New Complete-Data Spaces and Faster Algorithms for Penalized-Likelihood Emission Tomography

Jeffrey A. Fessler and Alfred O. Hero
University of Michigan

Abstract

The classical expectation-maximization (EM) algorithm for image reconstruction suffers from particularly slow convergence when additive background effects such as accidental coincidences and scatter are included. In addition, when smoothness penalties are included in the objective function, the M-step of the EM algorithm becomes intractable due to parameter coupling. This paper describes the space-alternating generalized EM (SAGE) algorithm, in which the parameters are updated sequentially using a sequence of small "hidden" data spaces rather than one large complete-data space. The sequential update decouples the M-step, so the maximization can typically be performed analytically. By choosing hidden-data spaces with considerably less Fisher information than the conventional complete-data space for Poisson data, we obtain significant improvements in convergence rate. This acceleration is due to statistical considerations, not to numerical overrelaxation methods, so monotonic increases in the objective function and global convergence are guaranteed.

Due to the space constraints, we focus on the unpenalized case in this summary, and we eliminate derivations that are similar to those in [1]. See [2, 3] for a more comprehensive literature review, the penalized likelihood algorithms, and experimental results.

I. The SAGE method

Let the observation $\boldsymbol{Y}$ have the probability density function $f(\boldsymbol{y};\boldsymbol{\theta}_{\text{true}})$, where $\boldsymbol{\theta}_{\text{true}}$ is a parameter vector residing in subset Θ of the p-dimensional space $\mathbb{R}^p$. Given a measurement realization $\boldsymbol{Y}=\boldsymbol{y}$, our goal is to compute an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}_{\text{true}}$:

$$\hat{\boldsymbol{\theta}} \stackrel{\triangle}{=} \arg\max_{\theta\in\Theta} \Phi(\boldsymbol{\theta})$$

where

$$\Phi(\boldsymbol{\theta}) \stackrel{\triangle}{=} \log f(\boldsymbol{y};\boldsymbol{\theta}) - P(\boldsymbol{\theta}), \qquad (1)$$

and P is an optional smoothness penalty. Direct maximization of Φ is often intractable due to the structure of f, the coupling in P, or both.

This work was supported in part by DOE grant DE-FG02-87ER60561, NSF grant BCS-9024370, and NIH grants CA-54362-02 and CA-60711-01.

An EM algorithm updates all parameters simultaneously, which is part of the reason for its slow convergence. In contrast, a SAGE algorithm updates only a subset $S \subset \{1,\ldots,p\}$. Let $\tilde{S}$ denote the complement of the subset S. By introducing a "hidden-data" space for $\boldsymbol{\theta}_S$ based on the statistical structure of the likelihood, we replace the maximization of $\Phi(\boldsymbol{\theta}_S, \boldsymbol{\theta}^i_{\tilde{S}})$ over $\boldsymbol{\theta}_S$ with the maximization of another functional $\phi^S(\boldsymbol{\theta}_S;\boldsymbol{\theta}^i)$. If the hidden-data space is chosen wisely, then the function $\phi^S(\cdot;\boldsymbol{\theta}^i)$ can be maximized *analytically*, obviating the need for line searches. Even if ϕ^S cannot be maximized analytically, one can often choose hidden-data spaces such that $\phi^S(\cdot;\boldsymbol{\theta}^i)$ is less expensive to evaluate then $\Phi(\cdot;\boldsymbol{\theta}^i_{\tilde{S}})$, so line searches for maximizing $\phi^S(\cdot;\boldsymbol{\theta}^i)$ would be cheaper than line searches for maximizing $\Phi(\cdot;\boldsymbol{\theta}^i_{\tilde{S}})$. Just as for an EM algorithm, the functional ϕ^S is constructed to ensure that increases in ϕ^S yield increases in Φ. The SAGE method uses the underlying statistical structure of the problem to replace cumbersome or expensive numerical maximizations with analytical or simpler maximizations.

Definition 1 *A random vector $\boldsymbol{X}^S$ with probability density function $f(\boldsymbol{x};\boldsymbol{\theta})$ is an* admissible hidden-data space *with respect to $\boldsymbol{\theta}_S$ for $f(\boldsymbol{y};\boldsymbol{\theta})$ if the joint density of $\boldsymbol{X}^S$ and $\boldsymbol{Y}$ satisfies*

$$f(\boldsymbol{y},\boldsymbol{x};\boldsymbol{\theta}) = f(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}_{\tilde{S}}) f(\boldsymbol{x};\boldsymbol{\theta}), \qquad (2)$$

i.e., the conditional distribution $f(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}_{\tilde{S}})$ must be independent of $\boldsymbol{\theta}_S$.

An essential ingredient of any SAGE algorithm is the following conditional expectation of the log-likelihood of $\boldsymbol{X}^S$:

$$\begin{aligned} Q^S(\boldsymbol{\theta}_S;\bar{\boldsymbol{\theta}}) &= Q^S(\boldsymbol{\theta}_S;\bar{\boldsymbol{\theta}}_S,\bar{\boldsymbol{\theta}}_{\tilde{S}}) \\ &= E\left\{\log f(\boldsymbol{X}^S;\boldsymbol{\theta}_S,\bar{\boldsymbol{\theta}}_{\tilde{S}}) | \boldsymbol{Y}=\boldsymbol{y};\bar{\boldsymbol{\theta}}\right\} \qquad (3) \\ &= \int f(\boldsymbol{x}|\boldsymbol{Y}=\boldsymbol{y};\bar{\boldsymbol{\theta}}) \log f(\boldsymbol{x};\boldsymbol{\theta}_S,\bar{\boldsymbol{\theta}}_{\tilde{S}})\, d\boldsymbol{x}. \end{aligned}$$

We combine this expectation with the penalty function:

$$\phi^S(\boldsymbol{\theta}_S;\bar{\boldsymbol{\theta}}) \stackrel{\triangle}{=} Q^S(\boldsymbol{\theta}_S;\bar{\boldsymbol{\theta}}) - P(\boldsymbol{\theta}_S,\bar{\boldsymbol{\theta}}_{\tilde{S}}). \qquad (4)$$

Let $\boldsymbol{\theta}^0 \in \Theta$ be an initial parameter estimate. A generic SAGE algorithm produces a sequence of estimates $\{\boldsymbol{\theta}^i\}_{i=0}^{\infty}$ via the following recursion:

Generic SAGE Algorithm

`For` $i = 0, 1, \ldots$ {

1. Choose an index set S.
2. Choose an admissible hidden-data space $\boldsymbol{X}^S$ for $\boldsymbol{\theta}_S$.
3. E-step: Compute $\phi^S(\boldsymbol{\theta}_S; \boldsymbol{\theta}^i)$ using (4).
4. M-step:

$$\boldsymbol{\theta}_S^{i+1} = \arg\max_{\boldsymbol{\theta}_S} \phi^S(\boldsymbol{\theta}_S; \boldsymbol{\theta}^i), \quad (5)$$

$$\boldsymbol{\theta}_{\tilde{S}}^{i+1} = \boldsymbol{\theta}_{\tilde{S}}^i. \quad (6)$$

5. Optional: Repeat steps 3 and 4.

}.

The maximization in (5) is over the set of $\boldsymbol{\theta}_S$ satisfying $(\boldsymbol{\theta}_S, \boldsymbol{\theta}_{\tilde{S}}^i) \in \Theta$. The classical EM algorithm is the special case where $S^i = \{1, \ldots, p\}$. In emission tomography, if we choose index sets and hidden data spaces appropriately, then the E-step and M-step can be combined via an analytical maximization into a recursion of the form $\boldsymbol{\theta}_S^{i+1} = g^S(\boldsymbol{\theta}^i)$. Convergence properties of the generic SAGE algorithm are discussed in [2]. Convergence for emission tomography is shown in [3].

II. Maximum likelihood

We first review the linear Poisson model for emission tomography and summarize the classical EM algorithm (ML-EM-1) for maximizing the likelihood [1]. We then introduce a new complete-data space that leads to a new, faster converging EM algorithm: ML-EM-2. For more dramatic improvements we introduce two SAGE algorithms. The second algorithm, ML-SAGE-2, is based on much less informative hidden data-spaces, which leads to convergence rates that are faster than even a line-search accelerated EM algorithm (ML-LINU).

A. The Problem

Assume that the emission distribution can be discretized into p voxels with emission rates $\{\lambda_k\}_{k=1}^p$. Assume that the emission source is viewed by N detectors, and let N_{nk} denote the number of emissions from the kth voxel that are detected by the nth detector. Assume the variates N_{nk} have independent Poisson distributions:

$$N_{nk} \sim \text{Poisson}\{a_{nk}\lambda_k\},$$

where the a_{nk} are nonnegative constants that characterize the system [1]. The detectors record emissions from several source locations, so at best one would observe only the sums $\sum_k N_{nk}$, rather than each N_{nk}. Background emissions, random coincidences, and scatter contaminate the measurements, so we observe

$$Y_n = \sum_k N_{nk} + R_n,$$

where $\{R_n\}$ are independent Poisson variates:

$$R_n \sim \text{Poisson}\{r_n\},$$

and $\{r_n\}$ are assumed known for simplicity. Thus,

$$Y_n \sim \text{Poisson}\{\sum_k a_{nk}\lambda_k + r_n\}. \quad (7)$$

Given realizations $\{y_n\}$ of $\{Y_n\}$, the log-likelihood for this problem is given by [1]:

$$\log f(\boldsymbol{y}; \boldsymbol{\lambda}) = \sum_k \left(-\hat{y}_n(\boldsymbol{\lambda}) + y_n \log \hat{y}_n(\boldsymbol{\lambda})\right),$$

where

$$\hat{y}_n(\boldsymbol{\lambda}) = \sum_k a_{nk}\lambda_k + r_n.$$

We would like to compute the ML estimate of $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_p]'$ from $\boldsymbol{y}$.

To apply coordinate ascent directly to this likelihood, one might try to update λ_k by equating the derivative of the likelihood to zero:

$$0 = -a_k + \sum_n a_{nk} \frac{y_n}{a_{nk}\lambda_k + \hat{y}_n^{(-k)}(\boldsymbol{\lambda}^i)}, \quad (8)$$

where $a_k = \sum_n a_{nk}$ and $\hat{y}_n^{(-k)}(\boldsymbol{\lambda}) = \sum_{j \neq k} a_{nj}\lambda_j + r_n$. Unfortunately, there is no analytical solution to this equation, hence the popularity of EM-type algorithms [1].

B. ML-EM Algorithms

The complete-data space for the classical EM algorithm [1] for this problem is the set of unobservable random variates

$$\boldsymbol{X}^1 = \{\{N_{nk}\}_{k=1}^p, \{R_n\}\}_{n=1}^N. \quad (9)$$

For this complete-data space, the Q function (3) becomes [1, eqn. (4)]:

$$Q^1(\boldsymbol{\lambda}; \bar{\boldsymbol{\lambda}}) = \sum_n \sum_k \left(-a_{nk}\lambda_k + \bar{N}_{nk}\log(a_{nk}\lambda_k)\right),$$

where [1]

$$\bar{N}_{nk} = E\{N_{nk} | \boldsymbol{Y} = \boldsymbol{y}; \bar{\boldsymbol{\lambda}}\} = \bar{\lambda}_k a_{nk} y_n / \hat{y}_n(\bar{\boldsymbol{\lambda}}).$$

Maximizing $Q^1(\cdot; \boldsymbol{\lambda}^i)$ analytically leads to the following algorithm:

ML-EM-1 Algorithm

`for` $i = 0, 1, \ldots$ {

$$\hat{y}_n := \sum_k a_{nk} \lambda_k^i + r_n, \; n = 1, \ldots, N$$

`for` $k = 1, \ldots, p$ {

$$e_k = \sum_n a_{nk} y_n / \hat{y}_n$$

$$\lambda_k^{i+1} = \lambda_k^i e_k / a_k \tag{10}$$

}

}.

In words, the current parameter estimate is used to compute predicted measurements, those predictions are divided into the measurements and backprojected to form multiplicative correction factors, and the estimates are *simultaneously* updated using those correction factors[2]. This EM algorithm converges globally [1, 4] but slowly. The root-convergence factor is very close to 1 (even if $p = 1$ [4]).

The slow convergence is largely explained by considering the Fisher information of the complete-data space $\boldsymbol{X}^1$ [4]. One can think of $\boldsymbol{X}^1$ as data from a hypothetical tomograph that knows whether each detected event is a true emission or a background event, and knows in which pixel each event originated. Such a tomograph would clearly be much more *informative* than the conventional tomographs, and this intuition is reflected in the Fisher information matrices. The Fisher information for the observed data $\boldsymbol{Y}$ at the ML estimate $\hat{\boldsymbol{\lambda}}$ is

$$\mathbf{F}_{\mathbf{Y}}(\hat{\boldsymbol{\lambda}}) = \mathbf{A}' \operatorname{diag}\{\mathbf{A}\hat{\boldsymbol{\lambda}} + \boldsymbol{r}\}^{-1} \mathbf{A},$$

whereas the Fisher information for $\boldsymbol{X}^1$ is diagonal:

$$\mathbf{F}_{\mathbf{X}^1}(\hat{\boldsymbol{\lambda}}) = \operatorname{diag}\{a_k / \hat{\lambda}_k\}$$

(provided $\hat{\boldsymbol{\lambda}}$ is positive.) One can show that $\mathbf{F}_{\mathbf{X}^1} > \mathbf{F}_{\mathbf{Y}}$. Indeed, $\mathbf{F}_{\mathbf{X}^1}$ is completely independent of the background rates $\{r_n\}$, reflecting the fact that the parameters are completely isolated from the uncertainty due to the background events $\{R_n\}$ in $\boldsymbol{X}^1$.

The first remedy one might try is to define a new complete-data space

$$\boldsymbol{X}^2 = \{\{X_{nk}\}_{k=1}^p\}_{n=1}^N,$$

where the $\{X_{nk}\}$ are unobservable independent variates:

$$X_{nk} \sim \text{Poisson}\{a_{nk}\lambda_k + r_n / p_n\}, \tag{11}$$

where $p_n = \#\{a_{nk} \neq 0\}$. Then clearly $Y_n = \sum_k X_{nk}$ has the appropriate distribution (7). However, when the Q function is formed, one finds that it has no analytical maximum, so one is no better off than with (8).

Therefore, we propose to use a complete-data space with the following form:

$$\boldsymbol{X}^3 = \{\{M_{nk}\}_{k=1}^p, \{B_n\}\}_{n=1}^N,$$

where $\{M_{nk}\}$ and $\{B_n\}$ are unobservable independent Poisson variates:

$$\begin{aligned} M_{nk} &\sim \text{Poisson}\{a_{nk}(\lambda_k + m_k)\} \\ B_n &\sim \text{Poisson}\{r_n - \sum_k a_{nk} m_k\}, \end{aligned} \tag{12}$$

where $\{m_k\}$ are design parameters that must satisfy

$$r_n \geq \sum_k a_{nk} m_k, \; \forall n, \tag{13}$$

so that the Poisson rates of $\{B_n\}$ are nonnegative.

With these definitions, clearly

$$Y_n = \sum_k M_{nk} + B_n$$

has the appropriate distribution (7). One choice for $\{m_k\}$ that satisfies (13) is:

$$m_k = \min_{n : a_{nk} \neq 0} \{r_n / (p_n a_{nk})\}, \tag{14}$$

which absorbs many, although not all, of the background events into the terms M_{nk}. Therefore, the parameters are less isolated from the background events. The Fisher information for $\boldsymbol{X}^3$ is diagonal:

$$\mathbf{F}_{\mathbf{X}^3}(\hat{\boldsymbol{\lambda}}) = \operatorname{diag}\{a_k / (\hat{\lambda}_k + m_k)\}$$

which now depends on r_n though (14). This Fisher information is smaller than $\mathbf{F}_{\mathbf{X}^1}(\hat{\boldsymbol{\lambda}})$, which leads to faster convergence.

Whereas the classical complete-data space $\boldsymbol{X}^1$ has some intuitive relationship with the underlying image formation physics, we developed the new complete-data space $\boldsymbol{X}^3$ using a statistical perspective on the problem and its Fisher information.

Using a similar derivation as in [1] one can show:

$$Q^3(\boldsymbol{\lambda}; \bar{\boldsymbol{\lambda}}) =$$

$$\sum_n \sum_k (-a_{nk}(\lambda_k + m_k) + \bar{M}_{nk} \log(a_{nk}(\lambda_k + m_k)), \tag{15}$$

where

$$\bar{M}_{nk} = E\{M_{nk} | \boldsymbol{Y} = \boldsymbol{y}; \bar{\boldsymbol{\lambda}}\} = (\bar{\lambda}_k + m_k) a_{nk} y_n / \hat{y}_n(\bar{\boldsymbol{\lambda}}).$$

Maximizing $Q^3(\cdot; \boldsymbol{\lambda}^i)$ analytically (subject to the nonnegativity constraint), yields the ML-EM-2 algorithm, which has the same form as ML-EM-1, except that (10) is replaced by:

$$\lambda_k^{i+1} = [(\lambda_k^i + m_k) e_k / a_k - m_k]_+,$$

[2] ML-EM-1 is essentially the ML-IB algorithm of [5]. Also proposed in [5] was the ML-IA algorithm, which has a more informative complete-data space and slower convergence [4].

where

$$[x]_+ = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} .$$

This is a trivial change to the implementation of ML-EM-1, but it does lead to improved convergence rates, both theoretically and empirically, provided of course that some $m_k \neq 0$. In PET, since since random coincidences are pervasive, we will have $r_n > 0\ \forall n$, so that $m_k > 0\ \forall k$.

Like ML-EM-1, since ML-EM-2 is an EM algorithm it monotonically increases the likelihood every iteration [2]. An interesting difference between the iterates generated by ML-EM-1 and ML-EM-2 is that the latter can move on and off the boundary of the nonnegative orthant from iteration to iteration. From a numerical perspective, this may partly explain the faster convergence of ML-EM-2, since when ML-EM-1 converges to the boundary, it can do so at *sublinear* rates [4].

C. ML-LINU Algorithms

ML-EM-1 is the special case where $\alpha = 1$ of the form [6]:

$$\lambda_k^{i+1} = \left[\lambda_k^i + \alpha \left(\frac{\lambda_k}{a_k}\right) \frac{\partial}{\partial \lambda_k} \log f(\boldsymbol{y}; \boldsymbol{\lambda}^i)\right]_+ .$$

The ML-LINU algorithm [6] uses a line-search to choose an $\alpha^i > 1$, which accelerates the convergence of ML-EM-1. Similarly, ML-EM-2 is the special case where $\alpha = 1$ of:

$$\lambda_k^{i+1} = \left[\lambda_k^i + \alpha \left(\frac{\lambda_k + m_k}{a_k}\right) \frac{\partial}{\partial \lambda_k} \log f(\boldsymbol{y}; \boldsymbol{\lambda}^i)\right]_+ .$$

In the few experiments tried, we found that "accelerating" ML-EM-2 by choosing $\alpha > 1$ did not improve convergence much, primarily because of the nonnegativity constraint. (When $\alpha > 1$ causes a "bent line" search, then evaluating the likelihood difference requires an expensive forward projection [6].)

D. ML-SAGE Algorithms

Motivations for the SAGE algorithms include:

- As shown by Sauer and Bouman [7], sequential update methods often converge much faster than simultaneous update methods.
- Sequential update methods solve the coupling problem introduced by smoothness penalties.
- By using an alternating sequence of hidden-data spaces, we can associate a large fraction of the background events with each parameter as it is updated, yielding much less informative hidden-data spaces and thus faster convergence. In contrast, in ML-EM-2 the background events are distributed among all of the pixels; since p_n is typically about 100, the values for m_k are small.

All of the SAGE algorithms we derive in this paper use individual pixels for the index sets: $S^i = \{k\}$, where $k = 1 + (i \text{ modulo } p)$.

The most obvious hidden-data for λ_k is just

$$\boldsymbol{X}_k^4 = \{N_{nk}, R_n\}_{n=1}^N,$$

which is a subset of the classical complete-data space (9). The Q function for the kth parameter is:

$$Q_k^4(\lambda_k; \bar{\boldsymbol{\lambda}}) = \sum_n \left(-a_{nk}\lambda_k + \bar{N}_{nk} \log(a_{nk}\lambda_k)\right) .$$

Maximizing $Q_k^4(\cdot; \boldsymbol{\lambda}^i)$ analytically yields the following algorithm:

ML-SAGE-1 Algorithm

`Initialize:` $\hat{y}_n = \sum_k a_{nk}\lambda_k^0 + r_n,\ n = 1, \ldots, N.$
`for` $i = 0, 1, \ldots$ {
  `for` $k = 1, \ldots, p$ {

$$\begin{aligned} e_k &= \sum_n a_{nk} y_n / \hat{y}_n \\ \lambda_k^{i+1} &= \lambda_k^i e_k / a_k \\ \lambda_j^{i+1} &= \lambda_j^i,\ j \neq k \\ \hat{y}_n &:= \hat{y}_n + (\lambda_k^{i+1} - \lambda_k^i) a_{nk},\ \forall n : a_{nk} \neq 0 \end{aligned} \tag{16}$$

  }
}.

This SAGE algorithm updates the parameters *sequentially*, and immediately updates the predicted measurements $\hat{y}_n$ within the inner loop, whereas the ML-EM algorithms wait until all parameters have been updated.

We found that ML-SAGE-1 converges somewhat faster than ML-EM-1 for well conditioned problems, but the difference is smaller for poorly conditioned problems. The reason is that $\boldsymbol{X}_k^4$ is still overly informative since the background events are isolated from the parameter being updated. We can improve on ML-SAGE-1 by using hidden-data spaces with similar forms as (12), only even less informative. Since we are updating one pixel at a time, we can associate nearly all of the background events with each pixel as it is updated. This is nonsense to a physicist, but is completely admissible from a statistical perspective. First, define

$$z_k = \min_{n: a_{nk} \neq 0} \{r_n / a_{nk}\},$$

and define unobservable independent Poisson variates:

$$\begin{aligned} Z_{nk} &\sim \text{Poisson}\{a_{nk}(\lambda_k + z_k)\} \\ B_{nk} &\sim \text{Poisson}\{r_n - a_{nk} z_k + \sum_{j \neq k} a_{nj}\lambda_j\}, \end{aligned} \tag{17}$$

and let the hidden-data space for λ_k *only* be

$$\boldsymbol{X}_k^5 = \{Z_{nk}, B_{nk}\}_{n=1}^N .$$

Then clearly

$$Y_n = Z_{nk} + B_{nk}$$

has the appropriate distribution (7) for any particular k. Since the definition of z_k does not include the p_n term contained in the definition (14) of m_k, the values of z_k are orders of magnitude larger than m_k. Thus a very large fraction of the background events is absorbed into the term Z_{nk} which is associated with λ_k while it is updated[3]. The Fisher information for $\boldsymbol{X}_k^5$ is

$$\mathbf{F}_{\mathbf{X}_k^5}(\hat{\boldsymbol{\lambda}}) = a_k/(\hat{\lambda}_k + z_k),$$

which is much smaller than the kth diagonal entry of $\mathbf{F}_{\mathbf{X}^3}$ since $z_k >> m_k$.

Using a similar derivation as in [1] one can show:

$$Q_k^5(\lambda_k; \bar{\boldsymbol{\lambda}}) = \sum_n (-a_{nk}(\lambda_k + z_k) + \bar{Z}_{nk} \log(a_{nk}(\lambda_k + z_k)),$$

where

$$\bar{Z}_{nk} = E\{Z_{nk} | \boldsymbol{Y} = \boldsymbol{y}; \bar{\boldsymbol{\lambda}}\} = (\bar{\lambda}_k + z_k) a_{nk} y_n / \hat{y}_n(\bar{\boldsymbol{\lambda}}).$$

Maximizing $Q_k^5(\cdot; \boldsymbol{\lambda}^i)$ analytically (subject to the nonnegativity constraint), yields the ML-SAGE-2 algorithm, which has the same form as ML-SAGE-1, except that (16) is replaced by:

$$\lambda_k^{i+1} = [(\lambda_k^i + z_k) e_k / a_k - z_k]_+.$$

This is a very small change to ML-SAGE-1, but one that significantly accelerates convergence. Indeed, the implementation differences between ML-EM-1, ML-EM-2, ML-SAGE-1, and ML-SAGE-2 are all remarkably minor, but the differences in convergence rates are not, as illustrated by the results in [3], one of which is shown in Fig. 1.

An alternative to SAGE is the coordinate-wise sequential Newton-Raphson updates recently proposed by Bouman and Sauer [8]. That method is not guaranteed to be monotonic, and it is somewhat more expensive per iteration since second derivatives must be computed. But when it converges, its asymptotic convergence rate may be somewhat faster than SAGE since it is even greedier [2]. Similar (but monotonic) greediness can be obtained by using multiple sub-iterations of the E- and M-steps in the SAGE algorithm. However, for the few cases we have tested, we have not observed any improvement in convergence rates using multiple sub-iterations. Although further investigation of the tradeoffs available is needed, including comparisons with possibly super-linear methods such as preconditioned conjugate gradient [6], it appears that the statistical perspective required by the SAGE method is a useful addition to conventional numerical tools.

Sequential methods like SAGE are less amenable to fine-grain parallelization than simultaneous methods like EM or conjugate gradient. By fine-grain we mean each processing unit corresponds to one pixel. However, emission tomography is ideally suited to coarse-grain parallelization since each tomographic slice can be assigned to a processing unit. For slice-by-slice reconstruction, coarse-grain parallelization can achieve 100% processor utilization.

[3] The aggregate of all p of the hidden-data spaces X_k^5 is *not* an admissible hidden-data space for the entire parameter vector λ, so one cannot use (17) in a simultaneous update EM algorithm.

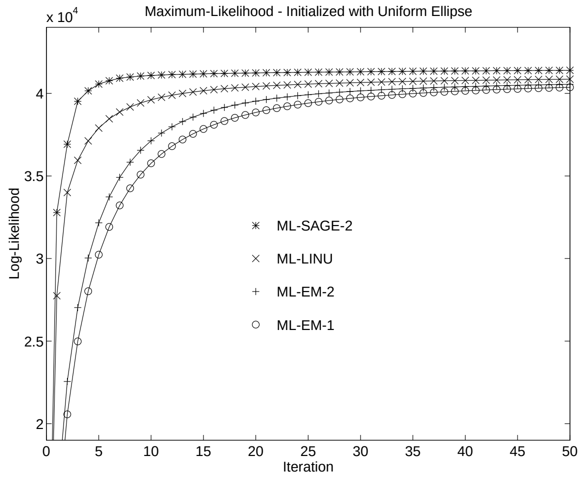

Figure 1: Comparison of log-likelihood increases $\log f(\boldsymbol{y}; \boldsymbol{\theta}^i) - \log f(\boldsymbol{y}; \boldsymbol{\theta}^0)$ vs iteration i of ML-EM-1, ML-EM-2, ML-LINU, and ML-SAGE-2 algorithms, for image reconstruction from PET measurements. ML-SAGE-2 clearly reaches the asymptote fastest.

References

[1] K. Lange and R. Carson. EM reconstruction algorithms for emission and transmission tomography. *Journal of Computer Assisted Tomography*, 8(2):306–316, April 1984.

[2] J. A. Fessler and A. O. Hero. Space-alternating generalized EM algorithm, 1993. Submitted to IEEE Trans. Signal Proc.

[3] J. A. Fessler. Penalized-likelihood image reconstruction using space-alternating generalized EM algorithms, 1993. In preparation.

[4] J. A. Fessler, N. H. Clinthorne, and W. Leslie Rogers. On complete data spaces for PET reconstruction algorithms. *IEEE Transactions on Nuclear Science*, 40(4):1055–1061, August 1993.

[5] D. G. Politte and D. L. Snyder. Corrections for accidental coincidences and attenuation in maximum-likelihood image reconstruction for positron-emission tomography. *IEEE Transactions on Medical Imaging*, 10(1):82–89, March 1991.

[6] L. Kaufman. Implementing and accelerating the EM algorithm for positron emission tomography. *IEEE Transactions on Medical Imaging*, 6(1):37–51, March 1987.

[7] K. Sauer and C. Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Transactions on Signal Processing*, 41(2):534–548, February 1993.

[8] C. Bouman and K. Sauer. Fast numerical methods for emission and transmission tomographic reconstruction. In *Proc. Conf. Info. Sci. Sys., Johns Hopkins*, 1993.