularization penalty functions with desired resolution properties. For penalized likelihood:

$$l_j(x) \approx [A'WA + \beta R]^{-1} A'WAx^{\text{true}}.$$ 

- Helps choose $\beta$ for desired spatial resolution
Modified Penalty Example, PET

a) filtered backprojection
b) Penalized unweighted least-squares
c) PWLS with conventional regularization
d) PWLS with certainty-based penalty [36]
e) PWLS with modified penalty [139]
Modified Penalty Example, SPECT - Noiseless

Target filtered object  FBP  Conventional PWLS

Truncated EM  Post-filtered EM  Modified Regularization
Modified Penalty Example, SPECT - Noisy

Target filtered object

FBP

Conventional PWLS

Truncated EM

Post-filtered EM

Modified Regularization
Regularized vs Post-filtered, with Matched PSF

Noise Comparisons at the Center Pixel

- Uniformity Corrected FBP
- Penalized–Likelihood
- Post–Smoothed ML
Reconstruction Noise Properties

For unconstrained (converged) minimization methods, the estimator is implicit:

$$\hat{x} = \hat{x}(y) = \arg\min_x \Psi(x, y) .$$

What is $\text{Cov}\{\hat{x}\}$? New simpler derivation.

Denote the column gradient by $g(x, y) \triangleq \nabla_x \Psi(x, y)$.

Ignoring constraints, the gradient is zero at the minimizer: $g(\hat{x}(y), y) = 0$.

First-order Taylor series expansion:

$$g(\hat{x}, y) \approx g(x_{\text{true}}, y) + \nabla_x g(x_{\text{true}}, y)(\hat{x} - x_{\text{true}})$$

Equating to zero:

$$\hat{x} \approx x_{\text{true}} - \left[\nabla^2_x \Psi(x_{\text{true}}, y)\right]^{-1} \nabla_x \Psi(x_{\text{true}}, y) .$$

If the Hessian $\nabla^2_{x} \Psi$ is weakly dependent on $y$, then

$$\text{Cov}\{\hat{x}\} \approx \left[\nabla^2_x \Psi(x_{\text{true}}, y)\right]^{-1} \text{Cov}\{\nabla_x \Psi(x_{\text{true}}, y)\} \left[\nabla^2_x \Psi(x_{\text{true}}, y)\right]^{-1} .$$

If we further linearize w.r.t. the data: $g(x, y) \approx g(x, \bar{y}) + \nabla_y g(x, \bar{y})(y - \bar{y})$, then

$$\text{Cov}\{\hat{x}\} \approx \left[\nabla^2_x \Psi\right]^{-1} (\nabla_x \nabla_y \Psi) \text{Cov}\{y\} (\nabla_x \nabla_y \Psi)' \left[\nabla^2_x \Psi\right]^{-1} .$$
Covariance Continued

Covariance approximation:

\[
\text{Cov}\{\hat{x}\} \approx \left[ \nabla^2_x \Psi(x^{\text{true}}, \bar{y}) \right]^{-1} \text{Cov}\{ \nabla_x \Psi(x^{\text{true}}, y) \} \left[ \nabla^2_x \Psi(x^{\text{true}}, \bar{y}) \right]^{-1}
\]

Depends only on chosen cost function and statistical model. Independent of optimization algorithm.

- Enables prediction of noise properties
- Can make variance images
- Useful for computing ROI variance (e.g., for weighted kinetic fitting)
- Good variance prediction for quadratic regularization in nonzero regions
- Inaccurate for nonquadratic penalties, or in nearly-zero regions
Qi and Huesman’s Detection Analysis

SNR of MAP reconstruction > SNR of FBP reconstruction (T-MI, Aug. 2001)

quadratic regularization
SKE/BKE task
prewhitened observer
non-prewhitened observer

Open issues

Choice of regularizer to optimize detectability?
Active work in several groups.
(e.g., 2004 MIC poster by Yendiki & Fessler.)
References


The literature on image reconstruction is enormous and growing. Many valuable publications are not included in this list, which is not intended to be comprehensive.

Slides and lecture notes available from: [http://www.eecs.umich.edu/~fessler](http://www.eecs.umich.edu/~fessler)