Chapter 16

Emission Image Reconstruction (Regularized)

ch,empl

todo

Contents

s,empl,intro

16.1 Introduction (s,empl,intro)

This *chapter* describes regularized versions of the ML algorithms described in the previous chapter for reconstructing emission images from Poisson sinogram measurements.

s,empl,state

16.2 Problem statement (s,empl,state)

As described in §15.2, the goal is to estimate the coefficients (voxel values) $\boldsymbol{x} = (x_1, \ldots, x_{n_\mathrm{p}})$ from projection measurements $\boldsymbol{y} = (y_1, \ldots, y_{n_\mathrm{d}})$, under the usual Poisson statistical model described in Chapter 6, *i.e.*,

$$y_i \sim \mathsf{Poisson}\{\bar{y}_i(\boldsymbol{x})\}, \qquad \bar{y}_i(\boldsymbol{x}) \triangleq [\boldsymbol{A}\boldsymbol{x}]_i + \bar{r}_i, \qquad i = 1, \ldots, n_\mathrm{d}.$$

The negative log-likelihood is:

e,L,emis

$$Ł(\boldsymbol{x}) \equiv \sum_{i=1}^{n_\mathrm{d}} \bar{y}_i(\boldsymbol{x}) - y_i \log \bar{y}_i(\boldsymbol{x}), \tag{16.2.1}$$

neglecting constants independent of $\boldsymbol{x}$. For penalized-likelihood image reconstruction, one seeks the image that minimizes a cost function as follows:

e,obj,pl

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x} \succeq \boldsymbol{0}} \Psi(\boldsymbol{x}), \qquad \Psi(\boldsymbol{x}) = Ł(\boldsymbol{x}) + \mathsf{R}(\boldsymbol{x}), \tag{16.2.2}$$

where $\mathsf{R}(\boldsymbol{x})$ is a roughness penalty included for regularization. This penalty controls the resolution/noise tradeoff, as elaborated in Chapter 22.

Most of the algorithms described in this chapter are examples of optimization transfer methods as described in Chapter 11. As seen in Chapter 15, there are many possible derivations of "the EM" algorithm for ML emission reconstruction. Similarly, one could derive the algorithms in this chapter from a variety of perspectives.

s,empl,uniq

16.3 Uniqueness (s,empl,uniq)

todo: show that Ψ has a unique minimizer if null space of $\boldsymbol{A}$ and $\boldsymbol{C}$ are disjoint and every ψ_k is strictly convex.

s,empl,em

16.4 EM-based algorithms (s,empl,em)

The EM surrogate function described in §15.5.3 serves as a reasonable starting point for deriving penalized-likelihood emission reconstruction algorithms. As derived in (15.5.10) and (15.5.7), the surrogate is

e,empl,em,Qem

$$Q_\mathrm{EM}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) = \sum_{j=1}^{n_\mathrm{p}} Q_j(x_j; \boldsymbol{x}^{(n)}) \tag{16.4.1}$$

e,empl,em,Qj

$$Q_j(x_j; \boldsymbol{x}^{(n)}) = (x_j + \gamma_j) a_j - e_j(\boldsymbol{x}^{(n)})(x_j^{(n)} + \gamma_j) \log(x_j + \gamma_j) \tag{16.4.2}$$

e,empl,em,ejn

$$e_j^{(n)} \triangleq e_j(\boldsymbol{x}^{(n)}) = \sum_{i \in \mathcal{I}_+} a_{ij} y_i / \bar{y}_i(\boldsymbol{x}^{(n)}) \tag{16.4.3}$$

(see (15.4.3)). Recall that $a_j \triangleq \sum_{i=1}^{n_\mathrm{d}} a_{ij}$. One can use $\gamma_j = 0$ for simplicity; otherwise the γ_j's must satisfy (15.4.6). The surrogate Q_EM satisfies the usual majorization conditions (11.1.3):

$$\begin{aligned} Ł(\boldsymbol{x}) &\leq Q_\mathrm{EM}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) \\ Ł(\boldsymbol{x}) &= Q_\mathrm{EM}(\boldsymbol{x}; \boldsymbol{x}) \,. \end{aligned}$$

Using this EM surrogate, one can construct a surrogate function for the cost function Ψ as follows:

$$\phi_\mathrm{EM}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) = Q_\mathrm{EM}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) + \mathsf{R}(\boldsymbol{x}) = \sum_{j=1}^{n_\mathrm{p}} Q_j(x_j; \boldsymbol{x}^{(n)}) + \mathsf{R}(\boldsymbol{x}),$$

in which case the M-step (11.1.1) of an optimization transfer algorithm becomes

e,empl,em,sur

$$\boldsymbol{x}^{(n+1)} = \arg\min_{\boldsymbol{x} \succeq \boldsymbol{0}} \phi_\mathrm{EM}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) \,. \tag{16.4.4}$$

One could attempt to find this minimizer by zeroing the gradient of ϕ_{EM}:

e,empl,em,der,xj

$$\frac{\partial}{\partial x_j} \phi_{\mathrm{EM}}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) = a_j - e_j^{(n)} \frac{x_j^{(n)} + \gamma_j}{x_j + \gamma_j} + \frac{\partial}{\partial x_j} \mathsf{R}(\boldsymbol{x}) \,. \tag{16.4.5}$$

Indeed, if $\mathsf{R}(\boldsymbol{x})$ is a *separable penalty* function, such as the gamma prior used in [1], then often one can find the minimizer $\boldsymbol{x}^{(n+1)}$ of the surrogate ϕ_{EM} analytically. But for roughness penalty functions, $\mathsf{R}(\boldsymbol{x})$ is a *nonseparable* function of $\boldsymbol{x}$, so zeroing the above partial derivatives leads to a *coupled* set of equations with no analytical solution. Thus, additional analysis is needed to find method for minimizing the surrogate $\phi_{\mathrm{EM}}(\boldsymbol{x}; \boldsymbol{x}^{(n)})$. In fact, usually we cannot find the exact minimizer of ϕ_{EM}, so we must use an iterative approach. In other words, we must use subiterations to *descend* ϕ_{EM} within the outer iteration over n. This approach is still guaranteed to monotonically decrease Ψ. Technically speaking, this type of approach belongs to the family of *generalized EM* (*GEM*) methods described in §11.7.5.

The basic structure of any such GEM algorithm is as follows, where M denotes the number of subiterations.

```
for n = 0, 1, 2, ... {
    Compute e_j^(n) using (16.4.3). (This requires forward- / back-projection.)
    x^(n,0) := x^(n)
    for m = 0, 1, ..., M − 1 {
```

e,empl,em,descend

$$\text{Find } \boldsymbol{x}^{(n,m+1)} \text{ such that } \phi_{\mathrm{EM}}\big(\boldsymbol{x}^{(n,m+1)}; \boldsymbol{x}^{(n)}\big) \le \phi_{\mathrm{EM}}(\boldsymbol{x}^{(n,m)}; \boldsymbol{x}^{(n)}) \tag{16.4.6}$$

```
    }
    x^(n+1) := x^(n,M)
}
```

The key step of this algorithm is the descent (16.4.6), and there are a variety of possible strategies.

s,empl,em,gem

16.4.1 Coordinate descent GEM algorithm (s,empl,em,gem)

One of the earliest GEM methods used the coordinate descent algorithm (§10.11) to descend ϕ_{EM} [2]. This approach is neither particularly fast nor parallelizable, so it is used infrequently.

s,empl,em,osl

16.4.2 Green's one step late (OSL) algorithm (s,empl,em,osl)

An *ad hoc* method for attempting to solve the coupled system of equations (16.4.5) is to replace the partial derivative of $\mathsf{R}(\boldsymbol{x})$ with the partial derivative *evaluated at the previous iteration*. This is the *one step late* (*OSL*) approach proposed by Green [3, 4], in which one "solves"

$$0 = a_j - e_j^{(n)} \frac{x_j^{(n)} + \gamma_j}{x_j + \gamma_j} + \frac{\partial}{\partial x_j} \mathsf{R}(\boldsymbol{x}^{(n)}),$$

leading to the iterative update

e,empl,em,osl

$$x_j^{(n+1)} = \left[(x_j^{(n)} + \gamma_j) \frac{e_j^{(n)}}{a_j + \frac{\partial}{\partial x_j} \mathsf{R}(\boldsymbol{x}^{(n)})} - \gamma_j \right]_+ . \tag{16.4.7}$$

This algorithm is *not* guaranteed to converge or to decrease the cost function Ψ monotonically. Indeed, it is not even well defined because the denominator $a_j + \frac{\partial}{\partial x_j} \mathsf{R}(\boldsymbol{x}^{(n)})$ can be zero or negative. Although this algorithm has the appeal of looking very "EM like" in its simplicity, it should be avoided because more recent methods are only slightly more complicated to implement but have better stability and convergence properties.

s,empl,dp

16.4.3 De Pierro's parallelizable GEM algorithms (s,empl,dp)

This section describes slight generalizations of the parallelizable penalized-likelihood algorithms for emission tomography developed by De Pierro [5]. The algorithm(s) described here are perhaps the most natural generalizations of the ML-EM algorithm to the regularized case, and the ordered-subset versions are a very reasonable practical choice. De Pierro's paper focused on the case of quadratic regularization. Using the quadratic surrogates developed by Huber [6, p. 184], as described in §11.4.4.4, it is easy to generalize De Pierro's method to the case of nonquadratic potential functions. This was done without fanfare in [7], for example, and detailed later in [8]. Here we consider a very general family of nonquadratic potential functions in which quadratics are simply a special case. We also generalize De Pierro's algorithm to use the "γ_j" factors described in (15.4.6).

We focus on penalty functions having the following form:

$$\mathsf{R}(\boldsymbol{x}) = \sum_{k=1}^{K} \psi_k([\boldsymbol{C}\boldsymbol{x}]_k),$$

where $\boldsymbol{C} = \{c_{kj}\}$ is a $K \times n_{\mathrm{p}}$ matrix. We assume that each potential function ψ_k has a quadratic surrogate. So for each k there exists a nonnegative curvature function $\breve{c}_k(\cdot)$ such that the parabola

$$q_k(t;s) = \psi_k(s) + \dot{\psi}_k(s)(t-s) + \frac{1}{2}\,\breve{c}_k(s)(t-s)^2$$

majorizes ψ_k, *i.e.*, $q_k(t;s) \geq \psi_k(t)$ for all $t, s \geq 0$ (*cf.* §11.4), and $q_k(t;t) = \psi_k(t)$. The simplest case is when the potential functions are themselves quadratic, *i.e.*, $\psi_k(t) = \beta_k t^2/2$, in which case $\breve{c}_k = \beta_k$, but the algorithm presented below accommodates the more general nonquadratic case easily.

De Pierro's optimization transfer algorithm is a GEM algorithm of the form (16.4.6). For performing the descent monotonically, we follow De Pierro and use optimization transfer by first finding a *separable* quadratic surrogate $\mathsf{R}_{\mathrm{SQS}}(\boldsymbol{x};\bar{\boldsymbol{x}})$ for the penalty function $\mathsf{R}(\boldsymbol{x})$, where $\bar{\boldsymbol{x}}$ is shorthand for $\boldsymbol{x}^{(n,m)}$. We then implement (16.4.6) as follows:

e,empl,dp,inner,min

$$\begin{aligned} \boldsymbol{x}^{(n,m+1)} &= \arg\min_{\boldsymbol{x} \succeq \boldsymbol{0}} \phi_{\mathrm{DP}}(\boldsymbol{x};\boldsymbol{x}^{(n,m)};\boldsymbol{x}^{(n)}) \\ \phi_{\mathrm{DP}}(\boldsymbol{x};\bar{\boldsymbol{x}};\boldsymbol{x}^{(n)}) &\triangleq Q_{\mathrm{EM}}(\boldsymbol{x};\boldsymbol{x}^{(n)}) + \mathsf{R}_{\mathrm{SQS}}(\boldsymbol{x};\bar{\boldsymbol{x}}) \,. \end{aligned} \tag{16.4.8}$$

Our construction of $\mathsf{R}_{\mathrm{SQS}}(\boldsymbol{x};\bar{\boldsymbol{x}})$ parallels §11.5.7.

Because each potential function ψ_k has a quadratic surrogate q_k by assumption, we first construct a (nonseparable) quadratic surrogate for $\mathsf{R}(\boldsymbol{x})$ as follows:

$$\mathsf{R}(\boldsymbol{x}) = \sum_{k=1}^{K} \psi_k([\boldsymbol{C}\boldsymbol{x}]_k) \leq \mathsf{R}_{\mathrm{Q}}(\boldsymbol{x};\bar{\boldsymbol{x}}) \triangleq \sum_{k=1}^{K} q_k([\boldsymbol{C}\boldsymbol{x}]_k;[\boldsymbol{C}\bar{\boldsymbol{x}}]_k)\,.$$

Now we use De Pierro's additivity trick [5] as described in §11.5.7:

$$[\boldsymbol{C}\boldsymbol{x}]_k = \sum_{j=1}^{n_{\mathrm{p}}} c_{kj} x_j = \sum_{j=1}^{n_{\mathrm{p}}} \gamma_{kj} \left[\frac{c_{kj}}{\gamma_{kj}}(x_j - \bar{x}_j) + [\boldsymbol{C}\bar{\boldsymbol{x}}]_k \right],$$

where we must choose factors $\gamma_{kj} \geq 0$ for which $\sum_{j=1}^{n_{\mathrm{p}}} \gamma_{kj} = 1$. Because q_k is convex, by the convexity inequality (25.3.5):

$$\begin{aligned} q_k([\boldsymbol{C}\boldsymbol{x}]_k;s) &= q_k\left(\sum_{j=1}^{n_{\mathrm{p}}} \gamma_{kj} \left[\frac{c_{kj}}{\gamma_{kj}}(x_j - \bar{x}_j) + [\boldsymbol{C}\bar{\boldsymbol{x}}]_k \right]; s \right) \\ &\leq \sum_{j=1}^{n_{\mathrm{p}}} \gamma_{kj}\, q_k\left(\frac{c_{kj}}{\gamma_{kj}}(x_j - \bar{x}_j) + [\boldsymbol{C}\bar{\boldsymbol{x}}]_k; s \right), \end{aligned}$$

for any $s \in \mathbb{R}$. Summing these leads to the following separable quadratic surrogate for the penalty function:

e,empl,dp,Rsqs

$$\mathsf{R}_{\mathrm{Q}}(\boldsymbol{x};\bar{\boldsymbol{x}}) \quad \leq \quad \mathsf{R}_{\mathrm{SQS}}(\boldsymbol{x};\bar{\boldsymbol{x}}) \triangleq \sum_{j=1}^{n_{\mathrm{p}}} R_j(x_j;\bar{\boldsymbol{x}}) \tag{16.4.9}$$

e,empl,dp,Rj

$$R_j(x_j;\bar{\boldsymbol{x}}) \quad \triangleq \quad \sum_{k=1}^{K} \gamma_{kj}\, q_k\left(\frac{c_{kj}}{\gamma_{kj}}(x_j - \bar{x}_j) + [\boldsymbol{C}\bar{\boldsymbol{x}}]_k; [\boldsymbol{C}\bar{\boldsymbol{x}}]_k\right), \tag{16.4.10}$$

the derivatives of which are

e,empl,dp,Rj,der

$$\begin{aligned}\frac{\partial}{\partial x_j} R_j(x_j;\bar{\boldsymbol{x}}) &= \sum_{k=1}^{K} c_{kj}\left[\dot{\psi}_k([\boldsymbol{C}\bar{\boldsymbol{x}}]_k) + \frac{c_{kj}}{\gamma_{kj}}\,\check{c}_k([\boldsymbol{C}\bar{\boldsymbol{x}}]_k)(x_j - \bar{x}_j)\right] \\ &= \frac{\partial}{\partial x_j}\,\mathsf{R}(\bar{\boldsymbol{x}}) + (x_j - \bar{x}_j)\, r_j(\bar{\boldsymbol{x}}),\end{aligned} \tag{16.4.11}$$

where we define the following penalty curvature function:

e,empl,dp,rj

$$r_j(\bar{\boldsymbol{x}}) \triangleq \sum_{k=1}^{K} \frac{c_{kj}^2}{\gamma_{kj}}\,\check{c}_k([\boldsymbol{C}\bar{\boldsymbol{x}}]_k)\,. \tag{16.4.12}$$

Because both the surrogate for the negative log-likelihood $Ł(\boldsymbol{x})$ and the surrogate $\mathsf{R}_{\mathrm{SQS}}$ in (16.4.9) for the penalty are separable, the inner minimization (16.4.8) becomes the following parallelizable update:

e,empl,dp,xj,argmin

$$x_j^{(n,m+1)} \quad = \quad \arg\min_{x_j \geq 0} \phi_j(x_j;\boldsymbol{x}^{(n,m)};\boldsymbol{x}^{(n)}), \qquad j = 1,\ldots,n_{\mathrm{p}} \tag{16.4.13}$$

e,empl,dp,surj

$$\phi_j(x_j;\bar{\boldsymbol{x}};\boldsymbol{x}^{(n)}) \quad \triangleq \quad Q_j(x_j;\boldsymbol{x}^{(n)}) + R_j(x_j;\bar{\boldsymbol{x}}). \tag{16.4.14}$$

Fortuitously for De Pierro (and us), we can find the minimizer of ϕ_j *analytically* by zeroing its derivative:

$$\begin{aligned}\frac{\partial}{\partial x_j}\,\phi_j(x_j;\bar{\boldsymbol{x}};\boldsymbol{x}^{(n)}) &= a_j - e_j^{(n)}\frac{x_j^{(n)} + \gamma_j}{x_j + \gamma_j} + \frac{\partial}{\partial x_j}\,\mathsf{R}(\bar{\boldsymbol{x}}) + (x_j - \bar{x}_j)\, r_j(\bar{\boldsymbol{x}}) \\ &= \frac{1}{x_j + \gamma_j}\left[(x_j + \gamma_j)^2\, r_j(\bar{\boldsymbol{x}}) + 2(x_j + \gamma_j)\, b_j(\bar{\boldsymbol{x}}) - e_j^{(n)}\,(x_j^{(n)} + \gamma_j)\right],\end{aligned}$$

using (16.4.11), where we define

e,empl,dp,bj

$$b_j(\bar{\boldsymbol{x}}) \triangleq \frac{1}{2}\left[a_j + \frac{\partial}{\partial x_j}\,\mathsf{R}(\bar{\boldsymbol{x}}) - (\bar{x}_j + \gamma_j)\, r_j(\bar{\boldsymbol{x}})\right]. \tag{16.4.15}$$

Thus, zeroing the derivative is equivalent to finding the appropriate root of the following quadratic formula:

$$0 = (x_j + \gamma_j)^2\, r_j(\bar{\boldsymbol{x}}) + 2(x_j + \gamma_j)\, b_j(\bar{\boldsymbol{x}}) - e_j^{(n)}\,(x_j^{(n)} + \gamma_j),$$

paying appropriate attention to the nonnegativity constraint in (16.4.13). We write the solution as follows:

e,empl,dp,xj,root

$$x_j^{(n,m+1)} = \left[\mathrm{root}\big(r_j(\boldsymbol{x}^{(n,m)}),\ b_j(\boldsymbol{x}^{(n,m)}),\ e_j^{(n)}(x_j^{(n)} + \gamma_j)\big) - \gamma_j\right]_+, \tag{16.4.16}$$

where $\mathrm{root}(\alpha,\beta,\gamma)$ returns the nonnegative root of $0 = \alpha x^2 + 2\beta x - \gamma$, or equivalently of $0 = \alpha + 2\beta x_j^{-1} - \gamma x_j^{-2}$, for $\alpha \geq 0$. In particular, using the numerically stable forms [9, p. 183]:

$$\mathrm{root}(\alpha,\beta,\gamma) = \begin{cases} \dfrac{\sqrt{\beta^2 + \alpha\gamma} - \beta}{\alpha}, & \alpha > 0,\ \beta < 0 \\ \dfrac{\gamma}{2\beta}, & \alpha = 0\ (\beta \neq 0) \\ \dfrac{\gamma}{\sqrt{\beta^2 + \alpha\gamma} + \beta}, & \alpha > 0,\ \beta \geq 0. \end{cases}$$

IRT *See* `eql_root.m`.

To synopsize, the essence of (our generalization of) De Pierro's algorithm is the update (16.4.16) with (16.4.15).

16.4.3.1 Simplifications

A recommended choice for γ_{kj} is the following:

e,empl,dp,ck

$$\gamma_{kj} = \frac{|c_{kj}|}{c_k}, \qquad c_k \triangleq \sum_{j=1}^{n_\mathrm{p}} |c_{kj}|, \tag{16.4.17}$$

for which the penalty curvatures r_j in (16.4.12) "simplify" to

$$r_j(\bar{\boldsymbol{x}}) = \sum_{k=1}^{K} |c_{kj}|\, c_k\, \breve{c}_k([\boldsymbol{C}\bar{\boldsymbol{x}}]_k)\,.$$

It can be somewhat inconvenient to have to recompute $r_j(\boldsymbol{x}^{(n,m)})$ every subiteration within a GEM algorithm. Most potential functions have quadratic surrogates with bounded curvatures, *i.e.*, $0 \le \breve{c}_k(s) \le \breve{c}_k^{\max}$, $\forall s \in \mathbb{R}$. In such cases, an alternative to continuously recomputing $r_j(\cdot)$ is to use the following upper bound on each curvature:

e,empl,dp,rj,max

$$r_j^{\max} \triangleq \sum_{k=1}^{K} |c_{kj}|\, c_k\, \breve{c}_k^{\max}\,. \tag{16.4.18}$$

Because these curvatures are independent of the iterates, one can precompute them prior to iterating. Such precomputation is always advisable for quadratic potential functions, because $\breve{c}_k$ is independent of $\boldsymbol{x}$ in those cases.

x,empl,em,dp,1st **Example 16.4.1** *As a concrete example, consider a 2D $N_1 \times N_2$ image $\boldsymbol{x}$ and roughness penalty of the form (1.10.1), based on first-order differences and a first-order neighborhood,* i.e.*,*

$$\mathsf{R}(\boldsymbol{x}) = \sum_{i_1=2}^{N_1} \sum_{i_2=1}^{N_2} \psi\big(x_{j(i_1,i_2)} - x_{j(i_1-1,i_2)}\big) + \sum_{i_1=1}^{N_1} \sum_{i_2=2}^{N_2} \psi\big(x_{j(i_1,i_2)} - x_{j(i_1,i_2-1)}\big),$$

where ψ is a potential function with maximum parabola surrogate curvature $\breve{c}^{\max} > 0$. In this case, $\boldsymbol{C}$ is a $[N_2(N_1-1) + N_1(N_2-1)] \times N_1 N_2$ matrix, as discussed in §1.10 and §1.19. Because the penalty uses first-order differences, each row of $\boldsymbol{C}$ has two nonzero entries: one $+1$ and one -1. Thus in (16.4.17) we have $c_k = 2$. For each of the $(N_1-1)(N_2-1)$ pixels that are not on the left or top edge of the image, the corresponding column of $\boldsymbol{C}$ has two $+1$ entries and two -1 entries. Thus the penalty curvatures in (16.4.18) are simply

$$r_j^{\max} = \sum_{k=1}^{K} |c_{kj}|\, 2\, \breve{c}^{\max} = 8\, \breve{c}^{\max}\,.$$

Similarly, for a 3D case using the 6 nearest neighbors, we would have $r_j^{\max} = 2 \cdot 6\, \breve{c}^{\max} = 12\, \breve{c}^{\max}$.

More generally, when using first-order differences between each pixel and M of its nearby neighbors, we have $r_j^{\max} = 2M\, \breve{c}^{\max}$. Of course in the quadratic case where $\psi(t) = t^2/2$, we have $\breve{c}^{\max} = 1$.

Consider the following modified quadratic regularizer [10]:

$$\mathsf{R}(\boldsymbol{x}) = \sum_{j=1}^{n_\mathrm{p}} \sum_{l \in \mathcal{O}} \beta_l \kappa_j \kappa_{j-l} \frac{1}{2} \left(x_j - x_{j-l}\right)^2,$$

where $\mathcal{O}$ denotes the set of pixel index offsets corresponding to the neighbors. Each row of $\boldsymbol{C}$ has one entry $\sqrt{\beta_l \kappa_j \kappa_{j-l}}$ and the only other nonzero entry is $-\sqrt{\beta_l \kappa_j \kappa_{j-l}}$. When the kth row of $\boldsymbol{C}$ corresponds to pair $\{j, j-l\}$, we have $c_k = 2\sqrt{\beta_l \kappa_j \kappa_{j-l}}$. Thus $r_j^{\max} = 2 \sum_{l \in \mathcal{O}} \beta_l \kappa_j \kappa_{j-l}$.

16.4.3.2 Ordered-subsets

To implement an ordered subsets version of this algorithm that uses n_subset equally sized blocks, we replace $e_j^{(n)}$ in the update (16.4.16) with its incremental version:

$$e_j^{(n)} \approx n_\mathrm{subset} \sum_{i \in \mathcal{S}} a_{ij} \frac{y_i}{\bar{y}_i(\boldsymbol{x}^{(n)})},$$

where $\mathcal{S} \subset \{1, \ldots, n_\mathrm{d}\}$ denotes the subset of measurements to be used.

A complication is that the penalty function is involved in the update for every subset, so an efficient implementation of the penalty function is important.

In [11] we showed that one can add relaxation to the certain OS algorithms for penalized-likelihood image reconstruction. However, that analysis did *not* include the algorithm described herein. The natural approach to introducing relaxation would be something like:

$$x_j^{(n,m+1)} = \left[x_j^{(n,m)} + \alpha_n \left(\mathrm{root}\big(r_j(\boldsymbol{x}^{(n,m)}),\ b_j(\boldsymbol{x}^{(n,m)}),\ e_j^{(n)}(x_j^{(n)} + \gamma_j) \big) - (x_j^{(n,m+1)} - \gamma_j) \right) \right]_+ .$$

The convergence properties of this variation are an *open problem.*

IRT *See* `eql_os_emdp.m` *for the case of quadratic regularization, and* `epl_os_emdp.m` *for the nonquadratic case.*

s,empl,em,gam

16.4.4 Gamma prior (s,empl,em,gam)

The M-step is simple in the case of a *gamma prior*, *i.e.*,

$$\mathsf{R}(\boldsymbol{x}) = \sum_{j=1}^{n_\mathrm{p}} \beta_j x_j - \zeta_j \log x_j ,$$

where β_j and ζ_j are free parameters. In this case

$$\frac{\partial}{\partial x_j} \phi_{\mathrm{EM}}(\boldsymbol{x}; \boldsymbol{x}^{(n)}) = a_j - e_j^{(n)} \frac{x_j^{(n)} + \gamma_j}{x_j + \gamma_j} + \beta_j - \frac{\zeta_j}{x_j} .$$

For $\gamma_j = 0$, the minimization (16.4.4) is simply

$$x_j^{(n+1)} = \frac{e_j^{(n)} x_j^{(n)} + \zeta_j}{a_j + \beta_j} .$$

For $\gamma_j \neq 0$, one can find the minimizer by solving a simple quadratic formula.

The primary virtue of the gamma prior would seem to be its simplicity of implementation.

s,empl,ps

16.5 Quadratic surrogate approaches (s,empl,ps)

The methods described in the previous section were based on the natural separable EM surrogate Q_{EM} for the negative log-likelihood $\text{Ł}(\boldsymbol{x})$. An alternative approach is to apply the optimization transfer principle using quadratic surrogates for $\text{Ł}(\boldsymbol{x})$.

An advantage of quadratic surrogates is that the M-step is somewhat simpler than for the EM surrogate. The disadvantage is that our construction of the quadratic surrogate holds only when $\bar{r}_i > 0$, *i.e.*, when there is a strictly nonzero background contribution (*e.g.*, from randoms or scatter). And the surrogate curvature can be large (leading to slow convergence rate) if the $\bar{r}_i$'s are close to zero. "Fortunately," the randoms and scatter fractions in emission tomography are often sufficiently large that $\bar{r}_i \gg 0$. But for cases where one to ignores (tisk tisk) or precorrects by subtraction (ditto) the contributions of background events, the EM surrogate may be preferable to the quadratic surrogate described here.

todo: does most of this belong in c-eml?

As described *above*, an approach to maximizing the emission log-likelihood (statistical surrogate) based on the following simple idea, illustrated in one dimension in Fig. 16.5.1. Because the log-likelihood $\mathsf{L}(\boldsymbol{x})$ is difficult to maximize directly, we endeavor to find a *surrogate function* $Q(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ that is easier to maximize, and maximize that function at the nth iteration, *i.e.*:

e,empl,iter

$$\boldsymbol{x}^{(n+1)} = \arg\max_{\boldsymbol{x} \succeq \boldsymbol{0}} Q(\boldsymbol{x}; \boldsymbol{x}^{(n)}) - \beta R(\boldsymbol{x}) \tag{16.5.1}$$

If we choose the sequence of surrogate functions $Q(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ properly, then the sequence of iterates $\{\boldsymbol{x}^{(n)}\}$ will converge to the maximizer $\hat{\boldsymbol{x}}$.

The SAGE algorithm is indirectly based on this idea; the expected conditional log-likelihood of the "complete" data space given the observed data is a type of surrogate function that indeed satisfies the conditions sufficient to ensure convergence [12]. However, the statistical construction of the surrogate functions for the SAGE and other EM algorithms can seem somewhat mysterious.

In this section, we describe an alternate approach to constructing surrogate functions that uses only basic calculus principles. We can rewrite the log-likelihood $\mathsf{L}(\boldsymbol{x})$ in (16.2.1) as follows:

$$\mathsf{L}(\boldsymbol{x}) = \sum_{i=1}^{n_{\mathrm{d}}} \mathsf{h}_i([\boldsymbol{G}\boldsymbol{x}]_i)$$

where

$$[\boldsymbol{G}\boldsymbol{x}]_i = \sum_{j=1}^{n_{\mathrm{p}}} g_{ij} x_j$$

is the "geometric" projection of the object along the ith ray, and

$$\mathsf{h}_i(l) = y_i \log(c_i l + \bar{r}_i) - (c_i l + \bar{r}_i)$$

is the *marginal log-likelihood* of the ith measurement. The h_i functions are concave, and strictly concave if $y_i > 0$.

To choose the surrogate function Q, we first find one-dimensional parabolic surrogate functions $q_i\big(l; \ell_i^{(n)}\big)$, as illustrated in Fig. 16.5.1, where

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

We then combine these 1D surrogate functions to form an overall surrogate function as follows:

e,empl,Q

$$\mathsf{L}(\boldsymbol{x}) \geq Q(\boldsymbol{x}; \boldsymbol{x}^{(n)}) \triangleq \sum_{i=1}^{n_{\mathrm{d}}} q_i\big([\boldsymbol{G}\boldsymbol{x}]_i; \ell_i^{(n)}\big) \,. \tag{16.5.2}$$

Provided that the 1D surrogate functions satisfy the following three properties:

- $q_i\big(\ell_i^{(n)}; \ell_i^{(n)}\big) = \mathsf{h}_i\big(\ell_i^{(n)}\big)$,
- $\dot{q}_i\big(\ell_i^{(n)}; \ell_i^{(n)}\big) = \dot{\mathsf{h}}_i\big(\ell_i^{(n)}\big)$
- $q_i\big(l; \ell_i^{(n)}\big) \leq \mathsf{h}_i(l), \ \forall l \geq 0$,

then it can be shown that the recursive algorithm given by (16.5.1) will monotonically increase $\mathsf{L}(\boldsymbol{x})$, and in fact can be shown to converge globally by a proof similar to that in [12].

Using the fact that the h_i's are concave and that the first derivatives of the h_i's are convex, one can show [13] that the above three conditions will be satisfied if we choose parabolic surrogate functions as follows:

e,empl,qi

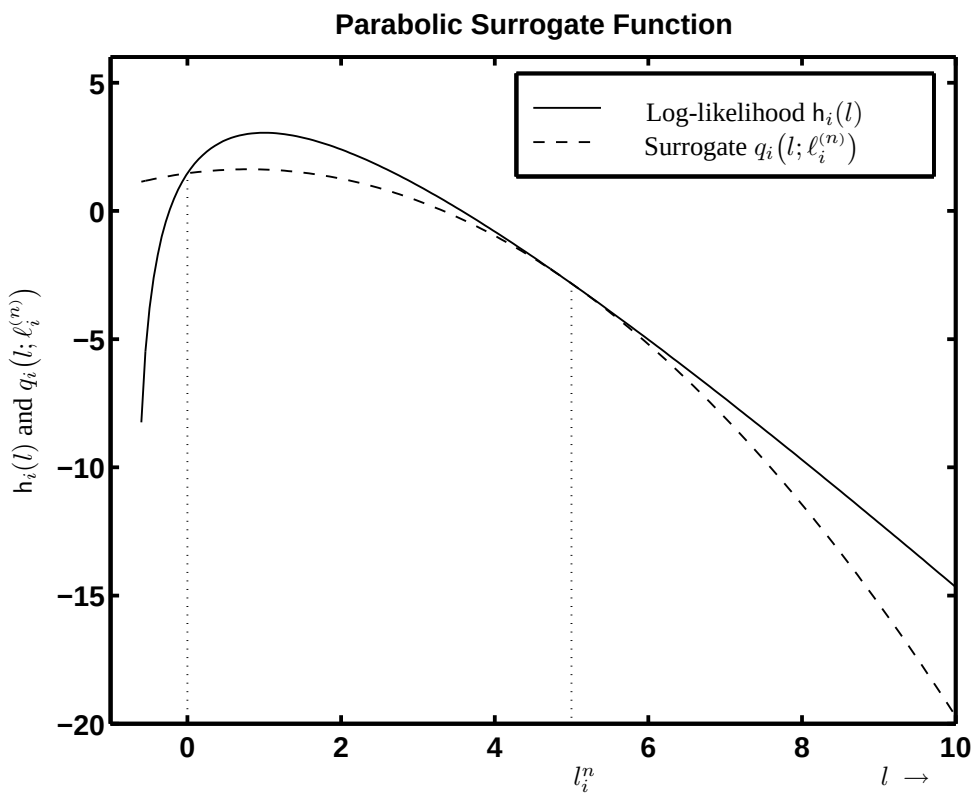
$$q_i\big(l;\ell_i^{(n)}\big) = \mathsf{h}_i\big(\ell_i^{(n)}\big) + \dot{\mathsf{h}}_i\big(\ell_i^{(n)}\big)(l-\ell_i^{(n)}) - \frac{1}{2}n_i(\ell_i^{(n)})\left(l-\ell_i^{(n)}\right)^2, \tag{16.5.3}$$

where

e,empl,ni

$$n_i(l) \triangleq \begin{cases} \dfrac{2}{l^2}\Big[\mathsf{h}_i(l) - \mathsf{h}_i(0) - l\,\dot{\mathsf{h}}_i(l)\Big], & l > 0 \\ -\ddot{\mathsf{h}}_i(0), & l = 0. \end{cases} \tag{16.5.4}$$

The basic idea is illustrated in Fig. 16.5.1. The parabolic surrogate function $q_i\big(l;\ell_i^{(n)}\big)$ has the same value as the marginal log-likelihood $\mathsf{h}_i(l)$ at the current projection value $l = \ell_i^{(n)}$, and has the same slope at that point. This is evident from (16.5.3). In addition, the parabolic function lies *below* $\mathsf{h}_i(l)$ for all nonnegative l. This is the key to having a monotonic algorithm[1]. The proof that this q_i choice satisfies the third of the above three conditions is somewhat detailed, and is described in [13]. When the parabolic q_i functions are "assembled" as in (16.5.2) to form an overall surrogate function, the final form of Q is a paraboloid, *i.e.*, a quadratic form. One can maximize this quadratic form by any number of different methods. (Exact maximization is not necessary, due to the iteration (16.5.1).)



fig,hiqi

Figure 16.5.1: Illustration of 1D parabolic surrogate function. Note that $q_i\big(l;\ell_i^{(n)}\big) \le \mathsf{h}_i(l)$ for $l \ge 0$.

16.5.1 Computing the n_i's for the Poisson emission model

$$h(l) = y\log(cl+r) - (cl+r)$$

[1] One could also find *approximating* parabolas using Newton's method, but these parabolas can cross $\mathsf{h}_i(l)$, and the result is a nonmonotonic algorithm that is not guaranteed to converge. Our construction using (16.5.4) avoids this problem.

$$\dot{\mathsf{h}}(l) = c\left[\frac{y}{cl+r} - 1\right]$$
$$-\ddot{\mathsf{h}}(l) = c^2 \frac{y}{(cl+r)^2}$$

Assuming that $r > 0$, the curvature from (e,hakan) is

$$n(l) = \begin{cases} \frac{2}{l^2}\left[h(l) - h(0) - l\,\dot{\mathsf{h}}(l)\right], & l > 0 \\ -\ddot{\mathsf{h}}(l), & l = 0 \end{cases} = c^2 \begin{cases} \frac{2}{(cl)^2} y f(cl), & l > 0 \\ y/r^2, & l = 0 \end{cases}$$

where

$$f(t) \triangleq \log\frac{t+r}{r} - \frac{t}{t+r}$$

Can we show $n(l) > 0$ for $l > 0$? It suffices to show that $f(t) > 0$ for $t > 0$.

For $t > 0$

$$\dot{f}(t) = \frac{1}{t+r} - \frac{(t+r)-t}{(t+r)^2} = \frac{t}{(t+r)^2},$$

so

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\frac{f(t)}{t^2} &= \frac{t^2\dot{f}(t) - 2tf(t)}{t^4} = \frac{t\dot{f}(t) - 2f(t)}{t^3} = \frac{1}{t^3}\left[\frac{t^2}{(t+r)^2} - 2\left(\log\frac{t+r}{r} - \frac{t}{t+r}\right)\right] \\ &= \frac{1}{t^3}\left[\frac{t^2 + 2t(t+r)}{(t+r)^2} - 2\log\frac{t+r}{r}\right]. \end{aligned}$$

In particular,

$$\dot{n}(0) =$$

needs work!

s,empl,sps

16.5.2 Separable Paraboloidal Surrogates (E-ML-SPS) Algorithm (s,empl,sps)

In the spirit of the fully parallelizable algorithms developed in §(s,ls,em), we would also like a fully parallelizable algorithm for the E-ML problem. To derive such an algorithm, we begin with the paraboloidal surrogate (16.5.2), and use the convexity of its constituent parabolas to form a separable surrogate function.

From (13.6.21)

$$[\boldsymbol{G}\boldsymbol{x}]_i = \sum_{j=1}^{n_\mathrm{p}} g_{ij} x_j = \sum_{j=1}^{n_\mathrm{p}} \pi_{ij}\left[\frac{1}{\pi_{ij}} g_{ij}(x_j - x_j^{(n)}) + [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i\right]$$

where the π_{ij}'s are any nonnegative constants for which $\sum_{j=1}^{n_\mathrm{p}} \pi_{ij} = 1, \forall i$, as in (13.6.22).

THIS NEEDS TO BE CHECKED SINCE THE PARABOLOID IS A SURROGATE ONLY FOR $l >= 0$!!!!

Thus, due to the convexity of parabolas:

$$\begin{aligned} q_i\big([\boldsymbol{G}\boldsymbol{x}]_i; \ell_i^{(n)}\big) &= q_i\left(\sum_{j=1}^{n_\mathrm{p}} \pi_{ij}\left[\frac{1}{\pi_{ij}} g_{ij}(x_j - x_j^{(n)}) + [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i\right]; \ell_i^{(n)}\right) \\ &\geq \sum_{j=1}^{n_\mathrm{p}} \pi_{ij}\, q_i\left(\frac{1}{\pi_{ij}} g_{ij}(x_j - x_j^{(n)}) + [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i; \ell_i^{(n)}\right). \end{aligned}$$

$$Q(\boldsymbol{x}; \boldsymbol{x}^{(n)}) \geq \sum_{j=1}^{n_\mathrm{p}} Q_j(x_j; \boldsymbol{x}^{(n)})$$

where

e,empl,sps,Qj

$$Q_j(x_j; \boldsymbol{x}^{(n)}) \triangleq \sum_{j=1}^{n_\mathrm{p}} \pi_{ij}\, q_i\left(\frac{1}{\pi_{ij}} g_{ij}(x_j - x_j^{(n)}) + [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i; \ell_i^{(n)}\right) \tag{16.5.5}$$

$$x_j^{(n+1)} = \arg\max_{x_j \geq 0} Q_j(x_j; \boldsymbol{x}^{(n)}), \qquad j = 1, \ldots, n_\mathrm{p}.$$

Because Q_j is quadratic, it is maximized by Newton's formula (see (newton)):

$$x_j^{(n+1)} = \left[x_j^{(n)} + \frac{\frac{\partial}{\partial x_j} Q_j(x_j; \boldsymbol{x}^{(n)}) \Big|_{x_j = x_j^{(n)}}}{-\frac{\partial^2}{\partial x_j^2} Q_j(x_j; \boldsymbol{x}^{(n)})} \right]_+,$$

which will be the nonnegative-constrained maximizer of Q_j regardless of the value of $x_j^{(n)}$.

In particular, from (16.5.5):

$$\frac{\partial}{\partial x_j} Q_j(x_j; \boldsymbol{x}^{(n)}) = \sum_{i=1}^{n_\mathrm{d}} g_{ij} \, \dot{q}_i \left(\frac{1}{\pi_{ij}} g_{ij}(x_j - x_j^{(n)}) + [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i \right)$$

$$-\frac{\partial^2}{\partial x_j^2} Q_j(x_j; \boldsymbol{x}^{(n)}) = -\sum_{i=1}^{n_\mathrm{d}} \frac{g_{ij}^2}{\pi_{ij}} \, \ddot{q}_i \left(\frac{1}{\pi_{ij}} g_{ij}(x_j - x_j^{(n)}) + [\boldsymbol{A}\boldsymbol{x}^{(n)}]_i \right)$$

$$\begin{aligned} \frac{\partial}{\partial x_j} Q_j(x_j; \boldsymbol{x}^{(n)}) \Big|_{x_j = x_j^{(n)}} &= \sum_{i=1}^{n_\mathrm{d}} g_{ij} \, \dot{q}_i([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i) \\ &= \sum_{i=1}^{n_\mathrm{d}} g_{ij} \, \dot{\mathsf{h}}_i([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i) = \frac{\partial}{\partial x_j} \Phi(\boldsymbol{x}) \Big|_{\boldsymbol{x} = \boldsymbol{x}^{(n)}} \end{aligned}$$

and

$$d_j \triangleq -\frac{\partial^2}{\partial x_j^2} Q_j(x_j; \boldsymbol{x}^{(n)}) = \sum_{i=1}^{n_\mathrm{d}} \frac{g_{ij}^2}{\pi_{ij}} n_i(\ell_i^{(n)}). \tag{16.5.6}$$ e,empl,sps,dj

Thus the general EPL-SPS algorithm is

$$x_j^{(n+1)} = x_j^{(n)} + \frac{\frac{\partial}{\partial x_j} \Phi(\boldsymbol{x}) \Big|_{\boldsymbol{x} = \boldsymbol{x}^{(n)}}}{d_j},$$

which can be expressed in matrix vector form as

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \boldsymbol{D}^{-1} \nabla \Phi(\boldsymbol{x}^{(n)}) \tag{16.5.7}$$ e,alg,empl,sps

where

$$\boldsymbol{D} = \mathsf{diag}\{d_j\}$$

and d_j is defined in (16.5.6).

This E-ML-SPS algorithm is fully parallelizable (we can update all pixels in parallel), it easily accommodates both the nonnegativity constraint, and is easily extended to include both quadratic and nonquadratic penalty functions (see (e,empl)).

In words, the algorithm alternates between two steps. The first step is to find the coefficients of the surrogate parabolas in (16.5.3) using (16.5.4). This involves a single-pass over the sinogram with trivial computation.

Then the paraboloid (16.5.2) is partially maximized by a single cycle of coordinate ascent.

This step requires roughly the equivalent of one forward and one backprojection, similar to most iterative algorithms.

As described in §(e,ls,alg,sps), a reasonable choice for the π_{ij}'s is $\pi_{ij} = \frac{g_{ij}}{g_i}$ where $g_i \triangleq \sum_{j=1}^{n_\mathrm{p}} g_{ij}$. In this case we can explicitly write the E-ML-SPS algorithm as follows:

$$x_j^{(n+1)} = \left[x_j^{(n)} + \frac{\sum_{i=1}^{n_\mathrm{d}} g_{ij} \, \dot{\mathsf{h}}_i([\boldsymbol{A}\boldsymbol{x}^{(n)}]_i)}{\sum_{i=1}^{n_\mathrm{d}} g_{ij} g_i n_i(\ell_i^{(n)})} \right]_+ = \left[x_j^{(n)} + \frac{\sum_{i=1}^{n_\mathrm{d}} g_{ij} c_i \left[\frac{y_i}{c_i[\boldsymbol{A}\boldsymbol{x}^{(n)}]_i + \bar{r}_i} - 1 \right]}{\sum_{i=1}^{n_\mathrm{d}} g_{ij} g_i n_i(\ell_i^{(n)})} \right]_+.$$

16.5.3 Paraboloidal Surrogates Coordinate Ascent (E-ML-PSCA) Algorithm

16.6 Abstract

We present a new algorithm for penalized-likelihood emission image reconstruction. The algorithm monotonically increases the objective function, converges globally to the unique maximizer, and easily accommodates the nonnegativity constraint and nonquadratic but convex penalty functions. The algorithm is based on finding paraboloidal surrogate functions for the log-likelihood at each iteration: quadratic functions that are tangent to the log-likelihood at the current image estimate, and lie below the log-likelihood over the entire nonnegative orthant. These conditions ensure monotonicity. The paraboloidal surrogates are maximized easily using existing algorithms such as coordinate ascent. Simulation results show that the proposed algorithm converges faster than the SAGE algorithm, yet the new algorithm is somewhat easier to implement.

s,empl,alg,psca

16.7 The New Algorithm (s,empl,alg,psca)

We had previously recommended the SAGE algorithm [12] as a fast globally-convergent algorithm for this problem. The construction of the SAGE algorithm requires certain minimizations that are somewhat unusual in the tomographic literature, and must be implemented carefully to achieve reasonable CPU time per iteration.

The algorithm described below requires less CPU time per iteration than the space-alternating generalize EM (SAGE) algorithm [12].

For reconstruction problems (such as 2D PET and SPECT) where the system matrix $\boldsymbol{G}$ can be precomputed and stored, we recommend this new algorithm over our previously published algorithms for penalized-likelihood reconstruction. For reconstruction problems where the system matrix is represented in factored form [14], methods that can better exploit this representation, such as the conjugate-gradient algorithm, appear to remain preferable.

The new algorithm we propose is based on the following simple idea, illustrated in one dimension in Fig. 16.5.1. Because the log-likelihood $\mathsf{L}(\boldsymbol{x})$ is difficult to maximize directly, we endeavor to find a *surrogate function* $Q(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ that is easier to maximize, and maximize that function at the nth iteration, *i.e.*:

e,iter

$$\boldsymbol{x}^{(n+1)} = \arg\max_{\boldsymbol{x} \succeq \boldsymbol{0}} Q(\boldsymbol{x}; \boldsymbol{x}^{(n)}) - \beta R(\boldsymbol{x}) \tag{16.7.1}$$

If we choose the sequence of surrogate functions $Q(\boldsymbol{x}; \boldsymbol{x}^{(n)})$ properly, then the sequence of iterates $\{\boldsymbol{x}^{(n)}\}$ will converge to the maximizer $\hat{\boldsymbol{x}}$ The SAGE algorithm is indirectly based on this idea; the expected conditional log-likelihood of the "complete" data space given the observed data is a type of surrogate function that indeed satisfies the conditions sufficient to ensure convergence [12]. However, the statistical construction of the surrogate functions for the SAGE and other EM algorithms can seem somewhat mysterious.

In this paper, we propose a new approach to constructing surrogate functions that uses only basic calculus principles. We can rewrite the log-likelihood $\mathsf{L}(\boldsymbol{x})$ in (16.2.1) as follows:

$$\mathsf{L}(\boldsymbol{x}) = \sum_{i=1}^{n_\mathrm{d}} \mathsf{h}_i([\boldsymbol{A}\boldsymbol{x}]_i)$$

where

$$[\boldsymbol{A}\boldsymbol{x}]_i = \sum_{j=1}^{n_\mathrm{p}} a_{ij} x_j$$

is the projection of the object along the ith ray, and

$$\mathsf{h}_i(l) = y_i \log(l + \bar{r}_i) - (l + \bar{r}_i)$$

is the *marginal log-likelihood* of the ith measurement. The h_i functions are concave, and strictly concave if $y_i > 0$.

Our new proposed strategy for choosing the surrogate function Q is to first find one-dimensional parabolic surrogate functions $q_i\big(l; \ell_i^{(n)}\big)$, as illustrated in Fig. 16.5.1, where

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

We then combine these 1D surrogate functions to form an overall surrogate function as follows:

e,Q

$$Q(\boldsymbol{x};\boldsymbol{x}^{(n)}) \triangleq \sum_{i=1}^{n_{\mathrm{d}}} q_i\big([\boldsymbol{A}\boldsymbol{x}]_i\,;\ell_i^{(n)}\big)\,. \tag{16.7.2}$$

Provided that the 1D surrogate functions satisfy the following three properties:
- $q_i\big(\ell_i^{(n)};\ell_i^{(n)}\big) = \mathsf{h}_i\big(\ell_i^{(n)}\big)$,
- $\dot{q}_i\big(\ell_i^{(n)};\ell_i^{(n)}\big) = \dot{\mathsf{h}}_i\big(\ell_i^{(n)}\big)$
- $q_i\big(l;\ell_i^{(n)}\big) \leq \mathsf{h}_i(l),\ \forall l \geq 0$,

then it can be shown that the recursive algorithm given by (16.7.1) will monotonically increase $\mathsf{L}(\boldsymbol{x})$, and in fact can be shown to converge globally by a proof similar to that in [12].

Using the fact that the h_i's are concave and that the first derivatives of the h_i's are convex, one can show [13] that the above three conditions will be satisfied if we choose parabolic surrogate functions as follows:

e,qi

$$q_i\big(l;\ell_i^{(n)}\big) = \mathsf{h}_i\big(\ell_i^{(n)}\big) + \dot{\mathsf{h}}_i\big(\ell_i^{(n)}\big)(l-\ell_i^{(n)}) - \frac{1}{2}n_i(\ell_i^{(n)})(l-\ell_i^{(n)})^2, \tag{16.7.3}$$

where

e,ni

$$n_i(l) \triangleq \begin{cases} \dfrac{2}{l^2}\Big[\mathsf{h}_i(l) - h(0) - l\,\dot{\mathsf{h}}_i(l)\Big], & l > 0 \\ -\ddot{\mathsf{h}}_i(l), & l = 0. \end{cases} \tag{16.7.4}$$

The basic idea is illustrated in Fig. 16.5.1. The parabolic surrogate function $q_i\big(l;\ell_i^{(n)}\big)$ has the same value as the marginal log-likelihood $\mathsf{h}_i(l)$ at the current projection value $l = \ell_i^{(n)}$, and has the same slope at that point. This is evident from (16.7.3). In addition, the parabolic function lies *below* $\mathsf{h}_i(l)$ for all nonnegative l. This is the key to having a monotonic algorithm[2]. The proof that this q_i choice satisfies the third of the above three conditions is somewhat detailed, and is described in [13]. When the parabolic q_i functions are "assembled" as in (16.7.2) to form an overall surrogate function, the final form of Q is a paraboloid, *i.e.*, a quadratic form. One can maximize this quadratic form by any number of different methods. (Exact maximization is not necessary, due to the iteration (16.7.1).) We have chosen to use successive over-relaxation or coordinate ascent [15] for this maximization, because it easily accommodates both the nonnegativity constraint as well as nonquadratic penalty functions.

In words, the algorithm alternates between two steps. The first step is to find the coefficients of the surrogate parabolas in (16.7.3) using (16.7.4). This involves a single-pass over the sinogram with trivial computation. Then the paraboloid (16.7.2) is partially maximized by a single cycle of coordinate ascent. This step requires roughly the equivalent of one forward and one backprojection, similar to most iterative algorithms.

s,empl,result

16.8 Results (s,empl,result)

We used the same brain emission simulation reported in [12] to evaluate the proposed algorithm. In [12], the SAGE algorithm was compared to many alternatives, including Green's one-step late (OSL) method [3], Kaufman's bounded line search [16], and the generalized EM (GEM) algorithm of Hebert and Leahy [2]. We found that the SAGE algorithm converged faster than all of the above methods, so here we focus on comparing the proposed algorithm to just the SAGE algorithm.

Figure 16.9.1 plots the increase in the log-likelihood $\Phi(\boldsymbol{x}^{(n)}) - \Phi(\boldsymbol{x}^{(0)})$ versus CPU time on a DEC AlphaStation 600 5/333 workstation. The proposed algorithm (paraboloidal surrogates coordinate ascent (PSCA) algorithm) converges a little bit faster than SAGE, in part because it requires about 10% less CPU time per iteration, and in part because it increases Φ more each iteration. The differences are modest because both algorithms converge quite quickly, so there is limited room remaining for improvement. The new algorithm is simpler to implement than SAGE as well.

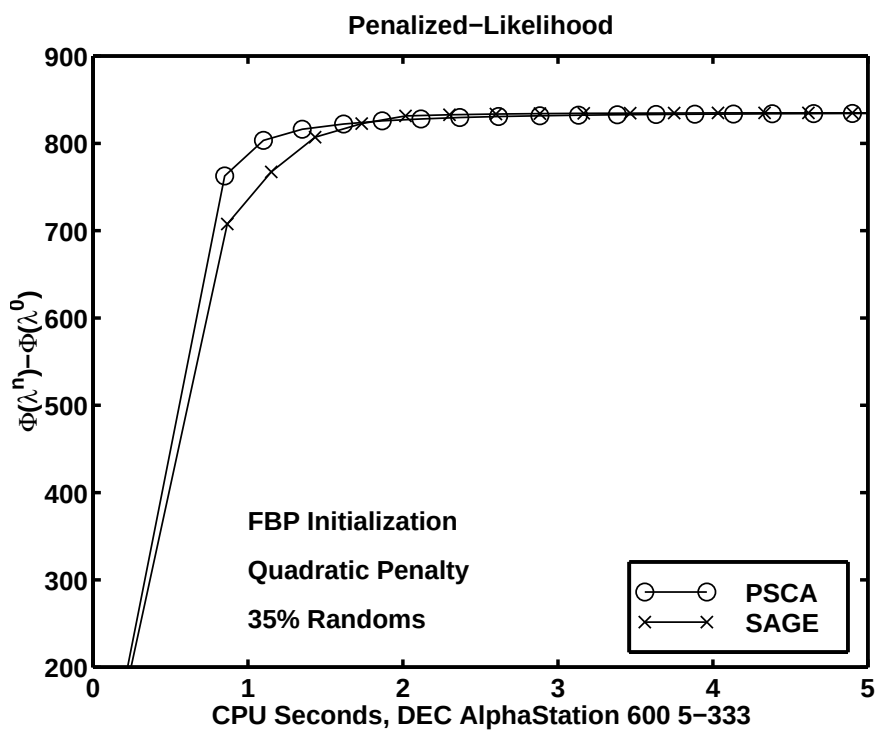
The resulting images are indistinguishable from those shown in [12] because we used the same objective function and both PSCA and SAGE algorithms are globally convergent.

[2] One could also find *approximating* parabolas using Newton's method, but these parabolas can cross $\mathsf{h}_i(l)$, and the result is a nonmonotonic algorithm that is not guaranteed to converge. Our construction using (16.7.4) avoids this problem.

s,empl,disc

16.9 Discussion (s,empl,disc)

The proposed algorithm is based on the "optimization transfer" principle. Because the original objective function Φ is cannot be maximized directly, we instead maximize a sequence of surrogate functions $\phi^{(n)}(\boldsymbol{x})$. The key is to choose surrogate functions that are easier to maximize than Φ, but have low curvature (high curvature surrogate functions lead to slow convergence rate [13,17]). Many algorithms in the literature are based (implicitly or explicitly) on optimization transfer ideas, include EM, SAGE, grouped coordinate ascent [18], the convex algorithm [19], and ISRA [20–22]. In almost all cases, the surrogate functions are *separable*, which makes them trivial to maximize, but also means that they have very high curvature and hence poor convergence rate. The paraboloidal surrogate functions that we have proposed here and in [13] are the first that we are aware of that are nonseparable. Generally nonseparable functions are harder to maximize. Fortunately, a notable exception is quadratic surrogate functions, which is the choice we have made. This choice may be suboptimal; it is possible that there exist other nonseparable surrogate functions that are easily maximized but have even lower curvature and hence yield faster convergence rates.



fig,cpu

Figure 16.9.1: Comparison of convergence rate of proposed paraboloidal surrogate coordinate ascent (PSCA) algorithm versus SAGE algorithm.

s,empl,other

16.10 Other stuff (s,empl,other)

Most of the other methods for developing reconstruction algorithms described in this *chapter* have counterparts for the emission problem. Monotonic acceleration is possible using line searches [16]. Replacing the sums over i in (16.12.4) with sums over subsets of the projections yields the emission OSEM algorithm [23]; see also the related variants RAMLA [24] and RBBI [25,26]. Although the OSEM algorithm fails to converge in general, it often gives reasonable looking images in a small number of iterations when initialized with a uniform image. Sequential updates rather than parallel updates leads to the fast converging SAGE algorithms [12] and coordinate ascent algorithms [27], including paraboloidal surrogate variations thereof [28]. The conjugate gradient algorithm has been applied extensively to the emission problem and is particularly effective provided one carefully treats the nonnegativity constraints [29].

s,empl,amp

16.11 Alternating minimization procedure

s,empl,alg,em

16.12 ML algorithms todo? (s,empl,alg,em)

16.12.1 EM Algorithm

One can derive the classical EM algorithm for the emission problem by a formal complete-data exposition [30], which is less complicated than the transmission case but still somewhat mysterious to many readers, or by fixed-point considerations [31] (which do not fully illustrate the monotonicity of the emission EM algorithm). Instead, we adopt the simple concavity-based derivation of De Pierro [22], which reinforces the surrogate function concepts woven throughout this chapter.

The key to the derivation is the following "multiplicative" trick, which applies if $x_j^{(n)} > 0$:

e,mult

$$[\boldsymbol{A}\boldsymbol{x}]_i + \bar{r}_i = \sum_{j=1}^{n_\mathrm{p}} \left(\frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right) \frac{x_j}{x_j^{(n)}} \bar{y}_i^{(n)} + \left(\frac{\bar{r}_i}{\bar{y}_i^{(n)}} \right) \bar{y}_i^{(n)}. \tag{16.12.1}$$

The $n_\mathrm{p} + 1$ terms in parentheses are nonnegative and sum to unity, so we can apply the concavity inequality. Because

$$g_i(m) \triangleq y_i \log m - m$$

is concave on $(0, \infty)$, it follows that

$$\begin{aligned}
\mathsf{L}(\boldsymbol{x}) &= \sum_{i=1}^{n_\mathrm{d}} g_i([\boldsymbol{A}\boldsymbol{x}]_i + \bar{r}_i) \\
&= \sum_{i=1}^{n_\mathrm{d}} g_i \left(\sum_{j=1}^{n_\mathrm{p}} \left(\frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right) \frac{x_j}{x_j^{(n)}} \bar{y}_i^{(n)} + \left(\frac{\bar{r}_i}{\bar{y}_i^{(n)}} \right) \bar{y}_i^{(n)} \right) \\
&\geq \sum_{i=1}^{n_\mathrm{d}} \sum_{j=1}^{n_\mathrm{p}} \left(\frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} \right) g_i \left(\frac{x_j}{x_j^{(n)}} \bar{y}_i^{(n)} \right) + \left(\frac{\bar{r}_i}{\bar{y}_i^{(n)}} \right) g_i(\bar{y}_i^{(n)}) \triangleq Q(\boldsymbol{x}; \boldsymbol{x}^{(n)}).
\end{aligned}$$

The surrogate function Q is separable:

e,Qj,emis

$$Q(\boldsymbol{x}; \boldsymbol{x}^{(n)}) = \sum_{j=1}^{n_\mathrm{p}} Q_j(x_j; \boldsymbol{x}^{(n)}), \quad Q_j(x_j; \boldsymbol{x}^{(n)}) \equiv \sum_{i=1}^{n_\mathrm{d}} \frac{a_{ij} x_j^{(n)}}{\bar{y}_i^{(n)}} g_i \left(\frac{x_j}{x_j^{(n)}} \bar{y}_i^{(n)} \right). \tag{16.12.2}$$

Thus the the following parallelizable maximization step is guaranteed to monotonically increase the log-likelihood $\mathsf{L}(\boldsymbol{x})$ each iteration:

e,xjnn,emis

$$x_j^{(n+1)} = \arg\max_{x_j \geq 0} Q_j(x_j; \boldsymbol{x}^{(n)}). \tag{16.12.3}$$

The maximization is trivial:

$$\frac{\partial}{\partial x_j} Q_j(x_j; \boldsymbol{x}^{(n)}) = \sum_{i=1}^{n_\mathrm{d}} a_{ij} \dot{g}_i \left(\frac{x_j}{x_j^{(n)}} \bar{y}_i^{(n)} \right) = \sum_{i=1}^{n_\mathrm{d}} a_{ij} \left[\frac{x_j^{(n)}}{x_j} \frac{y_i}{\bar{y}_i^{(n)}} - 1 \right].$$

Equating to zero and solving for x_j yields the famous update:

e,alg,em,emis

$$\boxed{x_j^{(n+1)} = x_j^{(n)} \frac{\sum_{i=1}^{n_\mathrm{d}} a_{ij} y_i / \bar{y}_i^{(n)}}{\sum_{i=1}^{n_\mathrm{d}} a_{ij}}, \; j = 1, \ldots, n_\mathrm{p}.} \tag{16.12.4}$$

Unfortunately, the emission EM algorithm (16.12.4) usually converges painfully slowly. To understand this, consider the curvatures of the surrogate functions Q_j:

e,ddQj,emis

$$-\frac{\partial^2}{\partial x_j^2} Q_j(x_j; \boldsymbol{x}^{(n)}) = \frac{x_j^{(n)}}{x_j^2} \sum_{i=1}^{n_\mathrm{d}} a_{ij} \frac{y_i}{\bar{y}_i^{(n)}}. \tag{16.12.5}$$

For any pixels converging towards zero, these curvatures grow without bound. This leads to very slow convergence; even sublinear convergence rates are possible [32].

16.12.2 An improved EM algorithm

One can choose a slightly better decomposition than (16.12.1) to get slightly faster converging EM algorithms [12]. First find any set of nonnegative constants $\{\gamma_j\}_{j=1}^{n_\mathrm{p}}$ that satisfy

e,ri,csj

$$\bar{r}_i \geq \sum_{j=1}^{n_\mathrm{p}} a_{ij}\gamma_j, \forall i. \tag{16.12.6}$$

Then an alternative to (16.12.1) is:

e,mult,new

$$[\boldsymbol{Ax}]_i + \bar{r}_i = \sum_{j=1}^{n_\mathrm{p}} \left(\frac{a_{ij}(x_j^{(n)} + \gamma_j)}{\bar{y}_i^{(n)}} \right) \frac{x_j}{x_j^{(n)} + \gamma_j} \bar{y}_i^{(n)} + \left(\frac{\hat{r}_i}{\bar{y}_i^{(n)}} \right) \bar{y}_i^{(n)}, \tag{16.12.7}$$

where $\hat{r}_i = \bar{r}_i - \sum_{j=1}^{n_\mathrm{p}} a_{ij}\gamma_j \geq 0$. Again the terms in parentheses in (16.12.7) are nonnegative and sum to unity. So a similar derivation as that yielding (16.12.2) leads to a new surrogate function:

$$Q_j(x_j; \boldsymbol{x}^{(n)}) = \sum_{i=1}^{n_\mathrm{d}} \frac{a_{ij}(x_j^{(n)} + \gamma_j)}{\bar{y}_i^{(n)}} g_i \left(\frac{x_j + \gamma_j}{x_j^{(n)}} \bar{y}_i^{(n)} \right).$$

Maximizing as in (16.12.3) leads to the following algorithm

e,alg,em,emis,3

$$\boxed{x_j^{(n+1)} = \left[(x_j^{(n)} + \gamma_j) \frac{\sum_{i=1}^{n_\mathrm{d}} a_{ij} y_i / \bar{y}_i^{(n)}}{\sum_{i=1}^{n_\mathrm{d}} a_{ij}} - \gamma_j \right]_+, \; j = 1, \ldots, n_\mathrm{p}.} \tag{16.12.8}$$

This algorithm was derived by a more complicated EM approach in [12], and called ML-EM-3. The surrogate function derivation is simpler to present and understand, and more readily generalizable to alternative surrogates.

The curvatures of the second Q_j's just derived are smaller than those in (16.12.2), due to the γ_j's (replace x_j with $x_j + \gamma_j$ in the denominator of (16.12.5)). The convergence rate improves as the γ_j's increase, but of course (16.12.6) must be satisfied to ensure monotonicity. Because the EM algorithm updates all parameters simultaneously, the γ_j values must be "shared" among all pixels, and typically are fairly small due to (16.12.6). In contrast the SAGE algorithm [12] updates the pixels sequentially, which greatly relaxes the constraints on the γ_j's, allowing larger values and hence faster convergence rates.

s,empl,note

16.13 Notes (s,empl,note)

[33] - regularized? and look at T-MI block Fisher paper

[34] C-OSEM for MAP (incremental EM)

Bibliography

lange:87:ats [1] K. Lange, M. Bahn, and R. Little. A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *IEEE Trans. Med. Imag.*, 6(2):106–14, June 1987.

hebert:89:age [2] T. Hebert and R. Leahy. A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. *IEEE Trans. Med. Imag.*, 8(2):194–202, June 1989.

green:90:brf [3] P. J. Green. Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Trans. Med. Imag.*, 9(1):84–93, March 1990.

green:90:ouo [4] P. J. Green. On use of the EM algorithm for penalized likelihood estimation. *J. Royal Stat. Soc. Ser. B*, 52(3):443–452, 1990.

depierro:95:ame [5] A. R. De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Trans. Med. Imag.*, 14(1):132–7, March 1995.

huber:81 [6] P. J. Huber. *Robust statistics.* Wiley, New York, 1981.

sotthivirat:03:ros [7] S. Sotthivirat and J. A. Fessler. Relaxed ordered-subsets algorithm for penalized-likelihood image restoration. *J. Opt. Soc. Am. A*, 20(3):439–49, March 2003.

chang:04:rir [8] J-H. Chang, J. M. M. Anderson, and J. R. Votaw. Regularized image reconstruction algorithms for positron emission tomography. *IEEE Trans. Med. Imag.*, 23(9):1165–75, September 2004.

press:88 [9] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes in C*. Cambridge Univ. Press, New York, 1988.

fessler:96:srp [10] J. A. Fessler and W. L. Rogers. Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs. *IEEE Trans. Im. Proc.*, 5(9):1346–58, September 1996.

ahn:03:gci [11] S. Ahn and J. A. Fessler. Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms. *IEEE Trans. Med. Imag.*, 22(5):613–26, May 2003.

fessler:95:pml [12] J. A. Fessler and A. O. Hero. Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms. *IEEE Trans. Im. Proc.*, 4(10):1417–29, October 1995.

erdogan:99:maf [13] H. Erdoğan and J. A. Fessler. Monotonic algorithms for transmission tomography. *IEEE Trans. Med. Imag.*, 18(9):801–14, September 1999.

qi:98:hr3 [14] J. Qi, R. M. Leahy, S. R. Cherry, A. Chatziioannou, and T. H. Farquhar. High resolution 3D Bayesian image reconstruction using the microPET small-animal scanner. *Phys. Med. Biol.*, 43(4):1001–14, April 1998.

fessler:94:pwl [15] J. A. Fessler. Penalized weighted least-squares image reconstruction for positron emission tomography. *IEEE Trans. Med. Imag.*, 13(2):290–300, June 1994.

kaufman:87:iaa [16] L. Kaufman. Implementing and accelerating the EM algorithm for positron emission tomography. *IEEE Trans. Med. Imag.*, 6(1):37–51, March 1987.

erdogan:98:ama [17] H. Erdoğan and J. A. Fessler. Accelerated monotonic algorithms for transmission tomography. In *Proc. IEEE Intl. Conf. on Image Processing*, volume 2, pages 680–4, 1998.

fessler:97:gca [18] J. A. Fessler, E. P. Ficaro, N. H. Clinthorne, and K. Lange. Grouped-coordinate ascent algorithms for penalized-likelihood transmission image reconstruction. *IEEE Trans. Med. Imag.*, 16(2):166–75, April 1997.

lange:95:gca [19] K. Lange and J. A. Fessler. Globally convergent algorithms for maximum a posteriori transmission tomography. *IEEE Trans. Im. Proc.*, 4(10):1430–8, October 1995.

witherspoon:86:aii [20] M. E. Daube-Witherspoon and G. Muehllehner. An iterative image space reconstruction algorithm suitable for volume ECT. *IEEE Trans. Med. Imag.*, 5(2):61–66, June 1986.

ollinger:90:irr [21] J. M. Ollinger. Iterative reconstruction-reprojection and the expectation-maximization algorithm. *IEEE Trans. Med. Imag.*, 9(1):94–8, March 1990.

depierro:93:otr [22] A. R. De Pierro. On the relation between the ISRA and the EM algorithm for positron emission tomography. *IEEE Trans. Med. Imag.*, 12(2):328–33, June 1993.

hudson:94:air [23] H. M. Hudson and R. S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans. Med. Imag.*, 13(4):601–9, December 1994.

browne:96:ara [24] J. A. Browne and A. R. De Pierro. A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography. *IEEE Trans. Med. Imag.*, 15(5):687–99, October 1996.

byrne:96:bim [25] C. L. Byrne. Block-iterative methods for image reconstruction from projections. *IEEE Trans. Im. Proc.*, 5(5):792–3, May 1996.

byrne:97:cbi [26] C. L. Byrne. Convergent block-iterative algorithms for image reconstruction from inconsistent data. *IEEE Trans. Im. Proc.*, 6(9):1296–304, September 1997.

bouman:96:aua [27] C. A. Bouman and K. Sauer. A unified approach to statistical tomography using coordinate descent optimization. *IEEE Trans. Im. Proc.*, 5(3):480–92, March 1996.

fessler:98:aps [28] J. A. Fessler and H. Erdoğan. A paraboloidal surrogates algorithm for convergent penalized-likelihood emission image reconstruction. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 2, pages 1132–5, 1998.

mumcuoglu:94:fgb [29] E. U. Mumcuoglu, R. Leahy, S. R. Cherry, and Z. Zhou. Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images. *IEEE Trans. Med. Imag.*, 13(3):687–701, December 1994.

lange:84:era [30] K. Lange and R. Carson. EM reconstruction algorithms for emission and transmission tomography. *J. Comp. Assisted Tomo.*, 8(2):306–16, April 1984.

shepp:82:mlr [31] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imag.*, 1(2):113–22, October 1982.

fessler:93:ocd [32] J. A. Fessler, N. H. Clinthorne, and W. L. Rogers. On complete data spaces for PET reconstruction algorithms. *IEEE Trans. Nuc. Sci.*, 40(4):1055–61, August 1993.

hudson:94:fmo [33] H. M. Hudson, J. Ma, and P. Green. Fisher's method of scoring in statistical image reconstruction: comparison of Jacobi and Gauss-Seidel iterative schemes. *Stat. Meth. Med. Res.*, 3(1):41–61, 1994.

hsiao:02:anc [34] I-T. Hsiao, A. Rangarajand, and G. Gindi. A new convergent map reconstruction algorithm for emission tomography using ordered subsets and separable surrogates. In *Proc. IEEE Intl. Symp. Biomed. Imag.*, pages 409–12, 2002.