SPACE-ALTERNATING GENERALIZED EM ALGORITHMS FOR PENALIZED MAXIMUM-LIKELIHOOD IMAGE RECONSTRUCTION

Jeffrey A. Fessler and Alfred O. Hero

COMMUNICATIONS & SIGNAL PROCESSING LABORATORY
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, MI 48109-2122
and
Division of Nuclear Medicine
Department of Internal Medicine
The University of Michigan
Ann Arbor, MI 48109-2122

February 1994

Technical Report No. 286
Approved for public release; distribution unlimited

Space-Alternating Generalized EM Algorithms for Penalized Maximum-Likelihood Image Reconstruction[1]

Jeffrey A. Fessler and Alfred O. Hero
Division of Nuclear Medicine and Department of Electrical Engineering and Computer Science
3480 Kresge III, Box 0552
University of Michigan
Ann Arbor, MI 48109-0552
email: fessler@umich.edu
(313) 763-1434

Technical Report 286

ABSTRACT

Most expectation-maximization (EM) type algorithms for penalized maximum-likelihood image reconstruction converge particularly slowly when one incorporates additive background effects such as scatter, random coincidences, dark current, or cosmic radiation. In addition, regularizing smoothness penalties (or priors) introduce parameter coupling, rendering intractable the M-steps of most EM-type algorithms. This report presents space-alternating generalized EM (SAGE) algorithms for image reconstruction, which update the parameters sequentially *using a sequence of small "hidden" data spaces, rather than* simultaneously *using one large complete-data space. The sequential update decouples the M-step, so the maximization can typically be performed analytically. We introduce new hidden-data spaces that are less informative than the conventional complete-data space for Poisson data and that yield significant improvements in convergence rate. This acceleration is due to statistical considerations, not numerical overrelaxation methods, so monotonic increases in the objective function are guaranteed. We provide a general global convergence proof for SAGE methods with nonnegativity constraints.*

[1] 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.

I. INTRODUCTION

Imaging techniques with Poisson measurement statistics include: positron emission tomography (PET) [1], single photon emission computed tomography (SPECT), gamma astronomy, various microscopy methods [2, 3], and photon-limited optical imaging [4]. Statistical methods for image reconstruction or restoration, such as maximum likelihood (ML), penalized maximum-likelihood (PML), or maximum *a posteriori* (MAP), are computationally challenging due to the transcendental form of the Poisson log-likelihood. The difficulty is exacerbated when one includes smoothness penalties or priors, since these functionals further couple the parameters. EM algorithms [5] have proven to be somewhat useful in such problems, except for two important drawbacks. The first problem is convergence rate: EM algorithms converge slowly, particularly when one includes the additive effects of "background" events such as random coincidences [6], scatter [7, 8], dark-current [9], or background cosmic radiation. The second problem is that the M-step of the EM algorithm becomes intractable when one includes smoothness penalties in the objective function. Since image reconstruction is ill-posed, such penalties are often very desirable.

Unlike the statistical applications that motivated the development of the EM algorithm [5], there is usually no "missing data" in image reconstruction problems. Here the EM algorithm serves primarily as a computational tool that replaces one difficult maximization problem with a recursion of easier maximizations. It is no ordinary numerical tool however, since each EM algorithm exploits the underlying statistical structure of the log-likelihood. In that same spirit, this report proposes a new tool for image reconstruction from Poisson measurements: the space-alternating generalized EM (SAGE) method. This method allows one to further exploit the structure of the log-likelihood.

In contrast to the *simultaneous* update used in nearly all EM-type algorithms, the SAGE algorithms presented in this report use *sequential* parameter updates, where each pixel can be updated individually. A sequential update eliminates the coupling problem introduced by smoothness penalties. Sauer and Bouman [10] have explained the rapid convergence of certain sequential updates using a novel frequency-domain analysis.

Although Dempster *et al.* [5] showed that the convergence rates of EM algorithms are related to the Fisher information matrices of their complete-data spaces, this property does not appear to have been widely appreciated or exploited. We have previously shown that reducing Fisher information can lead to remarkable improvements in convergence rates [11–16]. The relationship between Fisher information and convergence rate underscores all of the methods presented in this report. In particular, the sequential update of our SAGE algorithms allows us to use small hidden-data spaces that are considerably less informative than the ordinary complete-data space for image reconstruction, which leads to fast monotonic convergence.

Images reconstructed purely by using the maximum likelihood criterion [1, 17] have been found to be unacceptably noisy. A variety of methods have been proposed to reduce this noise, usually with some concordant resolution tradeoff. These methods include: aborting the iteration before convergence [18], using quadratic approximations to the likelihood with a penalty [19, 20], using a separable (non-smoothness) prior [21–23], and introducing a smoothing step into the ML-EM iteration [24–26]. Perhaps the most popular alternative is the method of sieves [27, 28]. Sieves are usually implemented by post-smoothing, even though the commutability requirement [28, eqn. (12)] is rarely met in practice. However, recent studies, e.g. [29], have found that MAP (or equivalently PML) methods outperform the method of sieves. Therefore, in this report, we focus on penalized maximum-likelihood image reconstruction, where one modifies the objective function to include a roughness penalty. This approach also has the flexibility to include spatially-variant penalties that reflect prior anatomical boundary information [30]. The new complete-data and hidden-data spaces we introduce are applicable to both penalized and unpenalized maximum-likelihood methods.

Penalized likelihood objective functions for Poisson statistics are difficult to maximize (in comparison with Gaussian problems), and dozens of algorithms have been proposed. Such algorithms can be categorized as: 1) *intrinsically monotonic* methods, 2) *forced monotonic* methods (typically made monotonic using a line search), and 3) *nonmonotonic methods*. Since one could convert any nonmonotonic method to a forced monotonic method by using a line search, the latter two categories overlap.

Intrinsically monotonic methods are those such as the ML-EM algorithm for PET where the form of the recursion inherently ensures that the objective function increases every iteration (ignoring finite precision computing). The only intrinsically monotonic methods for penalized maximum-likelihood that we are aware of are: 1) extensions of the EM algorithm including generalized expectation-maximization (GEM) algorithms [31–34] and expectation/conditional maximization (ECM) algorithms

[35, 36], 2) algorithms for the trivial case with separable (non-smoothness) priors [21–23], 3) the algorithms of De Pierro [37–39], 4) the "ICM-EM" algorithm of Abdalla and Kay [40], and 5) the new algorithms in this report. For the purposes of comparison, we derive new faster versions of the GEM and De Pierro algorithms in Section III using a new complete-data space. The "ICM-EM" algorithm of [40] is a special case of our SAGE method, but one that converges slower than our recommended method (Section IV). Most intrinsically monotonic algorithms have been shown to converge globally to the unique maximum for strictly concave objectives.

Nonmonotonic methods can diverge if one does not explicitly check that the objective increases, and in applications with many parameters it is often expensive to evaluate the likelihood or to "backtrack" when the likelihood decreases. The SAGE methods we propose avoid line searches; monotonicity of the algorithms is guaranteed by the *statistical* formulation. Although it is not our purpose to argue this point, we believe that convergence properties are relevant to clinical medical imaging, since algorithm divergence could have unfortunate consequences.

Perhaps a more accurate name for nonmonotonic methods would be "not necessarily monotonic" since indeed most such methods do have *local* convergence. In particular, the penalized maximum-likelihood estimate is nearly always a fixed point of such methods. An early approach was gradient ascent of the objective starting from an ML estimate [41, 42], which was stated to "not guarantee convergence to the global [max]imum." Gradient ascent is complicated by the nonnegativity constraint. Most other nonmonotonic methods are variations of the one-step late (OSL) method of Green [43, 44], first mentioned in [45]. In the OSL approach, one circumvents the problem of coupled equations by "plugging in" values from the previous iteration. Unfortunately, such an approach can diverge, unless modified to include a line search [46]. Similar strategies include the BIP algorithm [47, 48], the methods in [49, 50], and nested gradient or Jacobi iterations [29, 51, 52]. Most such strategies include a user-specified step size parameter, and one user has noted that "finding good values for [the step size] and the number of times to iterate requires painful experimentation [53]." Other OSL-like methods are given in [53, 54], which have been reported to occasionally diverge [54]. The sequential update of our SAGE methods is close in form (cf Type-III algorithms in Table 1) to the coordinate-wise Newton-Raphson ascent of the objective function proposed by Bouman and Sauer [55, 56]. As described in Section IV, that method is also not necessarily monotonic, although is has appeared to converge for the examples we have tried.

Any of the above methods could be forced to be monotonic by adding a line-search step. Lange has shown convergence for a line-search modification of OSL [46], and Mucuoglu *et al.* have adapted the conjugate gradient method [57]. We show below that our intrinsically monotonic ML-SAGE algorithm converges faster than even the line-search accelerated ML-EM algorithm of Kaufman [58].

The organization of this report is as follows. Section II describes the general structure of the SAGE method. Section III introduces new complete-data spaces and hidden-data spaces for Poisson data, and gives several algorithms for unpenalized maximum-likelihood. Section IV presents new algorithms for the penalized maximum-likelihood objective. These algorithms, along with the proof of global convergence in Appendix I, are the main contributions of this report. Sections V and VI illustrate the convergence rates of the various algorithms.

II. The SAGE Method

In previous work [13–15] we described the SAGE method within a statistical framework. In this section, we first describe a generalized version of the method without direct statistical considerations, and then introduce the statistical version as a special case. The non-statistical perspective is extended from the work of De Pierro [39, 59], and contains the algorithms of [13–15] and [38,39] as special cases.

A. Problem

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

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

where

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

and P is an optional penalty function. Analytical solutions for $\hat{\boldsymbol{\theta}}$ are often unavailable due to the complexity of f, the coupling in P, or both. Thus one must resort to iterative methods.

Most iterative image reconstruction methods update all pixels *simultaneously* each iteration. Recently however, the advantages of *sequential* pixel updates have been noted by Sauer and Bouman [10], including: fast convergence, natural enforcement of nonnegativity, and decoupled penalty functions. Unfortunately, in applications with Poisson statistics, there is no analytical form for maximizing the likelihood with respect to a single parameter while holding the other parameters fixed (see the next section). Thus, to implement a conventional coordinate-ascent sequential update [60], one must use one-dimensional line searches or Newton-Raphson updates. To ensure monotonicity, those approaches may require several evaluations of the objective, and are thus more expensive than the intrinsically monotonic methods we propose below.

To describe the SAGE method, we need to first establish some notation. As in [15], we define an *index set* to be a nonempty subset of $\{1, \ldots, p\}$. If S is an index set, then $\tilde{S}$ denotes the set complement of S intersected with $\{1, \ldots, p\}$. If the cardinality of S is m, then $\boldsymbol{\theta}_S$ denotes the m dimensional vector consisting of the m elements of $\boldsymbol{\theta}$ indexed by the members of S. Similarly $\boldsymbol{\theta}_{\tilde{S}}$ denotes the $p - m$ dimensional vector consisting of the remaining elements of $\boldsymbol{\theta}$. For example, if $p = 5$ and $S = \{1, 3, 4\}$, then $\tilde{S} = \{2, 5\}$, $\boldsymbol{\theta}_S = [\theta_1 \ \theta_3 \ \theta_4]'$, and $\boldsymbol{\theta}_{\tilde{S}} = [\theta_2 \ \theta_5]'$, where $'$ denotes vector transpose. Finally, functions like $\Phi(\boldsymbol{\theta})$ expect a p-dimensional vector argument, but it is often convenient to split the argument $\boldsymbol{\theta}$ into two vectors: $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_{\tilde{S}}$, as defined above. Therefore, we define expressions such as the following to be equivalent: $\Phi(\boldsymbol{\theta}_S, \boldsymbol{\theta}_{\tilde{S}}) = \Phi(\boldsymbol{\theta})$.

B. Algorithm

The algorithm below is a generalization of the method in [15]. Let $\boldsymbol{\theta}^0 \in \Theta$ be an initial parameter estimate. A SAGE algorithm produces a sequence of estimates $\{\boldsymbol{\theta}^i\}_{i=0}^{\infty}$ via the following recursion:

Generalized SAGE Algorithm

```
For $i = 0, 1, \ldots$ {
    1. Choose an index set $S^i$.
    2. E-step: Choose a functional $\phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)$ satisfying:
       $\Phi(\boldsymbol{\theta}_{S^i}, \boldsymbol{\theta}^i_{\tilde{S}^i}) - \Phi(\boldsymbol{\theta}^i) \geq \phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i) - \phi^i(\boldsymbol{\theta}^i_{S^i}; \boldsymbol{\theta}^i).$   (2)
    3. M-step:
       $\boldsymbol{\theta}^{i+1}_{S^i} = \arg\max_{\boldsymbol{\theta}_{S^i}} \phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)$   (3)
       $\boldsymbol{\theta}^{i+1}_{\tilde{S}^i} = \boldsymbol{\theta}^i_{\tilde{S}^i}$   (4)
}.
```

The maximization in (3) and the inequality in (2) are over the set

$$\{\boldsymbol{\theta}_{S^i} : (\boldsymbol{\theta}_{S^i}, \boldsymbol{\theta}^i_{\tilde{S}^i}) \in \Theta\}.$$

This is an "algorithm" in a loose sense, since there is considerable latitude for the algorithm designer when choosing the index sets $\{S^i\}$ and functionals $\{\phi^i\}$. The basic idea behind the SAGE method is borrowed directly from the EM method, but adapted to a sequential update. Rather than trying to maximize $\Phi(\boldsymbol{\theta}_{S^i}, \boldsymbol{\theta}^i_{\tilde{S}^i})$ over $\boldsymbol{\theta}_{S^i}$ at the ith iteration, we maximize instead some user-specified functional $\phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)$. That functional is carefully chosen to ensure (using (2)) that increases in ϕ^i yield increases in Φ. If ϕ^i and S^i are chosen wisely, then one can maximize $\phi^i(\cdot; \boldsymbol{\theta}^i)$ *analytically,* yielding a recursion of the form $\boldsymbol{\theta}^{i+1}_{S^i} = g^i(\boldsymbol{\theta}^i)$, which obviates the need for line searches. The image reconstruction algorithms given in the next sections illustrate this important aspect. Even if one cannot maximize ϕ^i analytically, one can often choose ϕ^i such that line searches for maximizing $\phi^i(\cdot; \boldsymbol{\theta}^i)$ are cheaper than line searches for maximizing $\Phi(\cdot, \boldsymbol{\theta}^i_{\tilde{S}^i})$. In some cases, maximizing $\phi^i(\cdot; \boldsymbol{\theta}^i)$ will increase $\Phi(\cdot, \boldsymbol{\theta}^i_{\tilde{S}^i})$ almost as much as maximizing $\Phi(\cdot, \boldsymbol{\theta}^i_{\tilde{S}^i})$ itself.

Rather than requiring a strict maximization in (3), one could settle simply for local maxima [16], or for mere increases in ϕ^i, in analogy with GEM algorithms [5]. These generalizations provide the opportunity to further refine the tradeoff between convergence rate and computation per-iteration.

C. Convergence Properties

It follows from (2) and (3) that the sequence of estimates $\{\boldsymbol{\theta}^i\}$ generated by any SAGE algorithm will monotonically increase the objective $\Phi(\boldsymbol{\theta}^i)$. If the objective function is bounded above, then this monotonicity ensures that $\{\Phi(\boldsymbol{\theta}^i)\}$ converges, but it does not guarantee convergence of the sequence $\{\boldsymbol{\theta}^i\}$. In [15], we provided regularity conditions under which the sequence $\{\boldsymbol{\theta}^i\}$ also converges monotonically *in norm,* and derived an expression for the asymptotic rate of convergence. The nonnegativity constraint for image reconstruction violates one of those regularity conditions. Therefore, in Appendix I we prove global convergence under mild conditions suitable for image reconstruction with nonnegativity constraints.

D. Hidden-Data Spaces

A natural approach to choosing functionals ϕ^i that satisfy (2) is to use the underlying statistical structure of the problem. In many problems, one can simplify the form of the log-likelihood by augmenting the observed data with some additional unobservable or "hidden" data. The following definition formalizes this concept.

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

$\boldsymbol{Y}$ satisfies

$$f(\boldsymbol{y}, \boldsymbol{x}; \boldsymbol{\theta}) = f(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta}_{\tilde{S}}) f(\boldsymbol{x}; \boldsymbol{\theta}), \tag{5}$$

i.e., the conditional distribution $f(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta}_{\tilde{S}})$ must be independent of $\boldsymbol{\theta}_S$.
Any complete-data space associated with a conventional EM algorithm is a special case of this definition [15].

Given an admissible hidden-data space $\boldsymbol{X}$, define the following conditional expectation of its log-likelihood:

$$\begin{aligned} Q(\boldsymbol{\theta}_S; \bar{\boldsymbol{\theta}}) &= Q(\boldsymbol{\theta}_S; \bar{\boldsymbol{\theta}}_S, \bar{\boldsymbol{\theta}}_{\tilde{S}}) \\ &= E\left\{\log f(\boldsymbol{X}; \boldsymbol{\theta}_S, \bar{\boldsymbol{\theta}}_{\tilde{S}}) | \boldsymbol{Y} = \boldsymbol{y}; \bar{\boldsymbol{\theta}}\right\} \qquad (6) \\ &= \int f(\boldsymbol{x}|\boldsymbol{Y} = \boldsymbol{y}; \bar{\boldsymbol{\theta}}) \log f(\boldsymbol{x}; \boldsymbol{\theta}_S, \bar{\boldsymbol{\theta}}_{\tilde{S}}) \, d\boldsymbol{x}. \end{aligned}$$

Combine this expectation with the penalty function:

$$\phi(\boldsymbol{\theta}_S; \bar{\boldsymbol{\theta}}) \stackrel{\triangle}{=} Q(\boldsymbol{\theta}_S; \bar{\boldsymbol{\theta}}) - P(\boldsymbol{\theta}_S, \bar{\boldsymbol{\theta}}_{\tilde{S}}). \tag{7}$$

It then follows from [15] that a functional ϕ generated using (5)-(7) satisfies (2). Such a functional also satisfies:

$$\nabla_{\boldsymbol{\theta}_S} \Phi(\boldsymbol{\theta}) = \nabla^{10}_{\boldsymbol{\theta}_S} \phi(\boldsymbol{\theta}_S; \boldsymbol{\theta}), \tag{8}$$

which is used in our proof of global convergence. Thus, one can easily design a SAGE algorithm by first choosing index sets $\{S^i\}$, choosing admissible hidden-data spaces $\{\boldsymbol{X}^i\}$, and then generating $\{\phi^i\}$ functionals using (5)-(7). The "majorization" method of De Pierro [39, 59] is an alternative method for choosing ϕ^i functionals; see Section IV.C.

E. Choosing Index Sets

In general, there are a wide variety of possible choices for the index sets S^i, as discussed in [15]. In this report we focus on single-pixel index sets, e.g.:

$$S^i = \{1 + (i \operatorname{modulo} p)\}. \tag{9}$$

In practice, rather than always using the same order of updates, we alternate between four natural raster scan orders (top-down, left-right, etc.).

III. Maximum Likelihood

In this section we first review the linear Poisson model that is often used in image reconstruction problems, and summarize the classical EM algorithm (ML-EM-1) for maximizing the likelihood [1, 17]. We then introduce a new complete-data space that leads to a new, faster converging EM algorithm: ML-EM-3. Even less informative hidden-data spaces lead to new SAGE algorithms that converge faster than both ML-EM-3 and the line-search accelerated EM algorithm (ML-LINU) [58]. We presented some of this material in [14, 15]; we include it here since the concepts behind the new complete-data spaces and hidden-data spaces are easier to explain in the maximum-likelihood framework than in the penalized maximum-likelihood case described in the next section.

The derivations sketched below all use the following property of scalar Poisson variates:

If $X_1 \sim \text{Poisson}\{\mu_1\}$ and $X_2 \sim \text{Poisson}\{\mu_2\}$ are independent and $Y = X_1 + X_2$, then [17]

$$E\{X_1|Y = y; \mu_1, \mu_2\} = \mu_1 \frac{y}{\mu_1 + \mu_2}. \tag{10}$$

A. The Problem

Assume that the emission distribution can be discretized into p pixels with emission rates $\boldsymbol{\lambda} = [\lambda_0, \ldots, \lambda_p]'$. Assume that the emission source is viewed by N detectors, and let N_{nk} denote the number of emissions from the kth pixel 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 [17]. 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:

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

Thus, our measurement model is

$$Y_n \sim \text{Poisson}\{\sum_k a_{nk}\lambda_k + r_n\}. \tag{11}$$

In this report, we assume the background rates $\{r_n\}$ are known. This assumption is not essential to the general method, and one could generalize the approach to accommodate joint estimation [12] of $\{\lambda_k\}$ and $\{r_n\}$. We assume the column sums $a_{\cdot k} = \sum_n a_{nk}$ are nonzero.

Given realizations $\{y_n\}$ of $\{Y_n\}$, the log-likelihood for this problem is given by [17]:

$$L(\boldsymbol{\lambda}) = \log f(\boldsymbol{y}; \boldsymbol{\lambda}) \equiv \sum_n \left(-\bar{y}_n(\boldsymbol{\lambda}) + y_n \log \bar{y}_n(\boldsymbol{\lambda})\right), \tag{12}$$

where

$$\bar{y}_n(\boldsymbol{\lambda}) = \sum_k a_{nk}\lambda_k + r_n. \qquad (13)$$

(Throughout this report, we use the symbol “$\equiv$” between expressions that are equivalent up to constant terms that are independent of $\boldsymbol{\lambda}$.) We would like to compute the ML estimate $\hat{\boldsymbol{\lambda}}$ from $\boldsymbol{y} = [y_1, \ldots, y_N]'$, where the elements of $\hat{\boldsymbol{\lambda}}$ are constrained to be nonnegative.

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

$$0 = -a_{\cdot k} + \sum_n a_{nk} \frac{y_n}{a_{nk}(\lambda_k - \lambda_k^i) + \bar{y}_n(\boldsymbol{\lambda}^i)}. \qquad (14)$$

Unfortunately, this equation has no analytical solution—hence the popularity of EM-type algorithms [17].

B. ML-EM Algorithms

che complete-data space for the classical EM algorithm [17] for this problem is the set of unobservable random variates

$$\boldsymbol{X}^1 = \{\{N_{nk}\}_{k=1}^p, \{R_n\}\}_{n=1}^N. \qquad (15)$$

The log-likelihood for this complete-data space is

$$\log f(\boldsymbol{X}^1; \boldsymbol{\lambda}) \equiv \sum_k \sum_n \left(-a_{nk}\lambda_k + N_{nk}\log(a_{nk}\lambda_k)\right).$$

Using (10) (see [17]), one finds that

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

Thus, for this complete-data space, the Q function (6) becomes [17, eqn. (4)]:

$$\begin{aligned} Q_{\mathbf{X}^1}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) &= E\{\log f(\boldsymbol{X}^1; \boldsymbol{\lambda}) | \boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\lambda}^i\} \\ &\equiv \sum_n \sum_k \left(-a_{nk}\lambda_k + \bar{N}_{nk}\log(a_{nk}\lambda_k)\right). \end{aligned}$$

By defining

$$e_k(\boldsymbol{\lambda}^i) = \sum_n a_{nk} y_n / \bar{y}_n(\boldsymbol{\lambda}^i), \qquad (16)$$

we can simplify $Q_{\mathbf{X}^1}$ to

$$Q_{\mathbf{X}^1}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) \equiv \sum_k \left(-a_{\cdot k}\lambda_k + \lambda_k^i \; e_k(\boldsymbol{\lambda}^i) \; \log \lambda_k\right). \qquad (17)$$

This Q function is a separable, concave function of $\lambda_1, \ldots, \lambda_p$. Maximizing $Q_{\mathbf{X}^1}(\cdot; \boldsymbol{\lambda}^i)$ analytically leads to the ML-EM-1 algorithm [17], which is a Type-I algorithm in Table 1 with its M-step (53) given by:

$$\lambda_k^{i+1} = \lambda_k^i e_k(\boldsymbol{\lambda}^i) / a_{\cdot k}. \qquad (18)$$

Interpreting the Type-I algorithm of Table 1 with (18) in words, ML-EM-1 works as follows: the current parameter estimate $\boldsymbol{\lambda}^i$ is used to compute predicted measurements $\{\bar{y}_n\}$, those predictions are divided into the measurements and backprojected to form multiplicative correction factors $\{e_k\}$, and the estimates are *simultaneously* updated using those correction factors[2]. This EM algorithm converges globally [12, 17] but slowly. The root-convergence factor is very close to 1 (even if $p = 1$ [12]).

The slow convergence is partly explained by considering the Fisher information of the complete-data space $\boldsymbol{X}^1$ [12]. 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 real tomographs, and this intuition is reflected in the Fisher information matrices. The Fisher information of the parameter vector $\boldsymbol{\lambda}$ for the observed data $\boldsymbol{Y}$ evaluated at the ML estimate $\hat{\boldsymbol{\lambda}}$ is

$$\begin{aligned} \boldsymbol{F}_{\mathbf{Y}}(\hat{\boldsymbol{\lambda}}) &= E\{-\nabla^2_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda})\}\Big|_{\boldsymbol{\lambda}=\hat{\boldsymbol{\lambda}}} \\ &= \boldsymbol{A}' \mathrm{diag}\left\{\boldsymbol{A}\hat{\boldsymbol{\lambda}} + \boldsymbol{r}\right\}^{-1} \boldsymbol{A}, \end{aligned}$$

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

$$\boldsymbol{F}_{\mathbf{X}^1}(\hat{\boldsymbol{\lambda}}) = \mathrm{diag}\left\{a_{\cdot k} / \hat{\lambda}_k\right\}$$

(provided $\hat{\boldsymbol{\lambda}}$ is positive.) One can show that $\boldsymbol{F}_{\mathbf{X}^1} > \boldsymbol{F}_{\mathbf{Y}}$ using a Fisher information version of the data processing inequality [61]. Indeed, $\boldsymbol{F}_{\mathbf{X}^1}$ 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 $\boldsymbol{X}^1$.

To improve the convergence rate, we would like to choose a complete-data space that is less informative than $\boldsymbol{X}^1$. To do so, we depart somewhat from the intuitive relationship between $\boldsymbol{X}^1$ and the underlying image formation physics, and instead exploit the statistical structure of (11). The first approach we tried was to define the following new complete-data space:

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

where the $\{X_{nk}\}$ are unobservable independent Poisson variates that include *all* of the background events:

$$X_{nk} \sim \mathrm{Poisson}\{a_{nk}(\lambda_k + r_n / a_{n\cdot})\}, \qquad (19)$$

[2] ML-EM-1 is essentially the ML-IB algorithm of [6]. The ML-IA algorithm of [6] has a more informative complete-data space and slower convergence [12].

where $a_{n\cdot} = \sum_k a_{nk}$. Then clearly $Y_n = \sum_k X_{nk}$ has the appropriate distribution (11). The Fisher information for $\boldsymbol{X}^2$ is diagonal:

$$\boldsymbol{F}_{\mathbf{X}^2}(\hat{\boldsymbol{\lambda}}) = \operatorname{diag}\left\{ \sum_n \frac{a_{nk}}{\hat{\lambda}_k + r_n/a_{n\cdot}} \right\},$$

which is smaller than $\boldsymbol{F}_{\mathbf{X}^1}$. However, upon forming the function $Q_{\mathbf{X}^2}$ using (6), one finds that it has no analytical maximum (unless the ratio $r_n/a_{n\cdot}$ is a constant independent of n), so one is usually little better off than with (14). This illustrates once again the tradeoff between convergence rate and computation per-iteration [12].

A comparison of $Q_{\mathbf{X}^1}$ and $Q_{\mathbf{X}^2}$ suggested that to obtain analytical maxima, we would like to replace the term $r_n/a_{n\cdot}$ in (19) with a term that is *independent* of n. 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:

$$\begin{aligned} M_{nk} &\sim \operatorname{Poisson}\{a_{nk}(\lambda_k + m_k)\} \\ B_n &\sim \operatorname{Poisson}\{r_n - \sum_k a_{nk} m_k\}, \end{aligned} \quad (20)$$

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

$$\sum_k a_{nk} m_k \leq r_n, \; \forall n, \quad (21)$$

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 (11) [3].

The Fisher information for $\boldsymbol{X}^3$ is diagonal:

$$\boldsymbol{F}_{\mathbf{X}^3}(\hat{\boldsymbol{\lambda}}) = \operatorname{diag}\left\{ a_{\cdot k}/(\hat{\lambda}_k + m_k) \right\}, \quad (22)$$

and now depends on r_n though (23) below. This Fisher information is smaller than $\boldsymbol{F}_{\mathbf{X}^1}(\hat{\boldsymbol{\lambda}})$, which leads to faster convergence. In light of (22), to make $\boldsymbol{F}_{\mathbf{X}^3}$ small we would like the design parameters $\{m_k\}$ to be "as large as possible," but still satisfying the constraint (21). In particular, we have found it natural to choose a set $\{m_k\}$ whose *smallest element is as large as possible* subject to (21). A simple solution to this min-max problem is:

$$m_k = \min_{n \,:\, a_{n\cdot} \neq 0} \left\{ \frac{r_n}{a_{n\cdot}} \right\}. \quad (23)$$

We discuss alternatives to (23) based on other min-max criteria in Appendix II, none of which we have found to perform significantly better than (23) in the few PET cases we have tried, but that might be advantageous in some situations.

The design (23) clearly satisfies (21), and at least one of the N constraints in (21) is met with equality. Thus, the M_{nk} terms absorb some of the background events, but usually not all. For tomographic systems, the $a_{n\cdot}$'s vary by orders of magnitude between rays traversing the center of the object and rays grazing the object's edge, so $\sum_k a_{nk} m_k \ll r_n$ for most n. Many of the background events remain separated in B_n. In contrast, in image restoration problems, if the point-spread function is roughly spatially invariant and the background rates $\{r_n\}$ are uniform, then the ratios $\{r_n/a_{n\cdot}\}$ will be fairly uniform and nearly all of the background events will be absorbed into $\{M_{nk}\}$.

Using a similar derivation as for (17) one can show:

$$Q_{\mathbf{X}^3}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) \equiv \sum_k \Big(-a_{\cdot k}(\lambda_k + m_k) + (\lambda_k^i + m_k) e_k(\boldsymbol{\lambda}^i) \log(\lambda_k + m_k) \Big), \quad (24)$$

where e_k was defined by (16). Like $Q_{\mathbf{X}^1}$, this function is also separable, and its partial derivatives are:

$$\frac{\partial}{\partial \lambda_k} Q_{\mathbf{X}^3}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) = -a_{\cdot k} + e_k(\boldsymbol{\lambda}^i) \frac{\lambda_k^i + m_k}{\lambda_k + m_k}.$$

To implement the M-step, one cannot simply maximize $Q_{\mathbf{X}^3}$ by zeroing its partial derivatives, because of the nonnegativity constraint. However, it is easy to verify that $Q_{\mathbf{X}^3}$ is a concave function with respect to λ_k, so that if its derivative vanishes at a negative λ_k, then the point $\lambda_k = 0$ will satisfy the Karush-Kuhn-Tucker conditions for the nonnegativity constraint (see Fig. 1). This leads to the ML-EM-3 algorithm, which, like ML-EM-1, is also a Type-I algorithm of Table 1, with (18) replaced by:

$$\lambda_k^{i+1} = \Big[(\lambda_k^i + m_k) e_k(\boldsymbol{\lambda}^i)/a_{\cdot k} - m_k \Big]_+, \quad (25)$$

where

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

[3] Under the conditions for global convergence discussed in Appendix I (e.g. strict concavity), the design parameters $\{m_k\}$ will affect only the rate of convergence of the EM sequence $\boldsymbol{\theta}^i$, but not the limit of that sequence. If the likelihood is not strictly concave, then the $\boldsymbol{\theta}^i$ limit will in general depend on both the starting value and the m_k's.

This is a simple 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 $m_k > 0$. In PET, since random coincidences are pervasive, we will have $r_n > 0$ for all n, so that $m_k > 0$ for all k.

Like ML-EM-1, since ML-EM-3 is an EM algorithm it monotonically increases the likelihood every iteration [15]. An interesting difference between the iterates generated by ML-EM-1 and ML-EM-3 is that the latter can move on and off the boundary of the nonnegative orthant from iteration to iteration. This may partly explain the faster convergence of ML-EM-3, since when ML-EM-1 converges to the boundary, it can do so at *sublinear* rates [12].

C. ML Line-Search Algorithms

caufman [58] noted that ML-EM-1 is the special case where $\alpha = 1$ of the form:

$$\lambda_k^{i+1} = \left[\lambda_k^i + \alpha \left(\frac{\lambda_k}{a_{\cdot k}}\right) \frac{\partial}{\partial \lambda_k} L(\boldsymbol{\lambda}^i)\right]_+ . \qquad (26)$$

The ML-LINB-1 and ML-LINU-1 algorithms [58] use a line-search to choose an $\alpha^i > 1$, which accelerates convergence. For ML-LINB-1, the search over α is *bounded* such that $\boldsymbol{\lambda}^{i+1}$ is positive, whereas ML-LINU-1 allows an *unconstrained* "bent line" search, in which α can be chosen large enough that some pixels would become negative, but are set to zero [58]. Similarly, ML-EM-3 is the special case where $\alpha = 1$ of the form:

$$\lambda_k^{i+1} = \left[\lambda_k^i + \alpha \left(\frac{\lambda_k + m_k}{a_{\cdot k}}\right) \frac{\partial}{\partial \lambda_k} L(\boldsymbol{\lambda}^i)\right]_+ . \qquad (27)$$

In the few PET experiments we tried, "accelerating" ML-EM-3 using a line-search to choose $\alpha^i > 1$ only slightly increased the convergence rate.

D. ML-SAGE Algorithms

The EM algorithms described above update all pixels simultaneously. Sauer and Bouman [10] have shown that sequential update methods often converge much faster than simultaneous update methods. There is also a subtle statistical motivation for using a sequential update: 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-3 the background events are distributed among all of the pixels, and the values for m_k are small. Therefore, we now derive three SAGE algorithms using individual pixels for the index sets: $S^i = \{k\}$, where $k = 1 + (i \bmod p)$.

The most obvious hidden-data for λ_k is just

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

which is a subset of the classical complete-data space (15). The $Q_{\mathbf{X}^{4,k}}$ function (6) for the kth parameter is therefore simply taken from (17):

$$Q_{\mathbf{X}^{4,k}}(\lambda_k; \boldsymbol{\lambda}^i) \equiv -a_{\cdot k}\lambda_k + \lambda_k^i e_k(\boldsymbol{\lambda}^i) \log \lambda_k.$$

Maximizing $Q_{\mathbf{X}^{4,k}}(\cdot; \boldsymbol{\lambda}^i)$ analytically yields the ML-SAGE-4 algorithm, which is a Type-III algorithm of Table 1 with the M-step (56) given by:

$$\lambda_k^{i+1} = \lambda_k^i e_k(\boldsymbol{\lambda}^i)/a_{\cdot k}. \qquad (28)$$

In words, Type-III algorithms update the parameters *sequentially*, and immediately update the predicted measurements $\bar{y}_n$ within the inner loop, whereas Type-I algorithms wait until all parameters have been updated[4].

The Fisher information for $\boldsymbol{X}^{4,k}$ is just the kth diagonal entry of $\boldsymbol{F}_{\mathbf{X}^1}$. It is therefore unsurprising that we have found ML-SAGE-4 to converge somewhat faster than ML-EM-1 for well conditioned problems but not for poorly conditioned problems. We now improve significantly on ML-SAGE-4 by introducing new hidden-data spaces similar to (20), only even less informative. The main idea is the following: *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 not very intuitive from the point of view of the imaging physics, but is completely admissible and sensible from a statistical perspective.

Define unobservable independent Poisson variates:

$$\begin{aligned} Z_{nk} &\sim \text{Poisson}\{a_{nk}(\lambda_k + z_k)\} \\ B_{nk} &\sim \text{Poisson}\{r_n - a_{nk}z_k + \sum_{j \neq k} a_{nj}\lambda_j\}, \end{aligned} \qquad (29)$$

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

$$a_{nk}z_k \le r_n + \sum_{j \neq k} a_{nj}\lambda_j^i, \; \forall n, \qquad (30)$$

so that the Poisson rates of B_{nk} are nonnegative. Note that this constraint is much less restrictive than (21). Then clearly

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

[4] Incremental updates like (57) will accumulate numerical error, so must be treated with caution if used repeatedly. Fortunately, the SAGE algorithms converge in a small number of iterations. In those rare occasions that we run SAGE for many iterations, we "reset" the estimated projections $\{\bar{y}_n\}$ using (13) roughly every 20 iterations.

has the appropriate distribution (11) for any k.

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

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

The Fisher information for $\boldsymbol{X}^{5,k}$ with respect to λ_k is the scalar value

$$F_{\mathbf{X}^5_k}(\hat{\lambda}_k) = a_{\cdot k}/(\hat{\lambda}_k + z_k),$$

which again suggests that we would like the z_k's to be as large as possible subject to the constraint (30).

We have investigated two choices for the z_k's. The first choice is independent of i:

$$z_k = z_k^0 = \min_{n: a_{nk} \neq 0} \{r_n / a_{nk}\}, \tag{31}$$

which clearly satisfies (30)[5]. This first choice is useful whenever the background rates $\{r_n\}$ are non-negligible. When the rates $\{r_n\}$ are negligible, the $\{z_k\}$ will be tiny, and ML-SAGE-5 is no better than ML-EM-1. However, since we are updating a single pixel, we can consider the contributions from all of the other pixels as “pseudo-background” events. This opportunity is indicated by the form of (29), which the reader should contrast with (20). Therefore, when the background rates are negligible, we use the following second choice for $\{z_k\}$, which is now dependent on iteration i:

$$\begin{aligned} z_k = z_k(\boldsymbol{\lambda}^i) &= \min_{n: a_{nk} \neq 0} \{(r_n + \sum_{j \neq k} a_{nj} \lambda_j^i)/a_{nk}\} \\ &= \min_{n: a_{nk} \neq 0} \{\bar{y}_n(\boldsymbol{\lambda}^i)/a_{nk}\} - \lambda_k^i. \end{aligned} \tag{32}$$

This choice also satisfies (30). Clearly $z_k(\boldsymbol{\lambda}^i) > z_k^0$, so using $z_k(\boldsymbol{\lambda}^i)$ should yield faster convergence. Nevertheless, the disadvantage of using $z_k = z_k(\boldsymbol{\lambda}^i)$ is that one must recompute the minimization (32) over n for every pixel each iteration, increasing the computation per iteration. Therefore we usually only use $z_k(\boldsymbol{\lambda}^i)$ when the background rates $\{r_n\}$ are negligible. These tradeoffs are illustrated in Sections V and VI.

The definition (31) of z_k involves only a single a_{nk} in each denominator, rather than the sum $a_{n\cdot}$ contained in the definition (23) of m_k. Thus, the values of z_k^0 and $z_k(\boldsymbol{\lambda}^i)$ are orders of magnitude larger than m_k, and 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. Therefore $F_{\mathbf{X}^5_k}$ is much smaller than the kth diagonal entry of $\boldsymbol{F}_{\mathbf{X}^3}$.

Using a similar derivation as for $Q_{\mathbf{X}^3}$, one can show:

$$Q_{\mathbf{X}^{5,k}}(\lambda_k; \boldsymbol{\lambda}^i) \equiv$$

$$-a_{\cdot k}(\lambda_k + z_k) + (\lambda_k^i + z_k)\, e_k(\boldsymbol{\lambda}^i) \log(\lambda_k + z_k). \tag{33}$$

Maximizing $Q_{\mathbf{X}^{5,k}}(\cdot; \boldsymbol{\lambda}^i)$ analytically (subject to the nonnegativity constraint), yields the ML-SAGE-5 algorithm, which is also a Type-III algorithm of Table 1, with (28) replaced by:

$$\lambda_k^{i+1} = \left[(\lambda_k^i + z_k^0) e_k(\boldsymbol{\lambda}^i)/a_{\cdot k} - z_k^0\right]_+. \tag{34}$$

This is a small change to ML-SAGE-4, but one that significantly accelerates convergence. Indeed, the implementation differences between ML-EM-1, ML-EM-3, ML-SAGE-4, and ML-SAGE-5 are all remarkably minor, but the differences in convergence rates are quite large, as illustrated by the results in Section V.

For clarity, we refer to the algorithm based on the choice $z_k = z_k(\boldsymbol{\lambda}^i)$ as ML-SAGE-6, which can be written:

$$\lambda_k^{i+1} = \left[(\lambda_k^i + z_k(\boldsymbol{\lambda}^i)) e_k(\boldsymbol{\lambda}^i)/a_{\cdot k} - z_k(\boldsymbol{\lambda}^i)\right]_+. \tag{35}$$

IV. Penalized Maximum Likelihood

Since image reconstruction is ill-conditioned, regularization is very desirable. We described the maximum likelihood algorithms above primarily to introduce the new hidden data spaces. In this section we turn to regularized image reconstruction using penalized likelihood objectives. We first present a new SAGE algorithm based on the hidden-data spaces $\{\boldsymbol{X}^{5,k}\}$. To provide a fair comparison with alternative methods, we also derive new versions of the GEM algorithm of Hebert and Leahy [32], the parallelizable algorithm of De Pierro [39], and the one-step late algorithm of Green [43], all using the new complete-data space $\boldsymbol{X}^3$. As we show in Section V, these modified algorithms based on $\boldsymbol{X}^3$ all converge somewhat faster than their original versions based on $\boldsymbol{X}^1$, but none converge as fast as SAGE on a conventional serial computer. Nevertheless, they may be useful in some pixel-based parallel computing environments, and they allow us to perform the most conservative comparison between SAGE and its alternatives.

We have implemented all of the algorithms given below with several convex penalty functions. However, to give explicit expressions for the algorithms without undue notation, we first focus on a simple quadratic smoothness penalty. At the end of the section we briefly discuss how to implement the non-quadratic case, which is fairly

[5] Note that these z_k's would violate (21), so attempting to substitute z_k for m_k in ML-EM-3 would violate the admissibility condition for hidden data spaces (5) and destroy the monotonicity of ML-EM-3.

straightforward once the quadratic case is understood. The quadratic smoothness penalty used below is:

$$P(\boldsymbol{\lambda}) = \beta \frac{1}{2} \sum_k \sum_{j \in \mathcal{N}_k} \frac{1}{2} w_{kj} (\lambda_k - \lambda_j)^2 \qquad (36)$$

where $\mathcal{N}_k$ is a neighborhood of the kth pixel and $w_{kj} = w_{jk}$. In the work reported in Section V, we let $\mathcal{N}_k$ be the 8 pixels adjacent to the kth pixel, and set $w_{kj} = 1$ for horizontal and vertical neighbors and $w_{kj} = 1/\sqrt{2}$ for diagonal neighbors. Combining this penalty with the log-likelihood (12) yields the penalized likelihood objective function (1):

$$\Phi(\boldsymbol{\lambda}) = \sum_n (-\bar{y}_n(\boldsymbol{\lambda}) + y_n \log \bar{y}_n(\boldsymbol{\lambda})) - \beta P(\boldsymbol{\lambda}).$$

It is easy to show that Φ is strictly concave for the penalty given by (36), under mild conditions on $\boldsymbol{A}$. Our goal is to maximize Φ.

A. Penalized SAGE Algorithm

For simplicity, our SAGE algorithms for the penalized maximum-likelihood case use single-pixel index sets[6]: $S^i = \{k\}$, where $k = 1 + (i \operatorname{modulo} p)$. We have implemented penalized maximum likelihood SAGE algorithms with both the $\boldsymbol{X}^{4,k}$ and $\boldsymbol{X}^{5,k}$ hidden-data spaces. The $\boldsymbol{X}^{4,k}$ version is essentially identical to the "ICM-EM" algorithm of Abdalla and Kay [40]. The $\boldsymbol{X}^{5,k}$ version is significantly faster, so we focus on that case. Following (7), define

$$\begin{aligned} \phi^{5,k}(\lambda_k; \boldsymbol{\lambda}^i) &= Q_{\mathbf{X}^{5,k}}(\lambda_k; \boldsymbol{\lambda}^i) - P(\lambda_k, \boldsymbol{\lambda}^i_{-k}) \\ \equiv -a_{\cdot k}(\lambda_k + z_k) &+ (\lambda_k^i + z_k) e_k(\boldsymbol{\lambda}^i) \log(\lambda_k + z_k) \\ &- \beta \sum_{j \in \mathcal{N}_k} w_{kj} \frac{1}{2} (\lambda_k - \lambda_j^i)^2, \qquad (37) \end{aligned}$$

where $Q_{\mathbf{X}^{5,k}}$ was defined in (33), and $\boldsymbol{\lambda}^i_{-k}$ is the vector of length $(p-1)$ obtained by removing the kth element from $\boldsymbol{\lambda}^i$. The M-step (3) requires maximizing $\phi^{5,k}(\cdot; \boldsymbol{\lambda}^i)$, which we can do analytically by zeroing its derivative since $\phi^{5,k}(\lambda_k; \boldsymbol{\lambda}^i)$ is a strictly concave function of λ_k. The derivative of $\phi^{5,k}(\cdot; \boldsymbol{\lambda}^i)$ with respect to λ_k is:

$$\frac{\partial}{\partial \lambda_k} \phi^{5,k}(\lambda_k; \boldsymbol{\lambda}^i) =$$

$$-a_{\cdot k} + e_k(\boldsymbol{\lambda}^i) \frac{\lambda_k^i + z_k}{\lambda_k + z_k} - \beta \sum_{j \in \mathcal{N}_k} w_{kj} (\lambda_k - \lambda_j^i).$$

Of course, since we are only updating one parameter, there is no problem with coupled equations. Equating this derivative to zero yields a quadratic formula:

$$A_k (\lambda_k + z_k)^2 + 2 B_k (\lambda_k + z_k) - C_k = 0,$$

where

$$\begin{aligned} A_k &= \beta \sum_{j \in \mathcal{N}_k} w_{kj} \\ B_k &= \frac{a_{\cdot k} - \beta \sum_{j \in \mathcal{N}_k} w_{kj} (\lambda_j^i + z_k^0)}{2} \\ C_k &= e_k(\boldsymbol{\lambda}^i)(\lambda_k^i + z_k^0). \end{aligned}$$

Just as in the derivation of (25), the constrained maximum of $\phi^{5,k}(\cdot; \boldsymbol{\lambda}^i)$ corresponds to either the positive root of the quadratic, or the value $\lambda_k = 0$, since $\phi^{5,k}$ is strictly concave. This leads to the PML-SAGE-5 algorithm, which is an algorithm of Type-III in Table 1 with the M-step (56) given by:

$$\lambda_k^{i+1} = \left[\frac{-B_k + \sqrt{B_k^2 + A_k C_k}}{A_k} - z_k^0 \right]_+ . \qquad (38)$$

We refer PML-SAGE-6 as the version of (38) where $z_k = z_k(\boldsymbol{\lambda}^i)$, as defined by (32). Again, we usually only use PML-SAGE-6 when the background rates $\{r_n\}$ are negligible. In words, we first compute the e_k correction term from the current projection estimate, then update the kth pixel using a quadratic formula that involves both the data and the neighboring pixels, and then immediately update the projection estimate before proceeding to the next pixel. In practice, the actual implementation has two important differences: 1) the pixels are updated in four different raster scan orders rather than using the same order each iteration (cf frequency analysis in [10]), and 2) the quadratic formula is computed using numerically stable formulae [60, p. 156] rather than the conventional form (38), i.e.

$$\lambda_k^{i+1} = \left[\frac{C_k}{B_k + \sqrt{B_k^2 + A_k C_k}} - z_k \right]_+ .$$

Note that as $\beta \to 0$, this last formula approaches the unpenalized update (34). Global convergence of PML-SAGE-5 and PML-SAGE-6 is established in Appendix I.

One "generalization" of this algorithm that could be explored further is the following. Rather than updating each pixel once using (38), we could loop M times over each pixel before moving onto the next pixel. This is like having a miniature EM iteration within every pixel update. In the limit as M increases, this algorithm would approach

[6] It is certainly feasible to update more than one pixel at a time, with some increase in the complexity of the M-step. Such tradeoffs are a subject for future exploration.

a coordinate ascent of $\Phi(\boldsymbol{\lambda})$. In the few experiments we have tried, this performed no better than using $M = 1$, which is consistent with the benefits of under-relaxation in successive over-relaxation methods [19], as demonstrated in Figures 8, 10, and 12.

B. Modified GEM Algorithm

The GEM algorithm for image reconstruction [32] is a very intuitive approach to extending the EM algorithm to the penalized maximum-likelihood case. Rather than using $\boldsymbol{X}^1$ as in [32], we now develop a GEM algorithm using the new complete-data space $\boldsymbol{X}^3$. Following (7), let

$$\phi^3(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) = Q_{\mathbf{X}^3}(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) - P(\boldsymbol{\lambda}),$$

where $Q_{\mathbf{X}^3}$ was defined in (24). The GEM algorithm is similar to the special case of the SAGE algorithm of Section II where $S^i = \{1, \ldots, p\}$ and $\phi^i = \phi^3$ for all i. Thus, the M-step (3) requires us to maximize $\phi^3(\cdot; \boldsymbol{\lambda}^i)$. Unfortunately, its partial derivatives are coupled:

$$\frac{\partial}{\partial \lambda_k} \phi^3(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) =$$

$$-a_{\cdot k} + e_k(\boldsymbol{\lambda}^i) \frac{\lambda_k^i + m_k}{\lambda_k + m_k} - \beta \sum_{j \in \mathcal{N}_k} w_{kj}(\lambda_k - \lambda_j). \quad (39)$$

This coupling prohibits analytical maximization. The basic idea behind the GEM method [32] is to forgo maximization in favor of simply increasing $\phi^3(\cdot; \boldsymbol{\lambda}^i)$ using a coordinate-ascent method. Increasing ϕ^3 using coordinate-ascent is easier than maximizing $\Phi(\cdot)$ by coordinate ascent since we can solve (39) with respect to λ_k (while holding the other parameters fixed) using essentially the same quadratic formula as (38). Extending the derivation in [31, 32] leads to the PML-GEM-3 algorithm, which is a Type-II algorithm of Table 1, with the M-step (54) given by:

$$\begin{aligned} A_k &= \beta \sum_{j \in \mathcal{N}_k} w_{kj} \\ B_k &= \frac{a_{\cdot k} - \beta \sum_{j \in \mathcal{N}_k} w_{kj}(\lambda_j^\star + m_k)}{2} \\ C_k &= e_k(\boldsymbol{\lambda}^i)(\lambda_k^i + m_k) \end{aligned}$$

$$\lambda_k^{i+1} = \left[\frac{-B_k + \sqrt{B_k^2 + A_k C_k}}{A_k} - m_k \right]_+ . \quad (40)$$

In this pseudo-code, $\lambda_k^\star$ denotes the *most recent* estimate of λ_k, e.g.:

$$\lambda_j^\star = \begin{cases} \lambda_j^{i+1}, & j < k \\ \lambda_j^i, & j \geq k \end{cases} .$$

In other words, the updates are done "in place". We refer to the conventional GEM algorithm based on $\boldsymbol{X}^1$ (where $m_k = 0 \, \forall k$) as PML-GEM-1.

Following [32], we usually cycle M times through the inner loop over k with different raster scan orders, so that the coordinate ascent can approach the maximum of $\phi^3(\cdot; \boldsymbol{\lambda}^i)$. Typically $M = 2$ seems adequate. This loop over M is relatively inexpensive since no projections are recomputed within it. Since A_k is independent of i, it can be precomputed, and since C_k is independent of $\lambda_k^\star$, it is initialized *before* the cycle over M.

One can easily verify that λ_k^{i+1} given by (40) satisfies the one-dimensional Karush-Kuhn-Tucker conditions with respect to the nonnegativity constraint. Thus PML-GEM-3 yields a sequence of estimates $\{\boldsymbol{\lambda}^i\}$ that monotonically increase the objective Φ. Global convergence of GEM follows from Theorem 3 of [35], provided the objective is strictly concave.

Note that PML-SAGE-5 and PML-GEM-3 are somewhat similar, except that PML-SAGE-5 uses the less informative hidden data space $\boldsymbol{X}^5$, and it updates the projections immediately after each parameter update. Although subtle, these two differences lead to PML-SAGE-5 converging significantly faster.

C. Modified De Pierro Algorithm

An alternate approach to circumventing the coupled equations is the novel majorization method of De Pierro [38, 39]. This monotonic method has the advantage that it is more parallelizable than GEM, and it is globally convergent[7]. This method applies a decomposition to the penalty $P()$ that is similar in concept to the decomposition that relates the log-likelihood to the Q function. First, note that for any convex function h,

$$h(ax + by) =$$

$$h\left(\frac{1}{2}(ax^i + 2by - by^i) + \frac{1}{2}(2ax + by^i - ax^i)\right)$$

$$\leq \frac{1}{2} h(ax^i + 2by - by^i) + \frac{1}{2} h(2ax + by^i - ax^i).$$

Thus, by defining

$$P^\circ(\boldsymbol{\lambda}; \boldsymbol{\lambda}^i) =$$

$$\frac{\beta}{8} \sum_k \sum_{j \in \mathcal{N}_k} (\lambda_k^i - 2\lambda_j + \lambda_j^i)^2 + (2\lambda_k - \lambda_j^i - \lambda_k^i)^2,$$

[7] Global convergence for De Pierro's method with the $\boldsymbol{X}^1$ complete-data space was shown in [39]. The $\boldsymbol{X}^3$ complete-data space version herein converges globally by a special case of our proof in Appendix I.

it follows that

$$\begin{aligned} P^\circ(\boldsymbol{\lambda};\boldsymbol{\lambda}) &= P(\boldsymbol{\lambda}) \\ P^\circ(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) &\geq P(\boldsymbol{\lambda}) \\ \nabla^{10} P^\circ(\boldsymbol{\lambda};\boldsymbol{\lambda}) &= \nabla^{10} P(\boldsymbol{\lambda}). \end{aligned}$$

De Pierro [39] used $Q_{\mathbf{X}^1}$; here we use $Q_{\mathbf{X}^3}$ to define:

$$\phi^\circ(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) = Q_{\mathbf{X}^3}(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) - P^\circ(\boldsymbol{\lambda};\boldsymbol{\lambda}^i).$$

Thus, our modified method of De Pierro is the special case of the SAGE algorithm in Section II with $S^i = \{1,\ldots,p\}$ and $\phi^i = \phi^\circ$ for all i. Note that the above construction of ϕ° is somewhat different than the formulation given by (7), but nevertheless ϕ° satisfies the essential condition (2). Remarkably, by this construction $\phi^\circ(\cdot;\boldsymbol{\lambda}^i)$ is separable, with partial derivatives given by

$$\begin{aligned} \frac{\partial}{\partial\lambda_k}\phi^\circ(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) = -a_{\cdot k} + e_k(\boldsymbol{\lambda}^i)\frac{\lambda_k^i + m_k}{\lambda_k + m_k} \\ -\beta\sum_{j\in\mathcal{N}_k} w_{kj}(2\lambda_k - \lambda_j^i - \lambda_k^i). \end{aligned}$$

Thus we can maximize $\phi^\circ(\cdot;\boldsymbol{\lambda}^i)$ by zeroing the partial derivative (and minding the Karush-Kuhn-Tucker conditions). This leads to the PML-DePierro-3 algorithm, which is a Type-I algorithm of Table 1 with the M-step (53) given by:

$$\begin{aligned} A_k &= 2\beta\sum_{j\in\mathcal{N}_k} w_{kj} \\ B_k &= \frac{a_{\cdot k} - \beta\sum_{j\in\mathcal{N}_k} w_{kj}(\lambda_j^i + m_k) - \lambda_k^i A_k/2}{2} \\ C_k &= e_k(\boldsymbol{\lambda}^i)(\lambda_k^i + m_k). \end{aligned}$$

$$\lambda_k^{i+1} = \left[\frac{-B_k + \sqrt{B_k^2 + A_k C_k}}{A_k} - m_k\right]_+. \qquad (41)$$

Strictly speaking, this method is actually a type of GEM algorithm since maximizing ϕ° does not yield the maximum of $\phi^3(\cdot;\boldsymbol{\lambda}^i)$ [5]. We have found empirically and theoretically [16] that PML-DePierro-3 converges slightly slower than PML-GEM-3 on a serial computer. Indeed, one can compare (41) with (40) to see that PML-DePierro-3 takes slightly smaller steps than PML-GEM-3. However, although not noted in [38, 39], one can add a loop analogous to the loop over M in the PML-GEM-3 algorithm, which then leads to comparable performance to PML-GEM-3. Again, typically $M = 2$ sub-iterations is adequate. Since the two algorithms have comparable performance on serial computers, we focus on the GEM algorithm in the next section. We include the modified De Pierro algorithm here because of its potential use with parallel computers.

D. *Modified One-Step-Late (OSL) Algorithm*

Green's OSL algorithm [43] “avoids” the problem of coupled equations by linearizing the penalty function, or equivalently, by substituting the parameter estimates from the previous iteration into the derivative of the penalty. We follow the development in [43], but substitute $Q_{\mathbf{X}^3}$ for the conventional $Q_{\mathbf{X}^1}$. Using $Q_{\mathbf{X}^3}$ with $S = \{1,\ldots,p\}$, from (3) the M-step requires maximizing

$$\phi^3(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) = Q_{\mathbf{X}^3}(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) - P(\boldsymbol{\lambda}),$$

where $Q_{\mathbf{X}^3}$ was defined in (24). Ignoring nonnegativity constraints, this maximization is equivalent to solving

$$\mathbf{0} = \nabla_{\boldsymbol{\lambda}}\left(Q_{\mathbf{X}^3}(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) - P(\boldsymbol{\lambda})\right).$$

The $\nabla_{\boldsymbol{\lambda}} Q_{\mathbf{X}^3}(\cdot;\boldsymbol{\lambda}^i)$ term is separable, but $\nabla_{\boldsymbol{\lambda}} P(\boldsymbol{\lambda})$ is not, so the suggestion of Green [43] is to assume

$$\nabla_{\boldsymbol{\lambda}} P(\boldsymbol{\lambda})\big|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^{i+1}} \approx \nabla_{\boldsymbol{\lambda}} P(\boldsymbol{\lambda})\big|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^i}.$$

Ironically, this approximation is particularly good for slow converging algorithms! Under that approximation, one has

$$\begin{aligned} \frac{\partial}{\partial\lambda_k}\phi^3(\boldsymbol{\lambda};\boldsymbol{\lambda}^i) \approx \\ -a_{\cdot k} + e_k(\boldsymbol{\lambda}^i)\frac{\lambda_k^i + m_k}{\lambda_k + m_k} - \beta\sum_{j\in\mathcal{N}_k} w_{kj}(\lambda_k^i - \lambda_j^i), \end{aligned}$$

which is now separable, so it can be treated analytically by the Karush-Kuhn-Tucker conditions. This leads to the PML-OSL-3 algorithm, which is a Type-I algorithm of Table 1, with the M-step (53) given by:

$$\lambda_k^{i+1} = \left[\frac{(\lambda_k^i + m_k)e_k(\boldsymbol{\lambda}^i)}{a_{\cdot k} + \beta\sum_{j\in\mathcal{N}_k} w_{kj}(\lambda_k^i - \lambda_j^i)} - m_k\right]_+. \qquad (42)$$

This approach is popular due to its simplicity, but it *can diverge*, particularly for large values of β. We include it for the purpose of comparison with PML-SAGE. We refer to the case of (42) where $m_k = 0\ \forall k$ as PML-OSL-1.

It is straightforward to show [46] that PML-OSL-1 and PML-OSL-3 can be expressed in the form (cf (27))

$$\lambda_k^{i+1} = \left[\lambda_k^i + \alpha\left(\frac{\lambda_k + m_k}{a_{\cdot k} + \frac{\partial}{\partial\lambda_k}P(\boldsymbol{\lambda}^i)}\right)\frac{\partial}{\partial\lambda_k}\Phi(\boldsymbol{\lambda}^i)\right]_+.$$

Therefore, one can also accelerate PML-OSL-1,3 and/or make them have global monotonic convergence by choosing α using a line-search [46]. For the case where $m_k = 0$, we refer to these algorithms as PML-LINB-1 and PML-LINU-1 for the bounded and unbounded searches for α (cf Section III.C).

E. Coordinate-wise Newton Raphson (CNR)

Bouman and Sauer [55, 56] have proposed a non-EM type of algorithm for maximizing $\Phi(\boldsymbol{\lambda})$ based on applying coordinate-ascent directly to the objective, which circumvents the problem of coupled parameters due to the penalty function. They used a one-dimensional Newton-Raphson update, which is based on a second order Taylor's approximation of the log-likelihood. Without a line-search, monotonicity is not guaranteed when using this approximation. Their method is equivalent to the following expansion:

$$\Phi(\boldsymbol{\theta}_S, \boldsymbol{\theta}^i_{\tilde{S}}) \approx L(\boldsymbol{\theta}^i) - \beta P(\boldsymbol{\theta}_S, \boldsymbol{\theta}^i_{\tilde{S}})$$
$$+\dot{\boldsymbol{L}}_S(\boldsymbol{\theta}^i)(\boldsymbol{\theta}_S - \boldsymbol{\theta}^i_S) - \frac{1}{2}(\boldsymbol{\theta}_S - \boldsymbol{\theta}^i_S)' \ddot{\boldsymbol{L}}_S(\boldsymbol{\theta}^i)(\boldsymbol{\theta}_S - \boldsymbol{\theta}^i_S), \quad (43)$$

where

$$\begin{aligned} \dot{\boldsymbol{L}}_S(\boldsymbol{\theta}^i) &= \nabla'_{\boldsymbol{\theta}_S} L(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^i} \\ \ddot{\boldsymbol{L}}_S(\boldsymbol{\theta}^i) &= -\nabla^2_{\boldsymbol{\theta}_S} L(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^i}. \end{aligned}$$

For a sequential update, one simply takes $S^i = \{k\} = \{1 + (i \text{ modulo } p)\}$, in which case:

$$\begin{aligned} \dot{\boldsymbol{L}}_k(\boldsymbol{\lambda}) &= \frac{\partial}{\partial \lambda_k} L(\boldsymbol{\lambda}) = e_k(\boldsymbol{\lambda}) - a_{\cdot k}, \\ \ddot{\boldsymbol{L}}_k(\boldsymbol{\lambda}) &= l_k(\boldsymbol{\lambda}) = -\frac{\partial^2}{\partial \lambda_k^2} L(\boldsymbol{\lambda}) = \sum_n a_{nk}^2 y_n / \bar{y}_n(\boldsymbol{\lambda})^2. \end{aligned}$$

Thus

$$\Phi(\lambda_k, \boldsymbol{\lambda}^i_{-k}) \approx L(\boldsymbol{\lambda}^i) + (e_k(\boldsymbol{\lambda}^i) - a_{\cdot k})(\lambda_k - \lambda^i_k)$$
$$-\frac{1}{2} l_k(\boldsymbol{\lambda}^i)(\lambda_k - \lambda^i_k)^2 - \beta \sum_{j \in \mathcal{N}_k} w_{kj} \frac{1}{2}(\lambda_k - \lambda^i_j)^2,$$

dropping terms independent of λ_k as usual. We can thus update λ_k by zeroing the derivative of the above approximation to $\Phi(\cdot, \boldsymbol{\lambda}^i_{-k})$. This leads to the PML-CNR algorithm, which is an algorithm similar to Type-III in Table 1 with the step (56) given by:

$$\begin{aligned} l_k &= \sum_n a_{nk}^2 y_n / \bar{y}_n^2 \\ \lambda_k^{\text{new}} &= \left[\frac{\lambda^i_k l_k + e_k(\boldsymbol{\lambda}^i) - a_{\cdot k} + \beta \sum_{j \in \mathcal{N}_k} w_{kj} \lambda^i_j}{l_k + \beta \sum_{j \in \mathcal{N}_k} w_{kj}} \right]_+ \\ \lambda_k^{i+1} &= \omega \lambda_k^{\text{new}} + (1 - \omega) \lambda^i_k. \end{aligned}$$

The algorithm presented in [55, 56] was for the case $\omega = 1$. We have added the parameter ω because under-relaxation (i.e., $\omega < 1$) is often useful for sequential methods [19], as we confirm in Section V. The PML-CNR algorithm is more expensive per-iteration than PML-SAGE-5, since one must compute the second derivative terms l_k. Including a line search to enforce monotonicity would add considerable expense.

F. Nonquadratic penalties

The SAGE method is not limited to quadratic penalties. One can easily implement other penalty functions such as the flexible penalty introduced by De Pierro [39, 59]:

$$P(\boldsymbol{\lambda}) = \sum_c h^c(\langle \boldsymbol{w}^c, \boldsymbol{\lambda} \rangle), \quad (44)$$

where each $\boldsymbol{w}^c$ is a vector of length p, and $\{h^c\}$ are potential functions such as those proposed in [41, 46].

For the proof of convergence in Appendix I, we assume that each function $h^c(\cdot)$ is strictly convex. Convexity is not needed to *define* or to *implement* the algorithms. However, we expect that if the convexity condition is violated, then all of the algorithms will have only local convergence rather than global convergence. Whether local convergence is acceptable will depend on many factors, including the quality of the initial estimate. Methods such as “deterministic annealing” [34] may be necessary to get good results for non-convex penalties.

For most non-quadratic penalty functions, there are not analytical forms for the maxima of the ϕ^i functionals. Therefore, for convex but non-quadratic objectives, we apply a single one-dimensional Newton-Raphson step to $\phi^{5,k}(\lambda_k; \boldsymbol{\lambda}^i)$ with respect to λ_k. We then use (37) to see if $\phi^{5,k}$ increased; if not, we halve the step size until it is increased. This ensures that PML-SAGE-5 and PML-SAGE-6 will be monotonic even with non-quadratic penalties. This halving search is inexpensive since evaluating (37) does not require reprojections, in contrast to an interval search applied to the objective function Φ.

Strictly speaking this halving approach does not quite meet the requirements of our global convergence proof in Appendix I, since λ_k^{i+1} will not exactly satisfy the Karush-Kuhn-Tucker conditions. Of course, finite-precision computing is never exact, so global convergence proofs should not be interpreted too literally. One could apply multiple Newton-Raphson steps rather than just one, but we doubt that the extra effort would be worthwhile. However, we conjecture that one could extend the convergence proof of Lange [46] to prove global convergence of SAGE for cases where the lack of a closed form for the M-step requires the use of 1-D interval searches.

An alternative method for “preserving edges” is to use penalty functions based on augmenting the emission parameters with a line process [62–64]. The SAGE method is applicable to such augmented objective functions, although with the same caveats about non-concavity that apply to other deterministic optimization methods.

V. Simulation Methods

We now provide some anecdotal results that demonstrate that the new complete-data spaces lead to faster convergence rates for the EM-type algorithms, and that SAGE methods converge even faster. These empirical results corroborate the analysis given in [12, 15, 16]. Our purpose is only to compare convergence rates, not to argue whether or not penalized maximum-likelihood images are better in any sense than filtered backprojection (FBP) images. Of course we hope that providing a new algorithm for rapidly computing images using statistical criteria will facilitate more comprehensive comparisons of image reconstruction methods in future work.

We have evaluated the algorithms' convergence rates using a 2-D slice of the digital Hoffman brain phantom shown in Fig. 2, with intensity 4 in the gray matter, 1 in the white matter, and 0 in the background. (All images are displayed by mapping the range [0,5] to gray levels [0,255].) The phantom is discretized on a 80 by 110 grid with 2mm square pixels. The phantom was forward projected using precomputed factors a_{nk} corresponding to an idealized PET system having 100 angles evenly spaced over $180°$, and 70 radial samples with 3mm spacing. Each a_{nk} was precomputed as the area of intersection between the square pixel and a strip of width 6mm. (Since the strip width is wider than the radial spacing, the strips overlap.) The detector response is thus a 6mm rectangular function. Only image pixels within a support ellipse of radii 39 by 54 pixels were reconstructed.

The projections were multiplied by nonuniform attenuation factors corresponding to an ellipse with radii 90 and 100 mm with attenuation coefficient 0.01/mm, surrounded by an elliptical 5mm thick skull with attenuation coefficient 0.015/mm. Nonuniform detector efficiencies were applied using pseudo-random log-normal variates with standard deviation 0.2. The sinogram was globally scaled to a mean sum of 900000 true events. All of the above effects were also incorporated into the a_{nk} factors. Pseudo-random independent Poisson variates were drawn according to (11), and a uniform field of Poisson distributed background events with known mean were added. Three data sets were studied, one with 5% background events, another with 35% background events, representing the range of random coincidence contamination typically found in PET scans, and one with 0% background events. Having no random coincidences is impossible in PET, but we include this case since the results may be of interest for other applications.

For the unpenalized maximum-likelihood algorithms, the initial estimate $\boldsymbol{\lambda}^0$ was a uniform ellipse. For the penalized maximum-likelihood algorithms the initial estimate was the image formed by applying FBP using a 3rd order Butterworth filter with cutoff 0.6 of Nyquist (10mm resolution). FBP image values below 0.1 were set to 0.1 so that $\boldsymbol{\lambda}^0$ was nonnegative.

VI. Results

The main results are illustrated by Figures 3-21. Not all algorithms are shown in all figures for the following reasons. We found that the LINU algorithms converged faster than the LINB algorithms only in the 0% background cases, so the LINU results are shown only in those cases. In the 0% background cases, all “-5” and “-4” algorithms are identical $(z_k^0 = 0)$, as are all “-3” and “-1” algorithms $(m_k = 0)$, so the -5 and -3 algorithms are not shown.

A. Maximum Likelihood

Figures 3-5 display the unpenalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ versus iteration for several of the maximum likelihood algorithms discussed in Section III. The following points are illustrated by these results.

- ML-EM-3 converges only slightly faster than ML-EM-1, although the difference grows with increasing background fraction. ML-LINU-1 converges faster than ML-EM-3.
- ML-SAGE-5 and ML-SAGE-6 converge faster than ML-LINU-1, and appear to reach an asymptote sooner. The difference grows with increasing background fraction. (ML-SAGE-5 is also easier to implement than the bent-line ML-LINU-1 method.) ML-SAGE-5 converges faster than ML-LINU-1 even when the background fraction is as small as 5%.
- For 5% and 35% background fractions, ML-SAGE-5 increases the likelihood faster than ML-SAGE-6 during the early iterations, but by the 10-20th iteration, ML-SAGE-6 passes ML-SAGE-5. In light of Figures 7-12, it may be useful to under-relax ML-SAGE-6.
- For 0% background events, the difference between ML-SAGE-6 and ML-LINU-1 is minimal, so for ML reconstruction, the SAGE methods presented in this report are most useful when the background is non-negligible.

Qualitatively, ML-SAGE-5 images exhibit the infamous noisy checkerboard effect in an order of magnitude fewer iterations than even ML-LINU-1, so some regularization method is clearly necessary[8].

B. Penalized Maximum Likelihood

Figures 7-12 display the penalized likelihood objective $\Phi(\boldsymbol{\lambda}^i) - \Phi(\boldsymbol{\lambda}^0)$ for the EM-type algorithms, including "close ups" of $\Phi(\boldsymbol{\lambda}^i) - \Phi(\boldsymbol{\lambda}^0)$ for PML-SAGE-5 and -6 and PML-CNR in the early iterations. It is also interesting to examine the rates of convergence in L_2 *norm,* i.e.

$$\|\boldsymbol{\lambda}^i - \hat{\boldsymbol{\lambda}}\| = \sum_k \lambda_k^i - \hat{\lambda}_k.$$

Unfortunately $\hat{\boldsymbol{\lambda}}$ is not known exactly, and cannot be computed exactly with finite precision computers. For illustrative purposes, we took $\hat{\boldsymbol{\lambda}}$ to be the 100th iteration of PML-SAGE-5, at which point it had converged to within single floating point machine precision. Figure 6 compares the norm convergence rates for the algorithms.

The following points are illustrated by these results.

- In all cases, GEM and OSL (and De Pierro' algorithm, not shown) had indistinguishable convergence rates.
- PML-GEM-3 and PML-OSL-3 converge faster than the conventional PML-GEM-1 and PML-OSL-1 respectively, and the increase in speed grows with the background fraction.
- Even with only 5% random coincidences, PML-SAGE-5 clearly increases faster and reaches its asymptote sooner than PML-GEM-3 and PML-OSL-3. The advantage for 35% background is even greater.
- For 0% background events, $z_k^0 = 0$, so PML-SAGE-5 is identical to PML-SAGE-4 (which is identical to the "ICM-EM" algorithm of [40]), and converges at the same rate as PML-GEM-1. For 0% background PML-SAGE-6 converges faster per-iteration than PML-GEM-1 or PML-LINU-1.
- We experimented with several values of ω for PML-CNR for this data set, and found it converged fastest when $\omega = 0.6$, i.e., which PML-CNR is *under*-relaxed. Using this under-relaxation, the convergence rates of PML-CNR and PML-SAGE-5 (or PML-SAGE-6 in the case of 0% background) were quite comparable for these data sets.
- The conclusions given above in terms of the convergence in the objective function $\Phi(\boldsymbol{\lambda}^i)$ also held true for convergence in L_2 norm, as shown in Fig. 6. The slopes of the lines in this logarithmic plot is related to the asymptotic convergence rate, and one can see that PML-SAGE-5 and PML-CNR (with $\omega = 0.6$) converge significantly faster than the other algorithms.

Since PML-SAGE-5 is a monotonic algorithm applied to a strictly concave objective function, it is very robust to the initial estimate. Figures 13 and 14 display several iterations of PML-SAGE-5 estimates initialized with a uniform image, a checkerboard image, and a FBP image. The difference images rapidly decrease to values that would be invisible on a conventional 8-bit display, so we have amplified the differences by a factor of 4 for display here. For 35% background events the effects of the checkerboard initial estimate are negligible by 8-10 iterations; for 5% background the effects of the initial estimate are negligible by 15-20 iterations.

As discussed in Section IV, SAGE is also applicable to non-quadratic penalties. The images in Figure 15 were reconstructed by applying PML-SAGE-5 to a penalized-likelihood objective with the following penalty function:

$$P(\boldsymbol{\lambda}) = \beta \frac{1}{2} \sum_k \sum_{j \in \mathcal{N}_k} w_{kj} h(\lambda_k - \lambda_j),$$

where

$$h(u) = \frac{1}{2}\delta^2 \left(|u/\delta| - \log(1 + |u/\delta|) \right),$$

where we used $\delta = 0.8$. This penalty is one of several suggested by Lange [46, Table III]. Figure 15 demonstrates that the iterates produced by PML-SAGE-5 converge rapidly even for non-quadratic penalties.

VII. Computation

Table II summarizes the computation times for 40 iterations on a DEC 3000/400 workstation. Also shown is the floating point operations (flops) for the algorithms. Based on flops alone, ML-SAGE-5 should at worst take 25% more time per iteration than ML-EM-1. The actually CPU time for ML-SAGE-5 was about 72% higher than ML-EM-1 per iteration, so apparently either floating point operations do not solely dominate the CPU time, or further code optimization is needed. Figures 16-21 are essentially the same as Figures 7-12¡ except that we have plotted CPU time on the horizontal axis. Even though our implementation of the SAGE algorithms runs slower than the floating point calculations would suggest, the curves in Figs. 16-21 demonstrate the SAGE algorithms converge faster than the other monotonic algorithms, and the gap widens with increasing background fraction. The reader should bear in

[8] Fast convergence is clearly desirable for penalized objective functions, but we advise caution when using "stopping rules" [18] in conjunction with coordinate-based algorithms (such as ML-SAGE-5) for the unpenalized case, since for such algorithms the *high* spatial frequencies converge faster than the low frequencies [10].

mind that these comparisons could vary significantly between implementations.

Why does ML-SAGE-5 require about 25% more floating point operations per iteration than ML-EM-3? The reason is due to the difference between equations (51), (52) and (55) in Table I. For a simultaneous update algorithm like ML-EM, the ratio $s_n = y_n/\bar{y}_n$ can be precomputed before computing the e_k terms, so there are only $m + N$ multiplies required, whereas for ML-SAGE, since the $\bar{y}_n$ terms are continually changing, the calculation of e_k using (55) requires $2m$ multiplications. There is an approach that can mitigate this 25% disadvantage; if one has enough dynamic memory to store both $\{a_{nk}\}$ as well as $\{q_{nk}\}$, where q_{nk} is precomputed as:

$$q_{nk} = a_{nk} y_n,$$

then for both ML-SAGE and ML-EM one can compute the e_k terms using

$$e_k = \sum_n q_{nk}/\bar{y}_n.$$

In this case the floating point computations of ML-SAGE and ML-EM are virtually identical, although the memory requirements of ML-SAGE will be roughly twice that of ML-EM.

For applications where it is currently impractical to precompute and store the a_{nk} factors, such as 3D PET or cone-beam SPECT, the above discussion is somewhat mute. In those applications, the extra m multiplications required by ML-SAGE will be inconsequential relative to the work required for on-line recalculation of the a_{nk}'s. In those cases, PML-SAGE-6 will be more favorable than is suggested by Table II since once one has expended the effort to compute the a_{nk}'s for the kth pixel in order to compute e_k using (55), one may as well also use those a_{nk}'s to compute $z_k(\boldsymbol{\lambda}^i)$ using (32).

VIII. Discussion

This report presents new algorithms (namely PML-SAGE-5 and PML-SAGE-6) for image reconstruction from Poisson measurements using a penalized likelihood objective function. The algorithms converge rapidly, monotonically, globally, and naturally enforce nonnegativity constraints. There are two main principles behind the new algorithms that lead to the improved convergence rates. The first principle is to update the pixel estimates sequentially rather than simultaneously. This idea has been used successfully by other authors as well [40,55,56]. The second principle is our use of new hidden data spaces that are less informative, formed by “mixing together” some of the emission events with the background events. Our results show that either of these ideas *by itself* leads to only small improvements in convergence rates (consider ML-EM-3 or ML-SAGE-4 relative to ML-EM-1), but the two principles applied in tandem (e.g. ML-SAGE-5 or ML-SAGE-6) lead to large improvements in convergence rates.

One very important issue that is beyond the scope of this report is the selection of the regularization parameter β. Qualitatively, increasing β leads to increased smoothness, similar to decreasing the cutoff frequency for conventional FBP reconstruction. Automatic methods for choosing smoothing parameters such as cross validation are one possibility, but such methods may be unstable in imaging problems [65]. We are currently investigating a frequency-domain method for relating the unitless parameter β to a quantitative measure of image resolution, so that one can choose appropriate values for β that yield consistent reconstructed resolution regardless of measurement variance.

We have attempted as fair of a comparison between SAGE methods and the alternatives as we think is possible. We presented slightly improved versions of several alternatives (GEM, OSL, De Pierro, etc.), and experimented with several choices for the design parameters m_k. Nevertheless, we cannot rule out the possibility that a better choice for $\{m_k\}$, or even a better choice for the complete-data space will be eventually found. Such an extension could be very useful since algorithms such as De Pierro's method have the advantage of being more suitable for fine-grain parallel computers than the SAGE algorithms we presented in this report. As described in Section II, the generic SAGE method offers more flexibility than we have used in this report. We are currently studying alternatives to PML-SAGE-5 that may be more suitable for fine-grain parallel computing (see Appendix III). Regardless however, in PET there is always the opportunity for coarse-grain parallel implementations with 100% processor utilization since contemporary PET systems produce dozens of slices.

By suitably *under*-relaxing the PML-CNR algorithm of Bouman and Sauer [55, 56], we were able to accelerate PML-CNR to the point where the convergence rates of PML-CNR and PML-SAGE-5,6 are comparable on the examples we have tried. The SAGE algorithms have the advantages of monotonicity and less computation time (for PML-SAGE-5). However, in principle it is intuitive to expect that since the PML-CNR method is based on a second-order approximation to the likelihood, in some situations it might converge faster locally to the maximum of the objective. We have not seen this yet, but the situation is confounded by the nonnegativity constraint inherent to Poisson problems; conventional wisdom about "supra-linear" convergence with Newton methods may not apply for coordinate-based methods with nonnegativity constraints. We believe further comparison of SAGE and PML-CNR is needed for a range of different phantoms and both quadratic and non-quadratic penalties. It may be that a hybrid algorithm is useful: monotonic PML-SAGE during the early iterations, and then greedy PML-CNR near the maximum. Another alternative to PML-SAGE that should be investigated is preconditioned conjugate gradient [57, 58], although such methods are trickier to implement due to the nonnegativity constraint.

Our results remind one that it is essential to use identical data sets when comparing the convergence rates of different algorithms. The convergence rate of PML-CNR appeared to be faster in [56], but for a rather different object and imaging system.

There is one subtle implementation issue that differs somewhat between SAGE and EM. The SAGE algorithm is optimized when the factors a_{nk} are precomputed and stored by *column* (n varying fastest). (If one uses on-line forward and backward projections, these should be pixel-driven for SAGE.) In contrast, the EM algorithm is indifferent to the storage organization, since the entire set of a_{nk} terms is used at once. Ironically, some EM implementations have been based on *row* storage (k varies fastest) due to historical use of row-action methods (e.g. ART). On Unix workstations, even dynamic memory size should not preclude use of precomputed a_{nk}'s, since one can often use the `mmap()` function to access the a_{nk}'s from disk faster than recomputing them on the fly.

We have compared several algorithms, and the reader may wonder what is the impact of these results on "practitioners" of penalized likelihood image reconstruction? In light of Fig. 9 and our experience with other experiments, we recommend using PML-SAGE-5 when processing Poisson measurements with a nonnegligible additive background (scatter, randoms, etc.) on conventional serial computers. For measurements with zero background, Figure 16 shows only a slight advantage for PML-SAGE-6 relative to PML-LINU-1, so we recommend that each user compare PML-SAGE-6 and PML-LINU-1 for her application.

In light of the considerable recent progress in improving the convergence rates of algorithms for maximum likelihood and penalized likelihood image reconstruction, it is highly unlikely that SAGE will be the final word. It is somewhat remarkable that the statistical principles behind the SAGE methods yield convergence rates that rival conventional numerical tools such as line-searches and Newton's methods, yet ensuring algorithm monotonicity. It seems likely that further development using statistical perspectives will lead to additional improvements.

IX. Appendix I: Convergence

The proof in [15] of local monotonic convergence in norm to a fixed point is inapplicable to problems with nonnegativity constraints, except when the fixed point happens to lie in the interior of the nonnegative orthant. In this appendix, we prove convergence of a very general form of SAGE that allows the limit to lie on the boundary of the nonnegative orthant. The proof structure is based on [17], with some aspects based on [39].

We begin by stating some general sufficient conditions for convergence. These conditions make no specific references to the Poisson likelihood or penalty used in this report, so this proof will apply to a broad class of nonnegatively constrained estimation problems. Following the general proof, we verify that the specific SAGE algorithms presented in this report meet the required conditions under the linear Poisson model.

Define the following sets:

$$\begin{aligned}\Re^+_S &= \{\boldsymbol{\theta}_S : \theta_k \geq 0,\ k \in S\},\\ \Theta^+ &= \{\boldsymbol{\theta} \in \Re^p : \theta_k \geq 0,\ k = 1, \ldots, p\},\\ \mathcal{S}(\boldsymbol{\theta}^0) &= \{\boldsymbol{\theta} \ :\ \Phi(\boldsymbol{\theta}) \geq \Phi(\boldsymbol{\theta}^0)\}.\end{aligned}$$

Also define:

$$\nabla^{10}_k \phi^i(\boldsymbol{\theta}^\star_{S^i}; \bar{\boldsymbol{\theta}}) \triangleq \frac{\partial}{\partial \theta_k} \phi^i(\boldsymbol{\theta}_{S^i}; \bar{\boldsymbol{\theta}}) \Big|_{\boldsymbol{\theta}_{S^i} = \boldsymbol{\theta}^\star_{S^i}}$$

and

$$\boldsymbol{J}^i(\boldsymbol{\theta}_{S^i}; \bar{\boldsymbol{\theta}}) \triangleq -\frac{1}{2} \nabla^{20} \phi^i(\boldsymbol{\theta}_{S^i}; \bar{\boldsymbol{\theta}}),$$

where

$$\left[\nabla^{20} \phi^i(\boldsymbol{\theta}_{S^i}; \bar{\boldsymbol{\theta}})\right]_{kj} = \frac{\partial^2}{\partial \theta_k \theta_j} \phi^i(\boldsymbol{\theta}_{S^i}; \bar{\boldsymbol{\theta}})$$

for $k, j \in S^i$.

To eliminate the interior restriction used in [15], we impose the following two regularity conditions on the objective.

Assumption 1: $\Phi(\boldsymbol{\theta})$ is strictly concave (and continuous and differentiable) on Θ^+.

Assumption 2: For any $\boldsymbol{\theta}^0 \in \Theta^+$, the set $\mathcal{S}(\boldsymbol{\theta}^0)$ is bounded.

As noted in [17], the assumption of strict concavity is adequate to "make up for" relaxing the restriction to the interior of Θ^+. We do not consider strict concavity to be an overly restrictive assumption; if Φ is not strictly concave, then typically either it does not have a unique maximum, in which case it is a questionable choice of objective, or it has local maxima, and no known deterministic algorithms are guaranteed to find the global maxima, including SAGE. Like any monotonic algorithm, for a non-strictly concave objective SAGE will only find a global maximum if initialized suitably close to one.

We assume the iterates are produced by an algorithm having the general form given in Section II, i.e., each iteration is associated with an index set S^i and a functional $\phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)$, and the iterates satisfy $\boldsymbol{\theta}^{i+1}_{\tilde{S}^i} = \boldsymbol{\theta}^i_{\tilde{S}^i}$. We assume that the functionals ϕ^i satisfy the following conditions.

Condition 1: The functionals ϕ^i satisfy (2), i.e.:

$$\Phi(\boldsymbol{\theta}_{S^i}, \boldsymbol{\theta}^i_{\tilde{S}^i}) - \Phi(\boldsymbol{\theta}^i) \geq \phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i) - \phi^i(\boldsymbol{\theta}^i_{S^i}; \boldsymbol{\theta}^i),$$

for $\boldsymbol{\theta}_{S^i} \in \Re^+_{S^i}$ and $\boldsymbol{\theta}^i \in \Theta^+$.

Condition 2: Each functional $\phi^i(\cdot; \boldsymbol{\theta})$ is strictly concave and twice differentiable on $\Re^+_{S^i}$ for any $\boldsymbol{\theta} \in \Theta^+$, and each $\phi^i(\cdot;\cdot)$ is continuous on $\Re^+_{S^i} \times \Theta^+$.

Condition 3: The following derivatives match $\forall i$:

$$\frac{\partial}{\partial \theta_k} \Phi(\boldsymbol{\theta}) = \nabla^{10}_k \phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta})$$

for any $\boldsymbol{\theta} \in \Theta^+$ and $k \in S^i$.

Condition 4: For $\boldsymbol{\theta}^i \in \Theta^+$, the iterates satisfy the Karush-Kuhn-Tucker conditions $\forall k \in S^i$:

$$\nabla^{10}_k \phi^i(\boldsymbol{\theta}^{i+1}_{S^i}; \boldsymbol{\theta}^i) \begin{cases} = 0, & \theta^{i+1}_k > 0 \\ \leq 0, & \theta^{i+1}_k = 0 \end{cases}.$$

Condition 5: For any bounded set $\mathcal{S}$, there exists a $C_{\mathcal{S}} > 0$ such that for every i, for all $\bar{\boldsymbol{\theta}} \in \mathcal{S}$, and for all $(\boldsymbol{\theta}_{S^i}, \bar{\boldsymbol{\theta}}_{\tilde{S}}) \in \mathcal{S}$:

$$\lambda_{\min} \left\{ \boldsymbol{J}^i(\boldsymbol{\theta}_{S^i}; \bar{\boldsymbol{\theta}}) \right\} \geq C_{\mathcal{S}},$$

where $\lambda_{\min}\{\boldsymbol{J}\}$ denotes the minimum eigenvalue of $\boldsymbol{J}$.

Condition 6: For each $k \in \{1, \ldots, p\}$, there is an index set $S^{(k)}$ containing k and functional $\phi^{(k)}$ that is used regularly to update the kth element of the parameter $\boldsymbol{\theta}$. Define $\mathcal{I}_k = \{i : S^i = S^{(k)} \text{ and } \phi^i = \phi^{(k)}\}$. Then for each k there exists an integer $i_{\max}$ (which may depend on k) such that

$$\forall n \geq 0 \; \exists i \in [n, n + i_{\max}] \text{ s.t. } i \in \mathcal{I}_k.$$

(This condition is clearly satisfied if the index sets and functionals are chosen periodically.)

Using the above Assumptions and Conditions, we can now prove a series of Lemmas that establish global convergence.

Lemma 1: The iterates $\{\boldsymbol{\theta}^i\}$ yield monotonic increases in $\Phi(\boldsymbol{\theta}^i)$, and are thus contained in the set $\mathcal{S}(\boldsymbol{\theta}^0)$. Furthermore, $\mathcal{S}(\boldsymbol{\theta}^0)$ is compact and convex.

Proof: Monotonicity follows from Conditions 1 and 4. Since Φ is strictly concave (Assumption 1), $\mathcal{S}(\boldsymbol{\theta}^0)$ is strictly convex. Since Φ is continuous (Assumption 1), $\mathcal{S}(\boldsymbol{\theta}^0)$ is closed [66, p. 91]. Thus $\mathcal{S}(\boldsymbol{\theta}^0)$ is compact since it is closed and bounded (Assumption 2), by the Heine-Borel theorem [66, p. 58]. □

Lemma 2: There exists a $C > 0$ such that for any i

$$\|\boldsymbol{\theta}^{i+1} - \boldsymbol{\theta}^i\|^2 \leq C^{-1}(\Phi(\boldsymbol{\theta}^{i+1}) - \Phi(\boldsymbol{\theta}^i)).$$

Proof: From Condition 1 and since $\boldsymbol{\theta}^{i+1}_{\tilde{S}^i} = \boldsymbol{\theta}^i_{\tilde{S}^i}$, it suffices to show $\forall i$:

$$\|\boldsymbol{\theta}^{i+1}_{S^i} - \boldsymbol{\theta}^i_{S^i}\|^2 \leq C^{-1}(\phi^i(\boldsymbol{\theta}^{i+1}_{S^i}; \boldsymbol{\theta}^i) - \phi^i(\boldsymbol{\theta}^i_{S^i}; \boldsymbol{\theta}^i)).$$

Expand $\phi^i(\cdot; \boldsymbol{\theta}^i)$ about $\boldsymbol{\theta}^{i+1}_{S^i}$ using Taylor's expansion with remainder [67, p. 599]:

$$\phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i) = \phi^i(\boldsymbol{\theta}^{i+1}_{S^i}; \boldsymbol{\theta}^i) +$$

$$\nabla^{10}\phi^i(\boldsymbol{\theta}^{i+1}_{S^i}; \boldsymbol{\theta}^i)(\boldsymbol{\theta}_{S^i} - \boldsymbol{\theta}^{i+1}_{S^i}) + (\boldsymbol{\theta}_{S^i} - \boldsymbol{\theta}^{i+1}_{S^i})'$$

$$\int_0^1 (1-t) \boldsymbol{J}^i((1-t)\boldsymbol{\theta}^{i+1}_{S^i} + t\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)\, dt\, (\boldsymbol{\theta}_{S^i} - \boldsymbol{\theta}^{i+1}_{S^i}). \quad (45)$$

From Condition 4, it follows that

$$\nabla^{10}\phi^i(\boldsymbol{\theta}^{i+1}_{S^i}; \boldsymbol{\theta}^i)(\boldsymbol{\theta}^{i+1}_{S^i} - \boldsymbol{\theta}^i_{S^i}) \geq 0,$$

so setting in $\boldsymbol{\theta}_{S^i} = \boldsymbol{\theta}^i_{S^i}$ in (45) yields

$$C\|\boldsymbol{\theta}^{i+1}_{S^i} - \boldsymbol{\theta}^i_{S^i}\|^2 \leq \phi^i(\boldsymbol{\theta}^{i+1}_{S^i}; \boldsymbol{\theta}^i) - \phi^i(\boldsymbol{\theta}^i_{S^i}; \boldsymbol{\theta}^i),$$

where $C = C_{\mathcal{S}(\boldsymbol{\theta}^0)}$. We have used Condition 5 and the fact that $\boldsymbol{x}'\boldsymbol{A}\boldsymbol{x} \geq \|\boldsymbol{x}\|^2 \lambda_{\min}\boldsymbol{A}$ for any positive definite matrix $\boldsymbol{A}$. □

Lemma 3:

$$\|\boldsymbol{\theta}^{i+1} - \boldsymbol{\theta}^i\| \to 0 \text{ as } i \to \infty.$$

Proof: Since $\{\Phi(\boldsymbol{\theta}^i)\}$ is monotone increasing (Lemma 1) and bounded above (by continuity of Φ and compactness (Lemma 1) of $\mathcal{S}(\boldsymbol{\theta}^0)$ [66, p. 78]), it follows that $\Phi(\boldsymbol{\theta}^{i+1}) - \Phi(\boldsymbol{\theta}^i) \to 0$. The Lemma then follows from Lemma 2. □

Lemma 4: The sequence $\{\boldsymbol{\theta}^i\}$ has a limit point[9] $\boldsymbol{\theta}^\star$. For any such limit point, if $\theta^\star_k > 0$, then

$$\frac{\partial}{\partial \theta_k} \Phi(\boldsymbol{\theta}^\star) = 0.$$

[9] The reader should note the distinction between limits and limit points (or cluster points) [66, p. 55].

Proof: By Lemma 1 and [66, p. 56], there is a subsequence i_m and limit point $\boldsymbol{\theta}^\star \in \mathcal{S}(\boldsymbol{\theta}^0)$ such that $||\boldsymbol{\theta}^{i_m} - \boldsymbol{\theta}^\star||^2 \to 0$ as $m \to \infty$. Now pick any index k, and define k_m to be the smallest $i \geq i_m$ such that $i \in \mathcal{I}_k$. By Condition 6, $k_m \leq i_m + i_{\max}$. By the triangle inequality:

$$||\boldsymbol{\theta}^{k_m} - \boldsymbol{\theta}^\star||^2 \leq ||\boldsymbol{\theta}^{k_m} - \boldsymbol{\theta}^{i_m}||^2 + ||\boldsymbol{\theta}^{i_m} - \boldsymbol{\theta}^\star||^2;$$

the second term of which goes to 0 as $m \to \infty$. For the first term, applying the triangle inequality repeatedly:

$$||\boldsymbol{\theta}^{k_m} - \boldsymbol{\theta}^{i_m}||^2 \leq \sum_{i=k_m}^{i_m-1} ||\boldsymbol{\theta}^{i+1} - \boldsymbol{\theta}^i||^2,$$

which is a sum of at most $i_{\max}$ terms by Condition 6, each of which goes to 0 as $m \to \infty$ by Lemma 3. Thus $||\boldsymbol{\theta}^{k_m} - \boldsymbol{\theta}^\star|| \to 0$ as $m \to \infty$. Again using the triangle inequality:

$$||\boldsymbol{\theta}^{k_m+1} - \boldsymbol{\theta}^\star||^2 \leq ||\boldsymbol{\theta}^{k_m+1} - \boldsymbol{\theta}^{k_m}||^2 + ||\boldsymbol{\theta}^{k_m} - \boldsymbol{\theta}^\star||^2.$$

Thus $||\boldsymbol{\theta}^{k_m+1} - \boldsymbol{\theta}^\star|| \to 0$ as $m \to \infty$.

Since $k_m \in S^{(k)}$, i.e. on iterations $\{k_m\}$ one updates θ_k, by Condition 4:

$$\theta_k^{k_m+1} \cdot \nabla_k^{10} \phi^{(k)}(\boldsymbol{\theta}_{S^{(k)}}^{k_m}; \boldsymbol{\theta}^{k_m}) = 0.$$

Taking the limit as $m \to \infty$ and using continuity (Condition 2) shows:

$$\theta_k^\star \cdot \nabla_k^{10} \phi^{(k)}(\boldsymbol{\theta}_{S^{(k)}}^\star; \boldsymbol{\theta}^\star) = 0.$$

The Lemma then follows from Condition 3. □

Lemma 5: The sequence $\{\boldsymbol{\theta}^i\}$ converges to a limit $\boldsymbol{\theta}^\infty$.
Proof: As in [17, Lemma 3], the number of limit points is finite (at most 2^p), due to Assumption 1, the nonnegativity constraint, and Lemma 4. However, since a bounded (Assumption 2) sequence $\{\boldsymbol{\theta}^i\}$ for which $||\boldsymbol{\theta}^{i+1} - \boldsymbol{\theta}^i|| \to 0$ (Lemma 3) has a connected and compact set of limit points [68, p. 173], there must be only one limit point. □

Lemma 6: The limit $\boldsymbol{\theta}^\infty$ satisfies the Karush-Kuhn-Tucker conditions for Φ.
Proof: For an element $\theta_k^\infty > 0$, we have $\partial/\partial\theta_k\Phi(\boldsymbol{\theta}^\infty) = 0$ by Lemma 4. Now suppose for some k we have $\theta_k = 0$ but $\partial/\partial\theta_k\Phi(\boldsymbol{\theta}^\infty) > 0$. Then by continuity (Assumption 1) and Lemma 3, $\partial/\partial\theta_k\Phi(\boldsymbol{\theta}^i) > 0$ for all i sufficiently large. Thus by Conditions 3 and 6,

$$\nabla_k^{10}\phi^{(k)}(\boldsymbol{\theta}_{S^{(k)}}^i; \boldsymbol{\theta}^i) > 0$$

for all $i \in \mathcal{I}_k$ sufficiently large. But since $\phi^{(k)}(\cdot; \boldsymbol{\theta}^i)$ is strictly concave (Condition 2), if $\nabla_k^{10}\phi^{(k)}(\boldsymbol{\theta}_{S^{(k)}}^i; \boldsymbol{\theta}^i) > 0$, then $\theta_k^{i+1} > \theta_k^i$. This contradicts $\theta_k^i \to 0$, so if $\theta_k^\infty = 0$ we must have $\partial/\partial\theta_k\Phi(\boldsymbol{\theta}^\infty) \leq 0$, establishing the Karush-Kuhn-Tucker conditions. □

Since a strictly concave objective has only one point that satisfies the Karush-Kuhn-Tucker conditions, namely the constrained maximum, the limit $\boldsymbol{\theta}^\infty$ must be that point. Lemma 6 thus establishes global convergence under a generic set of assumptions and conditions. All that remains is to verify that the conditions are satisfied for the SAGE algorithms presented in this report.
Remark:

In all of SAGE algorithms in this paper, the ϕ^i functionals are *additively separable* in their first argument, which means that the curvature matrices $\boldsymbol{J}^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)$ are diagonal. In this case, Condition 5 reduces to verifying that the diagonal elements of $\boldsymbol{J}^i$ have a positive lower bound. This is clearly the case for convex penalties such as the quadratic penalty (36). In other words, for separable ϕ^i functionals, a sufficient condition for Condition 5 is:
Condition 5′: *For any bounded set $\mathcal{S}$, there exists a $C_{\mathcal{S}} > 0$ such that for all $\boldsymbol{\theta} \in \mathcal{S}$*

$$-\frac{1}{2}\frac{\partial}{\partial\theta_k^2}P(\boldsymbol{\theta}) \geq C_{\mathcal{S}}.$$

Theorem 1: A sequence $\{\boldsymbol{\theta}^i\}$ generated by any of the PML-SAGE-4, PML-SAGE-5, or PML-SAGE-6 algorithms for penalized maximum-likelihood image reconstruction converges globally to the unique maximum of a strictly concave objective function Φ having a penalty function satisfying Condition 5′, provided $z_k > 0 \; \forall k$.
Proof:

- Assumption 2 follows from the behavior of the Poisson log-likelihood as $\lambda_k \to \infty$ [17].
- Condition 1 follows from [15, Theorem 1].
- Condition 2 is easily verified for the hidden-data spaces and penalty functions used in this report.
- Condition 3 follows by the construction of $\phi^{5,k}$ using (5)-(7).
- Condition 4 is built into the definition (3), and is satisfied by (38).
- Condition 5 follows from Condition 5' since the SAGE algorithms have separable ϕ^i functionals.
- Condition 6 is inherently satisfied by the cyclical sequential update used in PML-SAGE-5.

□

If one hopes for global convergence, then Condition 5′ is a reasonable restriction; it is clearly satisfied for the quadratic penalty (36), and for most strictly convex penalties.

There is an important technical difference between our proof and the assumptions in [17]. In [17] it was assumed that the sequence was initialized in the interior of Θ^+, and remained in the interior of Θ^+ for every iteration. With our new complete-data spaces and hidden-data spaces, the iterates can come and go from boundary of Θ^+ since the terms z_k are nonzero. However, when z_k is positive, one can verify that the corresponding functions $\phi^{5,k}$ are well-defined and differentiable on an open interval containing zero.

Condition 2 as stated is only met if $z_k > 0$ for all k, which will be true if $r_n > 0$ for all n. If one were to include the effects of say, cosmic radiation, then in practice it is always the case that $r_n > 0$. However, if some r_n, and hence some z_k are zero, it is simple to modify the proof to establish global convergence to the maximum. There is one important technical detail however; one cannot use $z_k > 0$ in one iteration and then switch to $z_k = 0$ in a later iteration, since then λ_k^i could get stuck on the boundary of Θ^+. Provided that one consistently uses either *only* the original complete-data space or *only* the new complete-data spaces, then global convergence is assured.

As stated above, the proof does not always apply to the unpenalized maximum-likelihood algorithms ML-EM-1, ML-EM-3, ML-SAGE-1, and ML-SAGE-5, because the curvature assumption Condition 5 is not necessarily satisfied without a strictly convex penalty. However, one can replace Condition 5 with an alternative condition that each $\phi^i(\boldsymbol{\theta}_{S^i}; \boldsymbol{\theta}^i)$ must be a monotonically decreasing function of θ_k. This approach was used in [12, 17]. With this condition, a small modification of the above proof establishes global convergence of the unpenalized algorithms, *provided that Assumption 1 is still satisfied*. This strict concavity will not be satisfied if the system matrix $\boldsymbol{A}$ does not have full column rank. We consider this to be a minor point since in the underdetermined case regularization is particularly essential, and the above proof shows that PML-SAGE-5 converges globally for strictly concave penalized maximum-likelihood objectives. We conjecture that the methods of [69, 70] could be extended to establish convergence of ML-EM-3, ML-SAGE-5, etc. without the strict concavity assumption, but such a proof would probably be of limited academic interest since in practice one rarely iterates a ML algorithm to convergence in the unregularized, underdetermined case.

If one is willing to be content with a local convergence result, then it is possible to relax the assumption of strict concavity for the ϕ^i functionals, using a region of convergence idea similar to that in [15, 16].

X. APPENDIX II: m_k DESIGN

The simple choice (23) for the design parameters $\{m_k\}$ for ML-EM-3 satisfies the constraints (21), but is not necessarily an optimal choice. In light of the form of the Fisher information (22) of $\boldsymbol{X}^3$, we would like the m_k's to be as "large as possible" subject to (21). One way to quantify this goal is the following weighted min-max criterion:

$$\max_{\boldsymbol{m} \in \mathcal{M}} \min_k \left\{ \frac{m_k}{w_k} \right\},$$

where

$$\mathcal{M} = \left\{ \boldsymbol{m} : \boldsymbol{m} \geq \boldsymbol{0}, \sum_k a_{nk} m_k \leq r_n \forall n \right\}.$$

The solution to this min-max problem is not unique in general, except for the smallest elements of $\boldsymbol{m}$ which one can easily show are given by

$$\begin{aligned} m_k &= w_k \rho_{n_1} \\ n_1 &= \arg\min_n \rho_n \\ \rho_n &= \frac{r_n}{\sum_k a_{nk} w_k}, \end{aligned} \tag{46}$$

where the minimization is only over the n for which the denominator is nonzero. For any nonnegative weights $\{w_k\}$, the design (46) clearly satisfies (21). Let

$$\mathcal{K}_n = \{k : a_{nk} \neq 0\}.$$

Then for $k \notin \mathcal{K}_{n_1}$, the corresponding m_k's are not restricted by the constraint

$$\sum_k a_{n_1 k} m_k \leq r_{n_1}.$$

so it is possible to further increase those m_k's. For $k \in \mathcal{K}_{n_1}$ we freeze the values of m_k to $m_k^\star = w_k \rho_{n_1}$, and then subtract from both sides of (21) to obtain the new constraint

$$\sum_{k \notin \mathcal{K}_{n_1}} a_{nk} m_k \leq r_n - \sum_{k \in \mathcal{K}_{n_1}} a_{nk} m_k^\star, \; n \neq n_1.$$

Defining

$$\rho_n' = \frac{r_n - \sum_{k \in \mathcal{K}_{n_1}} a_{nk} m_k^\star}{\sum_{k \notin \mathcal{K}_{n_1}} a_{nk} w_k}, \tag{47}$$

we can then let

$$n_2 = \arg\min_n \rho_n',$$

and assign $m_k^\star = w_k \rho_{n_2}'$ for $k \in \mathcal{K}_{n_2}$. This process can be repeated until every m_k is restricted by an active constraint, at which point the set $\{m_k^\star\}$ will satisfy (21) (easily shown by induction), and will in some sense be "as large as possible." However, in the few PET simulations we tried, the additional effort iterating to obtain $\{m_k^\star\}$ yielded only a very slight improvement over the simple design (23).

In light of (22), we have experimented with three choices for the weights: i) *uniform*: $w_k = 1$, ii) *sensitivity weighted*: $w_k = a_{\cdot k}$, and iii) *image weighted*: $w_k = 1/\max\{\hat{\lambda}_k, \delta\}$ for some small $\delta > 0$. For the few PET simulations we tried, all three performed nearly indistinguishably, so the results in Section V are simply based on uniform weights without iterating, i.e. (23). The basic problem is simply that in an EM-type algorithm with a *simultaneous* update, the background events $\{r_n\}$ must be spread out over all the pixels, so the values of m_k are fairly small relative to the pixel values λ_k. In contrast, in the *sequential* update of our ML-SAGE-5 method, many of the background events can be associated with pixel k as it is updated, so the values for z_k (see (31)) are orders of magnitude larger than the m_k's. We conclude that this difference largely explains the rapid convergence of ML-SAGE-5 relative to ML-EM-3 shown in Section V. However, it is possible that in other contexts a nonuniform weighting based on (47) would improve the convergence rate of ML-EM-3.

Considering (22) again, an alternative criterion for choosing the set $\{m_k\}$ is to ask that the largest element of $\boldsymbol{F}_{\mathbf{X}^3}$ be as small as possible, subject to *both* (21) *and* the constraint that $m_k \geq 0\ \forall k$. This is equivalent to:

$$\max_{\boldsymbol{m} \in \mathcal{M}} \min_k \frac{\hat{\lambda}_k + m_k}{a_{\cdot k}}.$$

Again, this min-max problem does not have a unique solution. However, we can give an algorithm for generating nonnegative m_k's that satisfy (21). First, by defining $x_k = m_k / a_{\cdot k}$, $\tau_k = \hat{\lambda}_k / a_{\cdot k}$, and $w_{nk} = a_{nk} a_{\cdot k}$, our problem is equivalent to:

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_k \{x_k + \tau_k\},$$

where

$$\mathcal{X} = \left\{ \boldsymbol{x} : \boldsymbol{x} \geq \mathbf{0}, \sum_k w_{nk} x_k \leq r_n \forall n \right\}. \tag{48}$$

Our algorithm for specifying the x_k's can be thought of as a "water-filling" algorithm where we initially let the solution have the form

$$x_k = [l - \tau_k]_+$$

for a level l, and then choose l as large as possible subject to (48). Only some of the x_k's will be restricted by whichever constraint in (48) is active, so we freeze those x_k values, and then continue to increase the water level l until another constraint in (48) becomes active. This process is repeated until all x_k's are restricted by an active constraint.

Formally then, define

$$l_1 = \max \left\{ l : \sum_k w_{nk} \left[l - \tau_k\right]_+ \leq r_n \ \forall n \right\},$$

and let $\mathcal{N}_1$ be the active constraints:

$$\mathcal{N}_1 = \left\{ n : \sum_k w_{nk} \left[l_1 - \tau_k\right]_+ = r_n \right\}.$$

Define

$$\mathcal{K}_1 = \{k : l_1 > \tau_k,\ w_{nk} \neq 0,\ n \in \mathcal{N}_1\},$$

and for $k \in \mathcal{K}_1$, fix

$$x_k^\star = [l_1 - \tau_k]_+ .$$

For the second iteration, define

$$l_2 = \max \left\{ l : \sum_{k \in \mathcal{K}_1} w_{nk} x_k^\star + \sum_{k \notin \mathcal{K}_1} w_{nk} \left[l - \tau_k\right]_+ \leq r_n \ \forall n \right\},$$

and let $\mathcal{N}_2$ be the active constraints for that maximization. For k in

$$\mathcal{K}_2 = \{k \notin \mathcal{K}_1 : l_2 > \tau_k,\ w_{nk} \neq 0,\ n \in \mathcal{N}_2\},$$

fix $x_k^\star = [l_2 - \tau_k]_+$. Repeat this process until all x_k's are restricted by an active constraint. One can show by induction that this process will satisfy the constraint (48). Furthermore, one can show that this construction yields a $\boldsymbol{x}^\star$ that has the following optimality property:

> For any other $\boldsymbol{x} \in \mathcal{X}$, if there is some k such that $x_k > x_k^\star$, then there must be some j such that $x_j < x_j^\star$, *and* $x_j + \tau_j \leq x_k + \tau_k$. In other words, one can only increase any of the elements of $\boldsymbol{x}^\star$ by decreasing some other element of $\boldsymbol{x}^\star$ whose sum $x_k^\star + \tau_k$ is smaller. Thus $\boldsymbol{x}^\star + \boldsymbol{\tau}$ is "as large as possible" in a strong sense.

Having found the x_k's using the above algorithm, one then computes the corresponding m_k's using $m_k = a_{\cdot k} x_k$. Finding the levels $l_1, l_2, \ldots$ is feasible but nontrivial computationally, so the performance of this "optimal" (in the sense of asymptotic convergence rate) set of m_k's has not been evaluated.

XI. Appendix III: Parallel SAGE

All of the preceding SAGE algorithms in this report were based on single pixel index sets. In this appendix, we sketch an approach for updating multiple pixels simultaneously. To simplify the presentation, we only present the unpenalized ML algorithms—the extension to PML is straightforward. The approach described below is very general (encompassing ML-SAGE-4,5,6 and ML-EM-1,3 as special cases), and can be made more amenable to parallel computing. It may also be possible to make it more computationally efficient than ML-SAGE-5,6.

First, split the set of image pixels into G *disjoint* groups $\mathcal{K}_1, \ldots, \mathcal{K}_G$ such that

$$\bigcup_{g=1}^{G} \mathcal{K}_g = \{1, \ldots, p\}.$$

For example, the "red-black" checkerboard groupings would correspond to $G = 2$, the ML-SAGE-4,5,6 algorithms would be $G = p$, and the ML-EM-1,3 algorithm would be $G = 1$. The SAGE algorithm alternates between updating the pixels in each group, i.e. $S^i = \mathcal{K}_g$, where $g = 1 + (i \operatorname{modulo} G)$. For updating the gth group, define the following hidden-data space:

$$\boldsymbol{X}^g = \left\{ \{Z_{nk}\}_{k \in \mathcal{K}_g}, \{B_{ng}\} \right\}_{n=1}^{N},$$

where

$$\begin{aligned} Z_{nk} &\sim \operatorname{Poisson}\{a_{nk}(\lambda_k + z_k)\},\ k \in \mathcal{K}_g, \\ B_{ng} &\sim \operatorname{Poisson}\{r_n + \sum_{k \notin \mathcal{K}_g} a_{nk}\lambda_k^i - \sum_{k \in \mathcal{K}_g} a_{nk} z_k\}. \end{aligned}$$

Note that

$$Y_n = \sum_{k \in \mathcal{K}_g} Z_{nk} + B_{ng}$$

has the proper distribution (11). The design parameters $\{z_k\}$ must satisfy:

$$\sum_{k \in \mathcal{K}_g} a_{nk} z_k \leq r_n + \sum_{k \notin \mathcal{K}_g} a_{nk} \lambda_k^i,\ \forall n, \qquad (49)$$

so that the Poisson rates of B_{ng} are nonnegative. The constraint (49) offers much more flexibility than both (21) and (30), and we have only begun to explore its potential. Perhaps the simplest approach to choosing the z_k's is to let $z_k = z_g(\boldsymbol{\lambda}^i)$ for $k \in \mathcal{K}_g$, where

$$z_g(\boldsymbol{\lambda}) = \min_n \left\{ \frac{r_n + \sum_{k \notin \mathcal{K}_g} a_{nk}\lambda_k}{\sum_{k \in \mathcal{K}_g} a_{nk}} \right\}. \qquad (50)$$

This choice is almost surely not opimal however.

The M-step for this complete-data space has the same form as (35), only all $k \in \mathcal{K}_g$ are updated simultaneously. In other words, the algorithm is a hybrid between Type I and Type III algorithms in Table I. Assuming the number of groups G is much less than p, then it will be more efficient to use (51) and (52) than (55), eliminating the 25% overhead for SAGE. However, one must perform the minimizations (50) each iteration (or find a better choice than (50)). We are currently exploring these tradeoffs.

XII. Acknowledgement

The authors gratefully acknowledge helpful discussions with Neal Clinthorne and W. Leslie Rogers, and Stephanie Sandor for bringing the `mmap()` function to their attention.

References

[1] L A Shepp and Y Vardi. Maximum likelihood reconstruction for emission tomography. *IEEE Tr. Med. Im.*, 1(2):113–122, October 1982.

[2] S Joshi and M I Miller. Maximum a posteriori estimation with Good's roughness for three-dimensional optical-sectioning microscopy. *J. Opt. Soc. Amer. Ser. A*, 10(5):1078–1085, May 1993.

[3] T J Holmes. Blind deconvolution of quantum-limited incoherent imagery: maximum-likelihood approach. *J. Opt. Soc. Amer. Ser. A*, 9(7):1052–1061, July 1992.

[4] D L Snyder, A M Hammoud, and R L White. Image recovery from data acquired with a charge-couple-device camera. *J. Opt. Soc. Amer. Ser. A*, 10(5):1014–1023, May 1993.

[5] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc. Ser. B*, 39(1):1–38, 1977.

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

[7] M E Daube-Witherspoon, R E Carson, Y Yan, and T K Yap. Scatter correction in maximum likelihood reconstruction of PET data. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 2, pages 945–947, 1992.

[8] J E Bowsher, D R Gilland, C E Floyd, R J Jaszczak, V E Johnson, and R E Coleman. Three-dimensional iterative reconstruction for SPECT. *J. Nuc. Med. (Abs. Book)*, 33(5):879, 1992.

[9] R Molina and B D Ripley. Deconvolution in optical astronomy. a Bayesian approach. In P Barone, A Frigessi, and M Piccioni, editors, *Stochastic Models, Statistical Methods, and Algorithms in Im. Analysis*, volume 74 of *Lecture Notes in Statistics*, pages 233–239. Springer, New York, 1992.

[10] K Sauer and C Bouman. A local update strategy for iterative reconstruction from projections. *IEEE Tr. Sig. Proc.*, 41(2):534–548, February 1993.

[11] J A Fessler. Hidden data spaces for maximum-likelihood PET reconstruction. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 2, pages 898–900, 1992.

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

[13] J A Fessler and A O Hero. Complete-data spaces and generalized EM algorithms. In *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, volume 4, pages 1–4, 1993.

[14] J A Fessler and A O Hero. New complete-data spaces and faster algorithms for penalized-likelihood emission tomography. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 3, pages 1897–1901, 1993.

[15] J A Fessler and A O Hero. Space-alternating generalized expectation-maximization algorithm. *IEEE Tr. Sig. Proc.*, 42(10):2664–77, October 1994.

[16] A O Hero and J A Fessler. Asymptotic convergence properties of EM-type algorithms. Technical Report 282, Comm. and Sign. Proc. Lab., Dept. of EECS, Univ. of Michigan, Ann Arbor, MI, 48109-2122, April 1993.

[17] K Lange and R Carson. EM reconstruction algorithms for emission and transmission tomography. *J. Comp. Assisted Tomo.*, 8(2):306–316, April 1984.

[18] E Veklerov and J Llacer. Stopping rule for the MLE algorithm based on statistical hypothesis testing. *IEEE Tr. Med. Im.*, 6(4):313–319, December 1987.

[19] J A Fessler. Penalized weighted least-squares image reconstruction for positron emission tomography. *IEEE Tr. Med. Im.*, 13(2):290–300, June 1994.

[20] D S Lalush and B M W Tsui. A fast and stable weighted-least squares MAP conjugate-gradient algorithm for SPECT. *J. Nuc. Med. (Abs. Book)*, 34(5):27, May 1993.

[21] K Lange, M Bahn, and R Little. A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *IEEE Tr. Med. Im.*, 6(2):106–114, June 1987.

[22] E M Levitan and G T Herman. A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography. *IEEE Tr. Med. Im.*, 6(3):185–192, September 1987.

[23] J Nunez and J Llacer. A fast Bayesian reconstruction algorithm for emission tomography with entropy prior converging to feasible images. *IEEE Tr. Med. Im.*, 9(2):159–171, June 1990.

[24] B W Silverman, M C Jones, J D Wilson, and D W Nychka. A smoothed EM approach to indirect estimation problems, with particular reference to stereology and emission tomography. *J. Royal Stat. Soc. Ser. B*, 52(2):271–324, 1990.

[25] D Nychka. Some properties of adding a smoothing step to the EM algorithm. *Statistics and Probability Letters*, 9:187–193, February 1990.

[26] Z Liang, R Jaszczak, C Floyd, and K Greer. A spatial interaction model for statistical image processing. In C N de Graaf and M A Viergever, editors, *Proc. Twelfth Intl. Conf. on Information Processing in Medical Im.*, pages 29–43. Plenum Press, New York, 1991.

[27] D L Snyder and M I Miller. The use of sieves to stabilize images produced with the EM algorithm for emission tomography. *IEEE Tr. Nuc. Sci.*, 32(5):3864–3871, October 1985.

[28] D L Snyder, M I Miller, L J Thomas, and D G Politte. Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography. *IEEE Tr. Med. Im.*, 6(3):228–238, September 1987.

[29] C S Butler and M I Miller. Maximum a Posteriori estimation for SPECT using regularization techniques on massively parallel computers. *IEEE Tr. Med. Im.*, 12(1):84–89, March 1993.

[30] J A Fessler, N H Clinthorne, and W L Rogers. Regularized emission image reconstruction using imperfect side information. In *Proc. IEEE Nuc. Sci. Symp. Med. Im. Conf.*, volume 3, pages 1991–1995, 1991.

[31] T Hebert and R Leahy. A Bayesian reconstruction algorithm for emission tomography using a Markov random field prior. In *Proc. SPIE 1092, Med. Im. III: Im. Proc.*, pages 458–4662, 1989.

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

[33] T J Hebert and R Leahy. Statistic-based MAP image reconstruction from Poisson data using Gibbs priors. *IEEE Tr. Sig. Proc.*, 40(9):2290–2303, September 1992.

[34] G Gindi, A Rangarajan, M Lee, P J Hong, and I G Zubal. Bayesian reconstruction for emission tomography via deterministic annealing. In H H Barrett and A F Gmitro, editors, *Information Processing in Medical Im.*, volume 687 of *Lecture Notes in Computer Science*, pages 322–338. Springer Verlag, Berlin, 1993.

[35] X L Meng and D B Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[36] C H Liu and D B Rubin. The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4):633–48, 1994.

[37] A R De Pierro. A generalization of the EM algorithm for maximum likelihood estimates from incomplete data. Technical Report MIPG119, Med. Im. Proc. Group, Dept. of Radiol., Univ. of Pennsylvania, February 1987.

[38] G T Herman, A R De Pierro, and N Gai. On methods for maximum a posteriori image reconstruction with a normal prior. *J. Visual Comm. Im. Rep.*, 3(4):316–324, December 1992.

[39] A R De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Tr. Med. Im.*, 14(1):132–137, March 1995.

[40] M Abdalla and J W Kay. Edge-preserving image restoration. In P Barone, A Frigessi, and M Piccioni, editors, *Stochastic Models, Statistical Methods, and Algorithms in Im. Analysis*, volume 74 of *Lecture Notes in Statistics*, pages 1–13. Springer, New York, 1992.

[41] S Geman and D E McClure. Bayesian image analysis: an application to single photon emission tomography. In *Proc. of Stat. Comp. Sect. of Amer. Stat. Assoc.*, pages 12–18, 1985.

[42] S Geman and D E McClure. Statistical methods for tomographic image reconstruction. *Proc. 46 Sect. ISI, Bull. ISI*, 52:5–21, 1987.

[43] P J Green. Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Tr. Med. Im.*, 9(1):84–93, March 1990.

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

[45] P J Green. Statistical methods for spatial image analysis (Discussion). *Proc. 46 Sect. ISI, Bull. ISI*, 52:43–44, 1987.

[46] K Lange. Convergence of EM image reconstruction algorithms with Gibbs smoothing. *IEEE Tr. Med. Im.*, 9(4):439–446, December 1990. Corrections, June 1991.

[47] H Hart and Z Liang. Bayesian image processing in two dimensions. *IEEE Tr. Med. Im.*, 6(3):201–208, September 1987.

[48] Z Liang, R Jaszczak, and K Greer. On Bayesian image reconstruction from projections: uniform and nonuniform a priori source information. *IEEE Tr. Med. Im.*, 8(3):227–235, September 1989.

[49] G T Herman and D Odhner. Performance evaluation of an iterative image reconstruction algorithm for positron-emission tomography. *IEEE Tr. Med. Im.*, 10(3):336–346, September 1991.

[50] G T Herman, D Odhner, K D Toennies, and S A Zenios. A parallelized algorithm for image reconstruction from noisy projections. In T F Coleman and Y Li, editors, *Large-Scale Numerical Optimization*, pages 3–21. SIAM, Philadelphia, 1990.

[51] A W McCarthy and M I Miller. Maximum likelihood SPECT in clinical computation times using mesh-connected parallel processors. *IEEE Tr. Med. Im.*, 10(3):426–436, September 1991.

[52] M I Miller and B Roysam. Bayesian image reconstruction for emission

tomography incorporating Good's roughness prior on massively parallel processors. *Proc. Natl. Acad. Sci.*, 88:3223–3227, April 1991.

[53] A D Lantermann. A new way to regularize maximum-likelihood estimates for emission tomography with Good's roughness penalty. Technical Report ESSRL-92-05, Elec. Sys. and Sig. Res. Lab., Washington Univ., January 1992.

[54] D L Snyder, A D Lanterman, and M I Miller. Regularizing images in emission tomography via an extension of Good's roughness measure. Technical Report ESSRL-92-17, Washington Univ., January 1992.

[55] C Bouman and K Sauer. Fast numerical methods for emission and transmission tomographic reconstruction. In *Proc. 27th Conf. Info. Sci. Sys., Johns Hopkins*, pages 611–616, 1993.

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

[57] E U Mumcuoglu, R Leahy, S R Cherry, and Z Zhou. Fast gradient-based methods for Bayesian reconstruction of transmission and emission PET images. Technical Report 241, Signal and Image Processing Inst., USC, September 1993.

[58] L Kaufman. Implementing and accelerating the EM algorithm for positron emission tomography. *IEEE Tr. Med. Im.*, 6(1):37–51, March 1987.

[59] A R De Pierro. On the relation between the ISRA and the EM algorithm for positron emission tomography. *IEEE Tr. Med. Im.*, 12(2):328–333, June 1993.

[60] W H Press, B P Flannery, S A Teukolsky, and W T Vetterling. *Numerical recipes in C*. Cambridge Univ. Press, 1988.

[61] A O Hero and J A Fessler. A recursive algorithm for computing Cramer-Rao-type bounds on estimator covariance. *IEEE Tr. Info. Theory*, 40(4):1205–10, July 1994.

[62] C T Chen, V E Johnson, W H Wong, X Hu, and C E Metz. Bayesian image reconstruction in positron emission tomography. *IEEE Tr. Nuc. Sci.*, 37(2):636–641, April 1990.

[63] R Leahy and X H Yan. Statistical models and methods for PET image reconstruction. In *Proc. of Stat. Comp. Sect. of Amer. Stat. Assoc.*, pages 1–10, 1991.

[64] X H Yan and R Leahy. MAP image reconstruction using intensity and line processes for emission tomography data. In *Proc. SPIE 1452, Im. Proc. Alg. and Tech. II*, 1991.

[65] J W Hilgers and W R Reynolds. Instabilities in the optimal regularization parameter relating to image recovery problems. *J. Opt. Soc. Amer. Ser. A*, 9(8):1273–1279, August 1992.

[66] M Rosenlicht. *Introduction to analysis*. Dover, New York, 1985.

[67] R E Williamson, R H Crowell, and H F Trotter. *Calculus of vector functions*. Prentice Hall, New Jersey, 1972.

[68] A M Ostrowski. *Solution of equations in Euclidian and Banach spaces*. Academic Press, 1973.

[69] Y Vardi, L A Shepp, and L Kaufman. A statistical model for positron emission tomography. *J. Am. Stat. Ass.*, 80(389):8–37, March 1985.

[70] T M Cover. An algorithm for maximizing expected log investment return. *IEEE Tr. Info. Theory*, 30(2):369–373, March 1984.

Type-I Algorithm (e.g. ML-EM, PML-OSL)
and
Type-II Algorithm (e.g. PML-GEM)

```
Initialize $\boldsymbol{\lambda}^0$
    for $i = 0, 1, \ldots \{$
            $\bar{y}_n = \sum_k a_{nk}\lambda_k^i + r_n,\ n = 1, \ldots, N$
            $s_n = y_n/\bar{y}_n,\ n = 1, \ldots, N$                                  (51)
            $e_k = \sum_n a_{nk}s_n,\ k = 1, \ldots, p$                                (52)
        for $k = 1, \ldots, p \{$
            $\lambda_k^{i+1} = g_k(e_k; \boldsymbol{\lambda}^i)$,          (Type-I), or          (53)
            $\lambda_k^{i+1} = g_k(e_k; \boldsymbol{\lambda}^\star; \boldsymbol{\lambda}^i)$,     (Type-II)             (54)
        $\}$
    $\}$
```

Type-III Algorithm (e.g. ML-SAGE, PML-SAGE)

```
Initialize $\boldsymbol{\lambda}^0$,          $\bar{y}_n = \sum_k a_{nk}\lambda_k^0 + r_n,\ n = 1, \ldots, N.$
    for $i = 0, 1, \ldots \{$
            $k = 1 + (i \text{ modulo } p)$
            $e_k = \sum_n a_{nk}y_n/\bar{y}_n$                                          (55)
            $\lambda_k^{i+1} = g_k(e_k; \boldsymbol{\lambda}^i)$                                     (56)
            $\lambda_j^{i+1} = \lambda_j^i,\ j \neq k,$
            $\bar{y}_n := \bar{y}_n + (\lambda_k^{i+1} - \lambda_k^i)a_{nk},\ \forall n : a_{nk} \neq 0$            (57)
    $\}$
```

TABLE I

Three generic pseudo-code algorithm types for penalized maximum-likelihood image reconstruction. All of the algorithms presented in the text are of one of these three types. Within each type, the algorithms differ in form of the functions $g()$ used in the M-step.

	CPU Seconds							
	% Background			Floating Point Operations				
	0%	5%	35%	multiply	add	$x > 0$?	$x > y$?	log, $\sqrt{}$
ML-EM-1	25	25	25	$2m + 2p + N$	$2m - p$	N		
ML-EM-3		24	24	$2m + 2p + N$	$2m + p$	$N + p$		
ML-LINB-1	26	26	26	$2m + 3p + 4N$	$2m + 5N$	$2N$		$2N$
ML-LINU-1[10]	34	26	27	$4m + 4p + 9N$				
ML-SAGE-4	45	37	37	$3m + 2p$	$2m$	m		
ML-SAGE-5		36	35	$3m + 2p$	$2m + 2p$	$m + p$		
ML-SAGE-6	67	64	64	$4m + 2p$	$2m + 3p$	$m + p$	$m - p$	
ML-CNR				$4m + 2p$	$3m + 3p$	$m + p$		
WLS+SOR [19]				$3m + 2p$	$2m + p$	p		
PML-OSL-1	26	26	37	$2m + 5p + N$	$2m + 8p$	N		
PML-OSL-3		25	35	$2m + 5p + N$	$2m + 10p$	$N + 2p$		
PML-GEM-1	26	26	38	$2m + 9p + N$	$2m + 9p$	N		p
PML-GEM-3		26	37	$2m + 10p + N$	$2m + 12p$	$N + p$		p
PML-LINB-1	29	29	40	$2m + 10p + 4N$	$2m + 18p + 5N$	$2N$		$2N$
PML-SAGE-4	42	39	68	$3m + 9p$	$2m + 8p$	m		p
PML-SAGE-5		40	68	$3m + 10p$	$2m + 13p$	$m + p$		p
PML-SAGE-6	42	41	114	$3m + 10p$	$2m + 13p$	$m + p$	$m - p$	p
PML-CNR	42	40	85	$?m + 5p$	$3m + 12p$	$m + p$		

TABLE II

CPU seconds for 40 iterations of each algorithm. The number of nonzero a_{nk}'s is denoted m; typically $m << pN$. Since $m >> N$ and $m >> p$, the terms involving m dominate. Most of the floating point comparisons with 0 are unnecessary when $r_n > 0$, since then $\bar{y}_n > 0$, hence the faster execution time of PML-SAGE-5 for 5% and 35% background. The execution times for the -CNR and -OSL algorithms are *without* checking for monotonicity; all the other algorithms are monotonic.

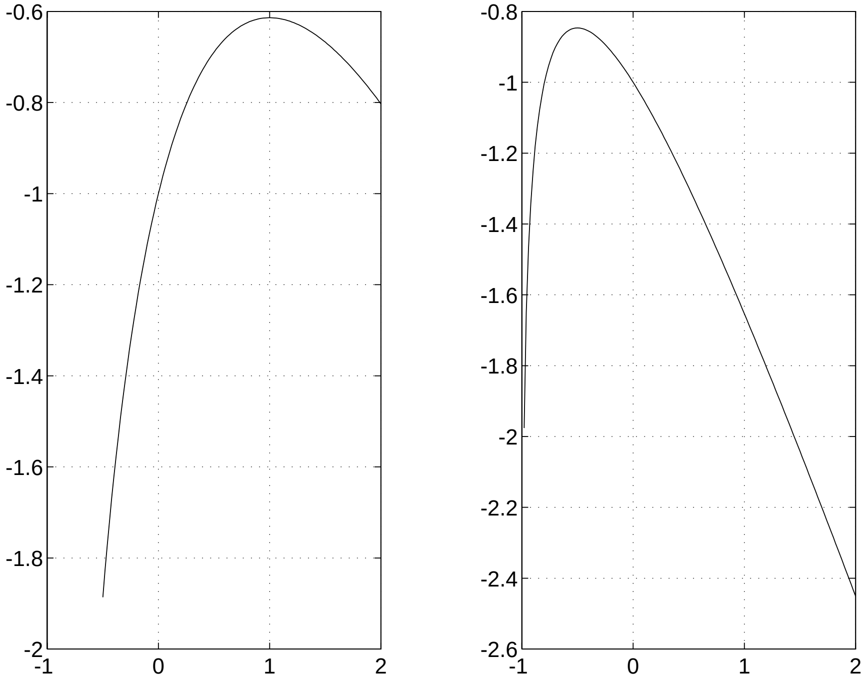

Fig. 1. Typical plots of $Q_{\mathbf{X}^3}(\boldsymbol{\lambda};\boldsymbol{\lambda}^i)$ versus λ_k. In the left plot, the maximum occurs for $\lambda_k > 0$. For the right plot, although the unconstrained maximum occurs for a negative λ_k, the nonnegatively constrained maximum is at $\lambda_k = 0$, due to the concavity of $Q_{\mathbf{X}^3}$.

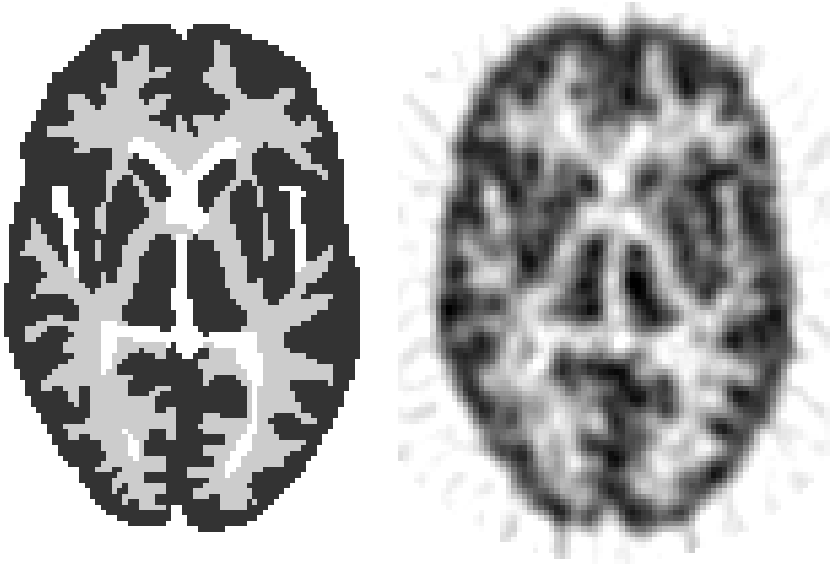

Fig. 2. Digital brain phantom (left), and filtered backprojection reconstructed image (right).

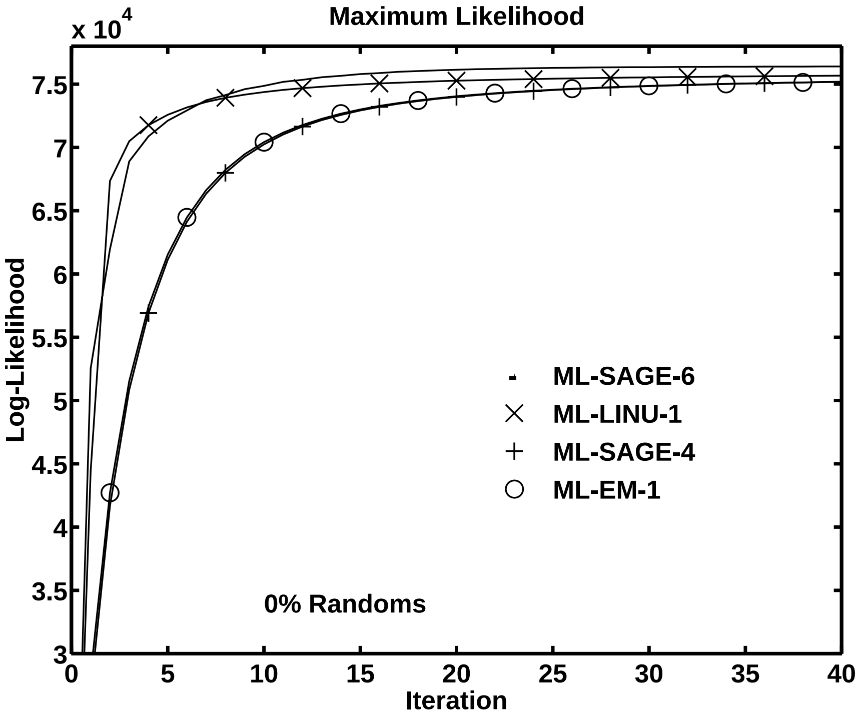

Fig. 3. Log-likelihood $L(\boldsymbol{\lambda}^i) - L(\boldsymbol{\lambda}^0)$ vs. iteration for unpenalized maximum-likelihood reconstruction from data with 0% random coincidences.

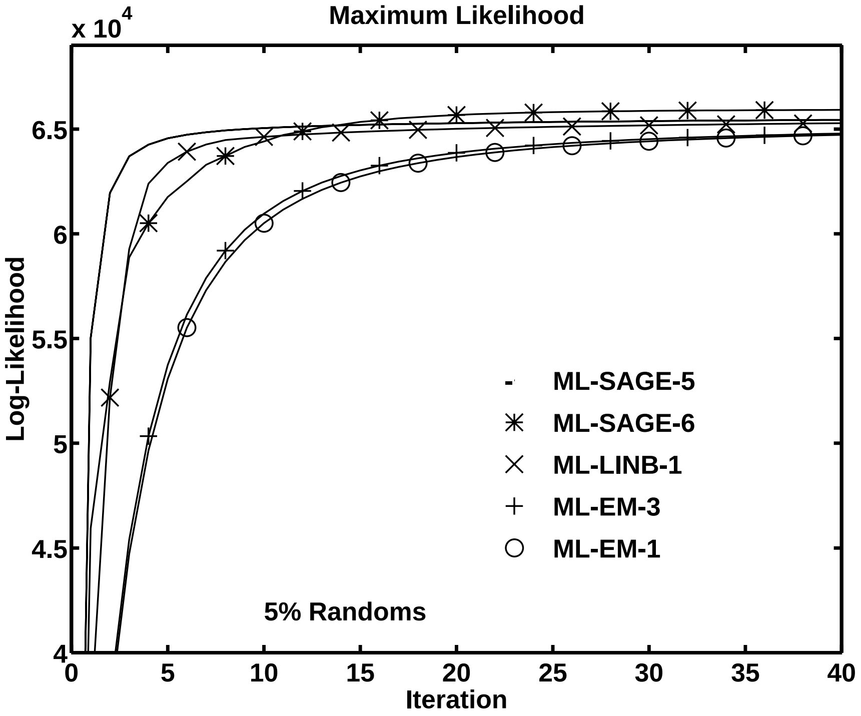

Fig. 4. Log-likelihood $L(\boldsymbol{\lambda}^i) - L(\boldsymbol{\lambda}^0)$ vs. iteration for unpenalized maximum-likelihood reconstruction from data with 5% random coincidences. Not shown is ML-SAGE-4, which is indistinguishable from ML-EM-1.

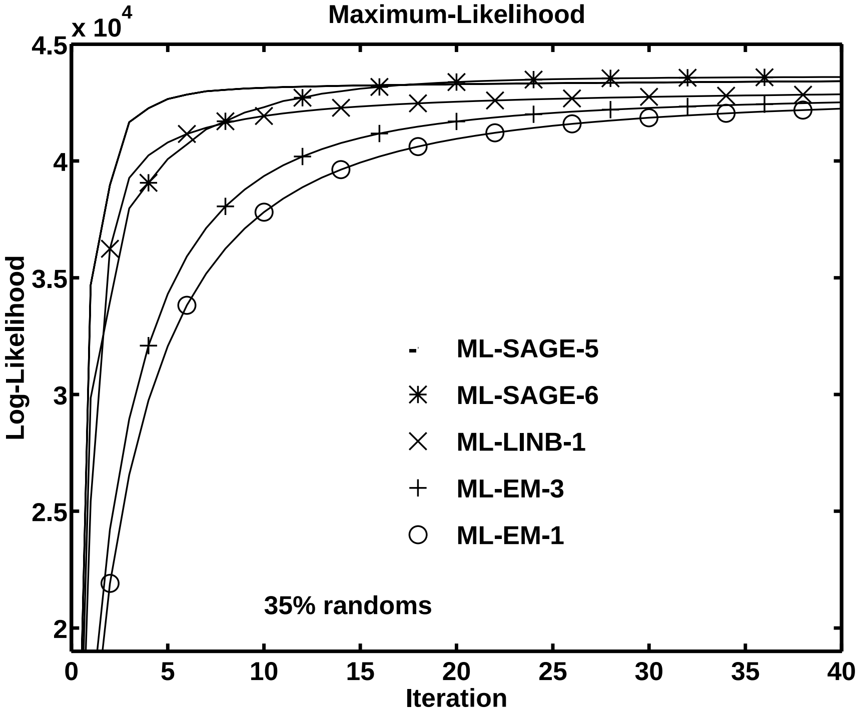

Fig. 5. Log-likelihood $L(\boldsymbol{\lambda}^i) - L(\boldsymbol{\lambda}^0)$ vs. iteration for unpenalized maximum-likelihood reconstruction from data with 35% random coincidences. Not shown is ML-SAGE-4, which is indistinguishable from ML-EM-1.

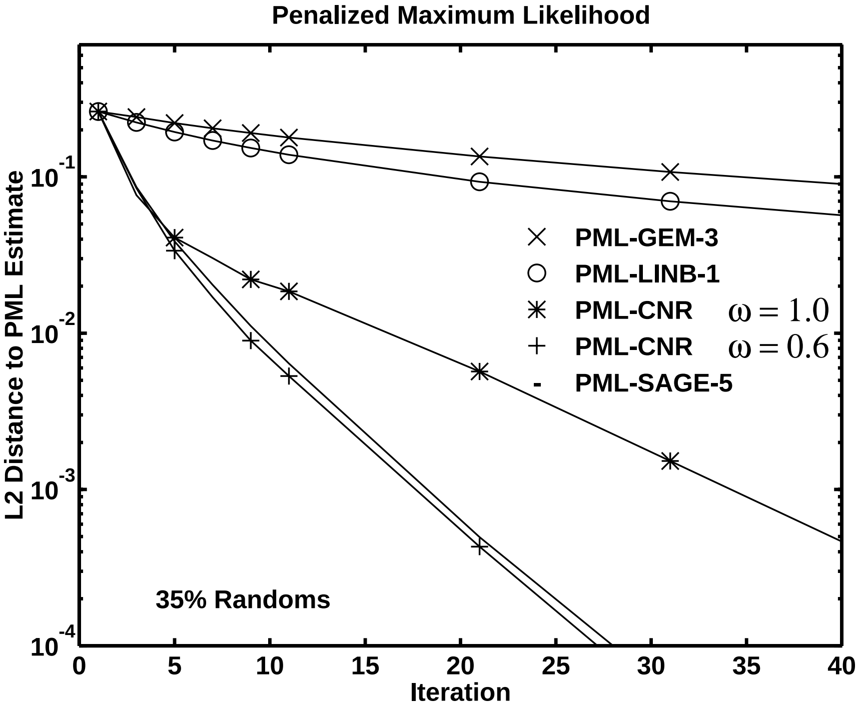

Fig. 6. L_2 distance $||\boldsymbol{\lambda}^i - \hat{\boldsymbol{\lambda}}||$ vs. iteration from data with 35% random coincidences. Not shown is PML-CNR ($\omega = 0.6$), which is indistinguishable from PML-SAGE-5.

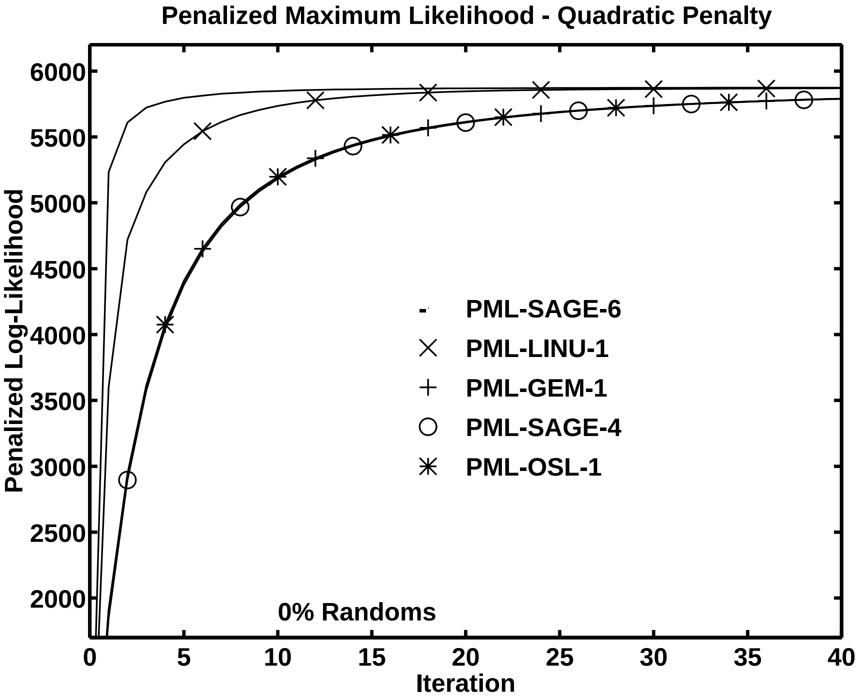

Fig. 7. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. iteration from data with 0% random coincidences.

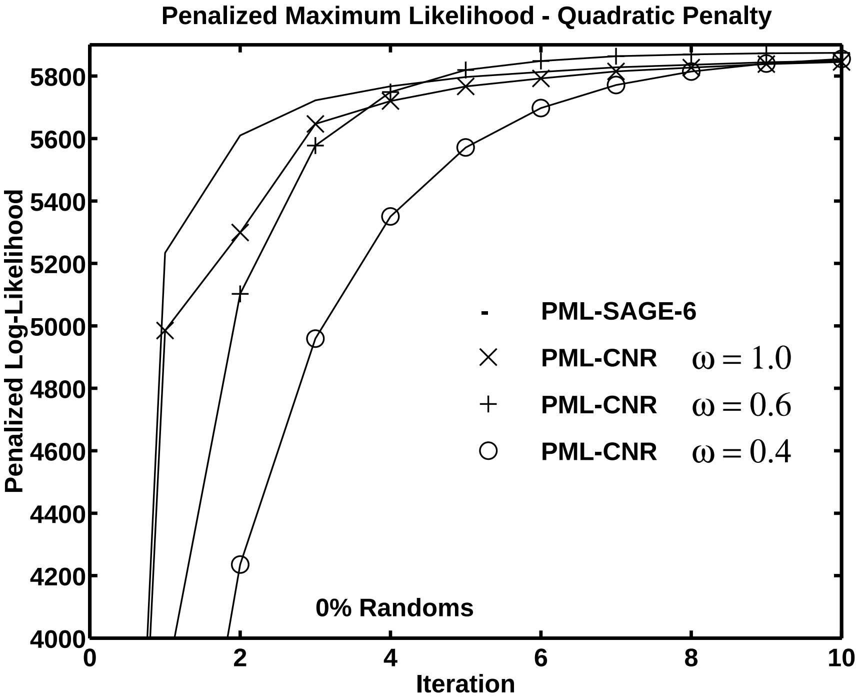

Fig. 8. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. iteration from data with 0% random coincidences.

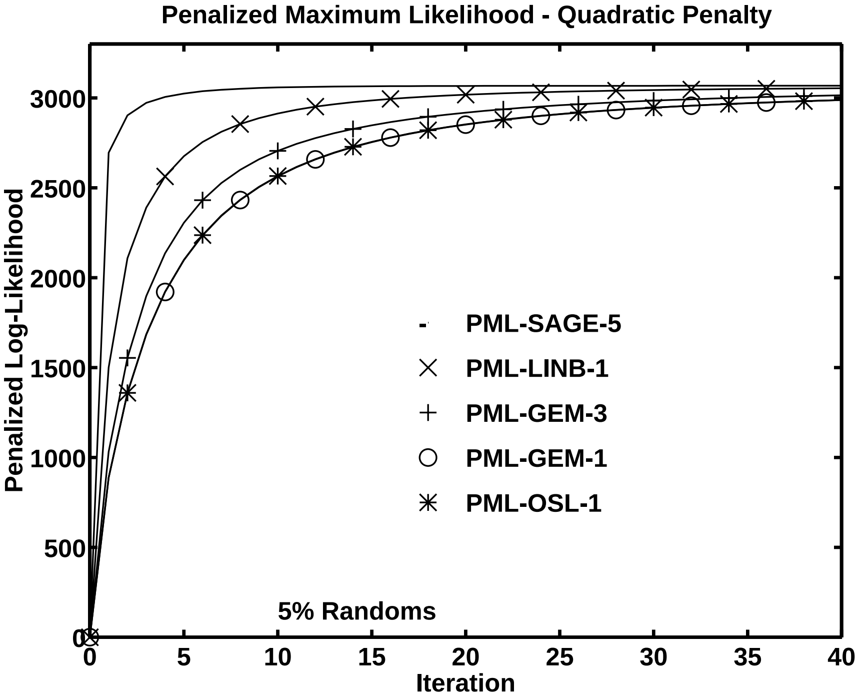

Fig. 9. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. iteration from data with 5% random coincidences. Not shown is PML-SAGE-6, which is indistinguishable from PML-SAGE-5. Also not shown is PML-SAGE-4, which is indistinguishable from PML-OSL-1.

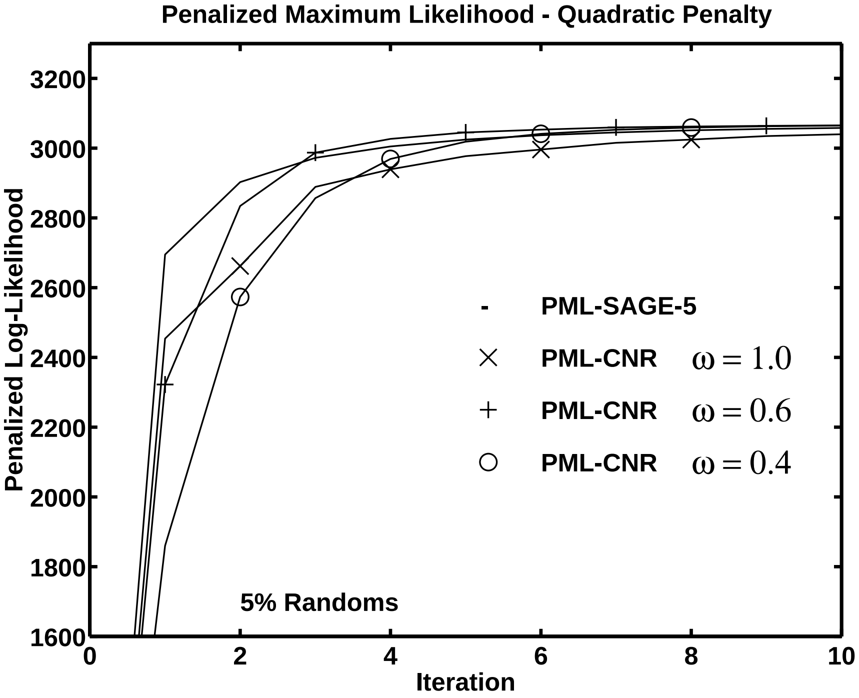

Fig. 10. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. iteration from data with 5% random coincidences.

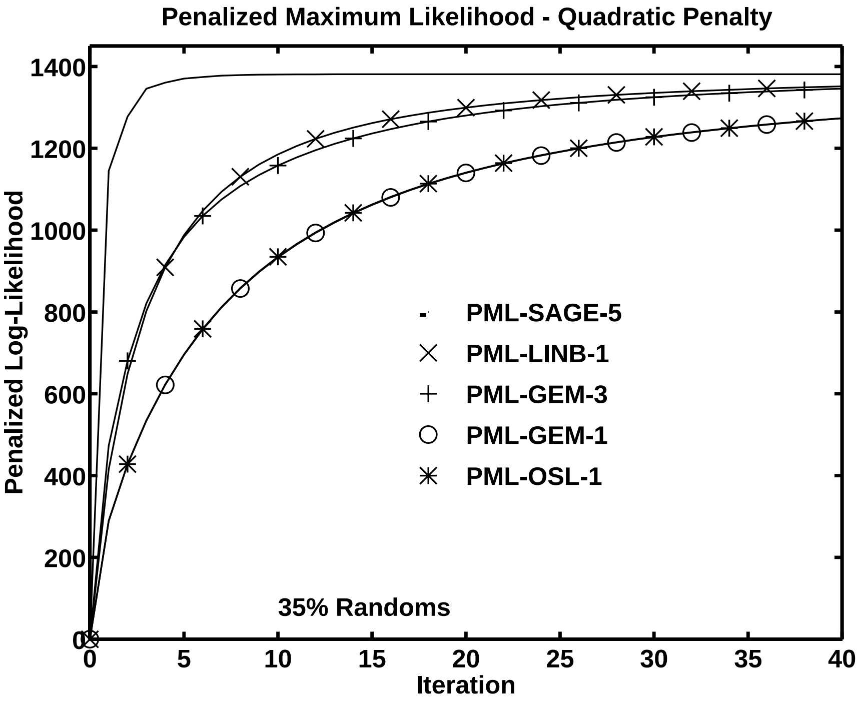

Fig. 11. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. iteration from data with 35% random coincidences. Not shown is PML-SAGE-6, which is indistinguishable from PML-SAGE-5. Also not shown is PML-SAGE-4, which is indistinguishable from PML-OSL-1.

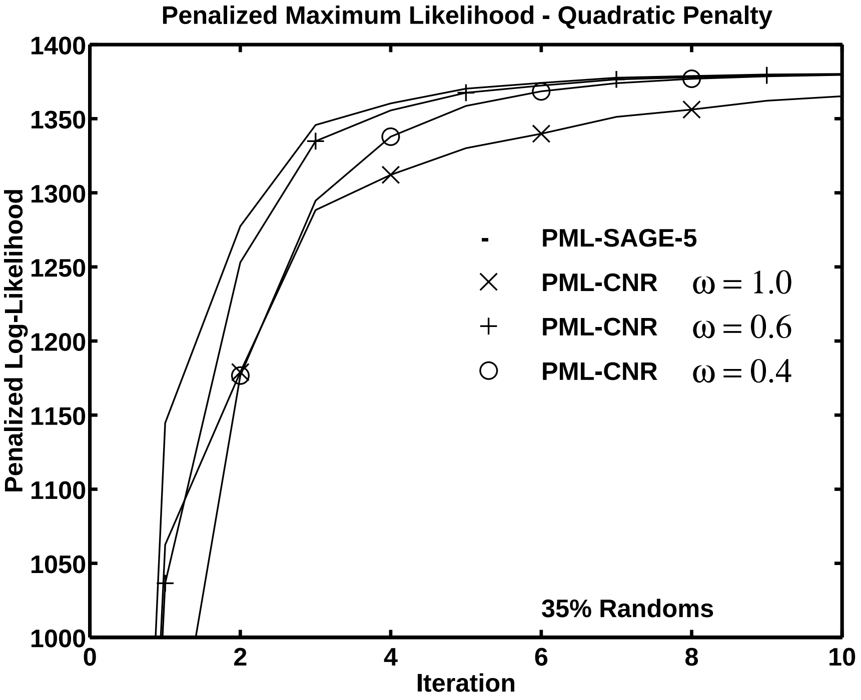

Fig. 12. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. iteration from data with 35% random coincidences.

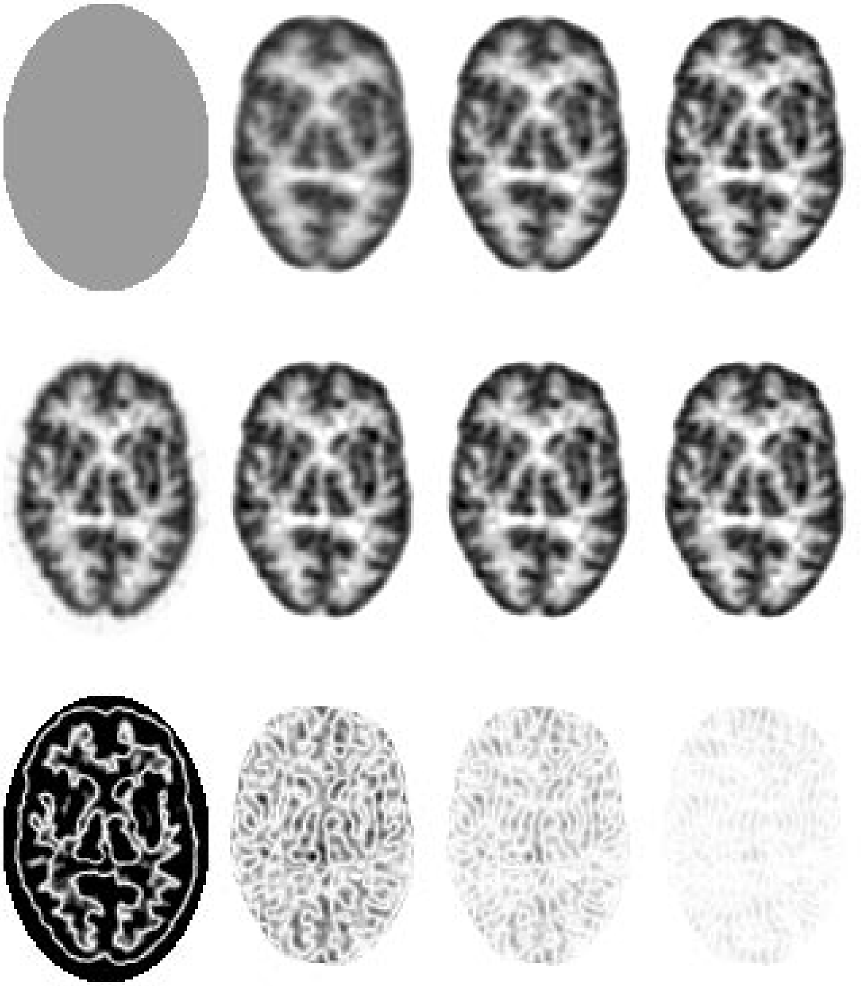

Fig. 13. PML-SAGE-5 estimates from data with 5% random coincidences at iterations $i = 0, 5, 10, 20$ (left to right). Top row: initialized with uniform image. Middle row: initialized with thresholded filtered-backprojection image. Bottom row: absolute value of difference between top and middle rows amplified by a factor of 4.

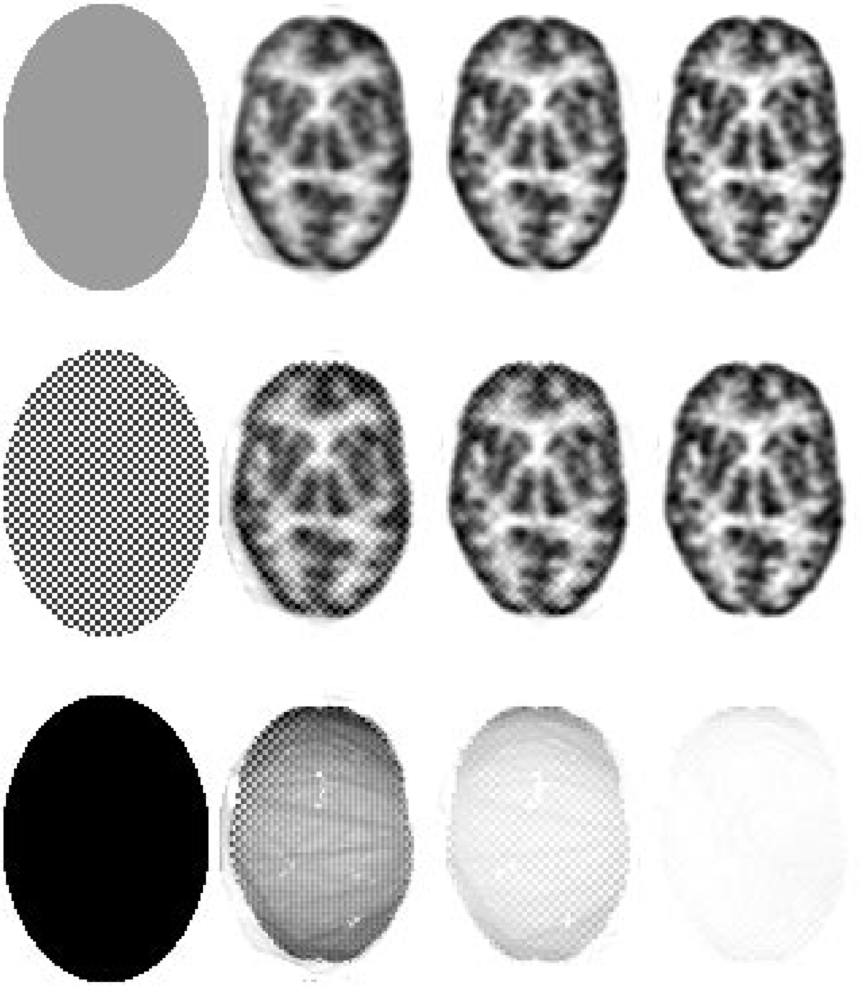

Fig. 14. PML-SAGE-5 estimates from data with 35% random coincidences at iterations $i = 0, 2, 4, 8$ (left to right). Top row: initialized with uniform image. Middle row: initialized with "checkerboard" image alternating between intensities 0 and 4. Bottom row: absolute value of difference between top and middle rows amplified by a factor of 4.

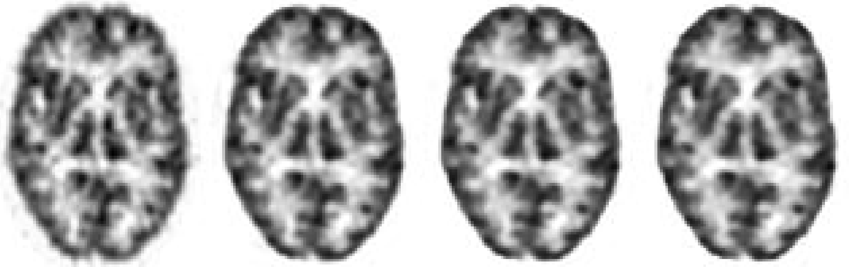

Fig. 15. PML-SAGE-5 estimates from data with 35% random coincidences at iterations $i = 0, 2, 4, 10$ (left to right) for a penalized maximum likelihood objective with a nonquadratic "edge-preserving" penalty (see text). The iterates produced by the SAGE method stabilize rapidly even for nonquadratic penalties.

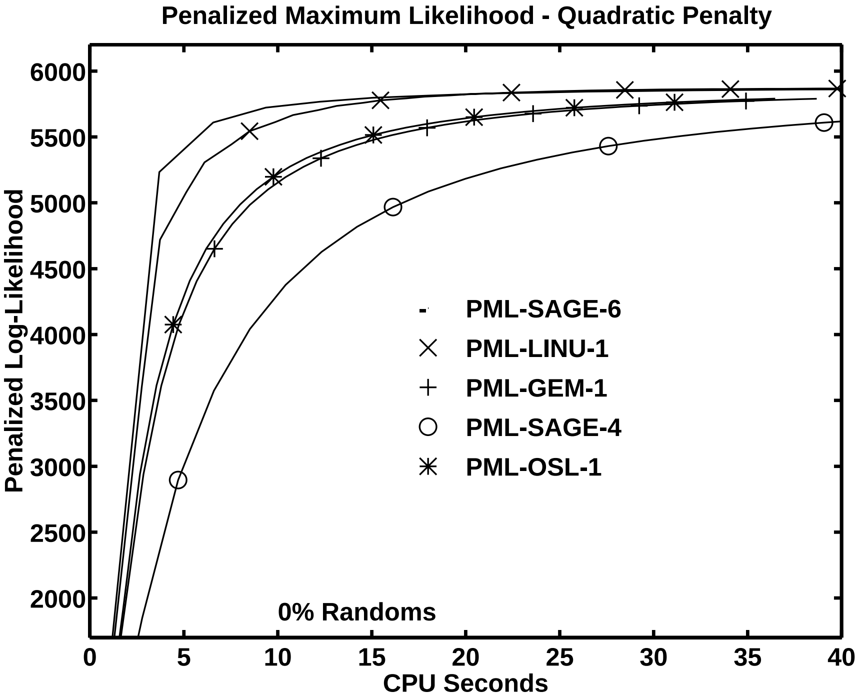

Fig. 16. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. CPU time from data with 0% random coincidences.

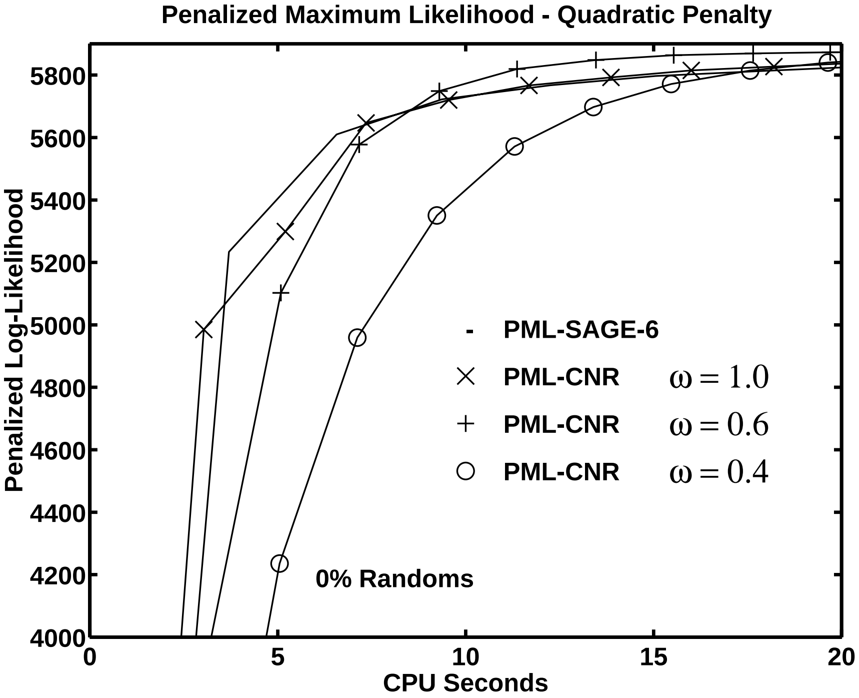

Fig. 17. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. CPU time from data with 0% random coincidences.

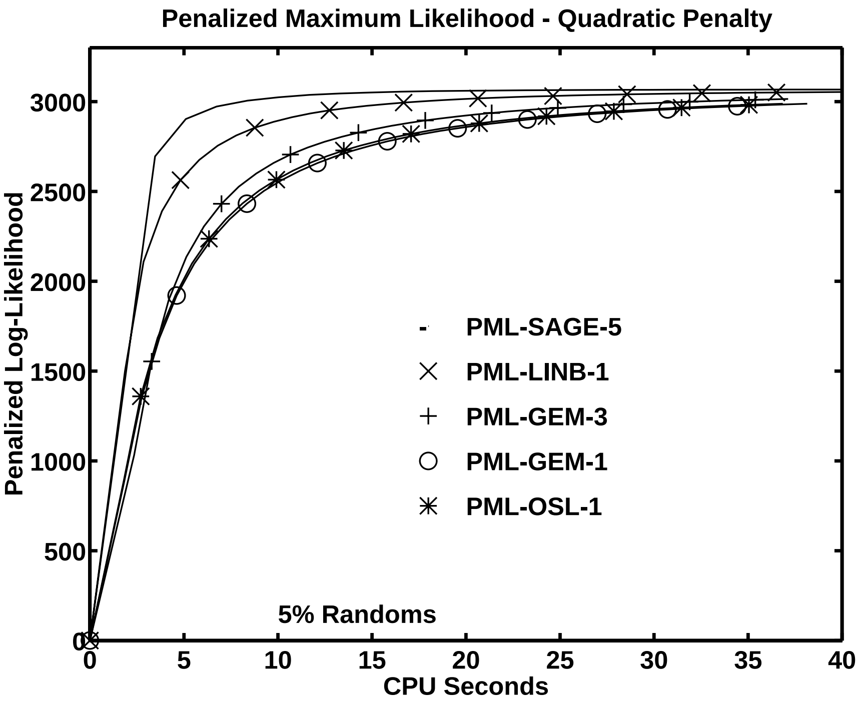

Fig. 18. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. CPU time from data with 5% random coincidences.

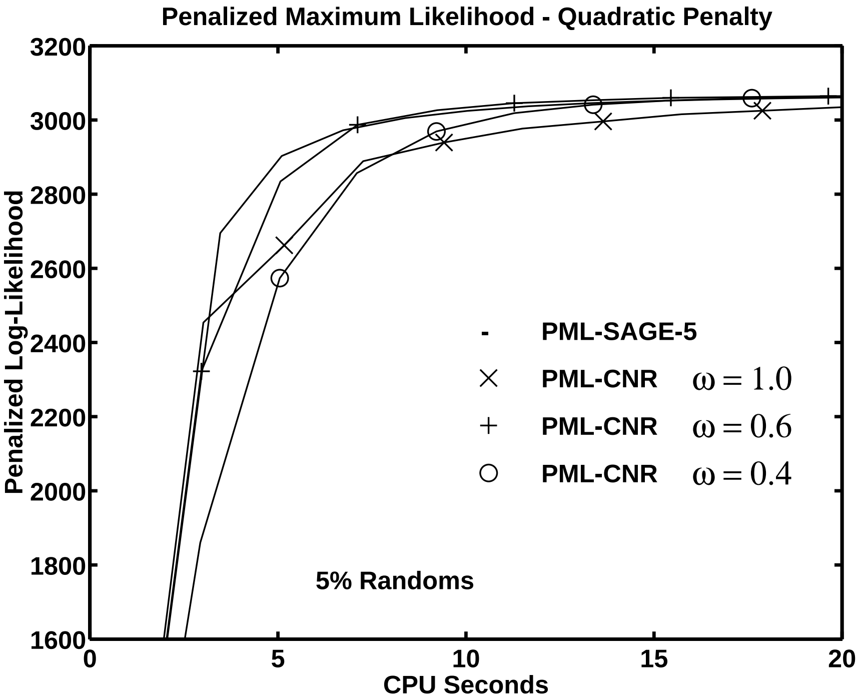

Fig. 19. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. CPU time from data with 5% random coincidences.

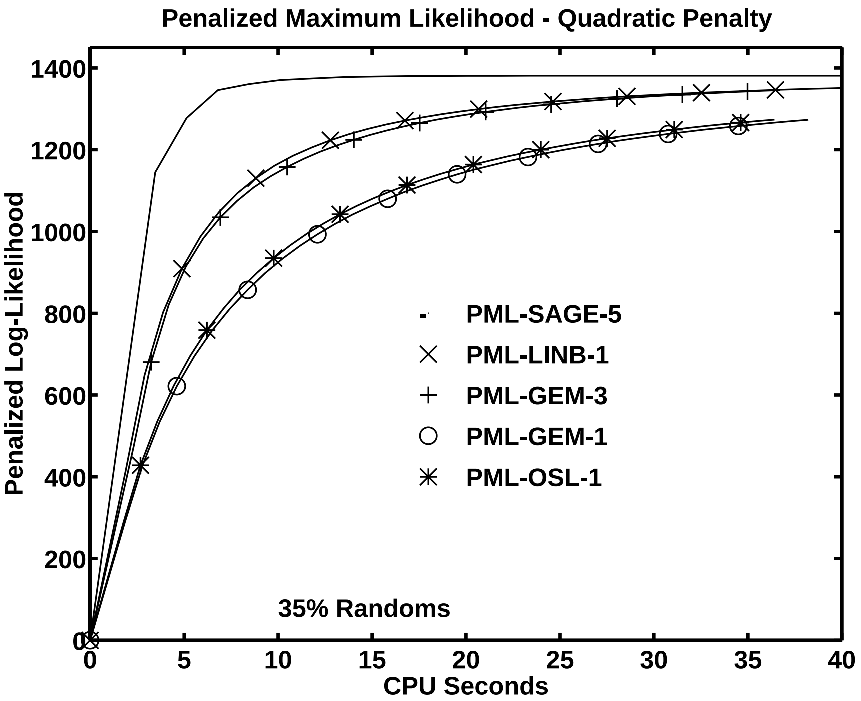

Fig. 20. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. CPU time from data with 35% random coincidences.

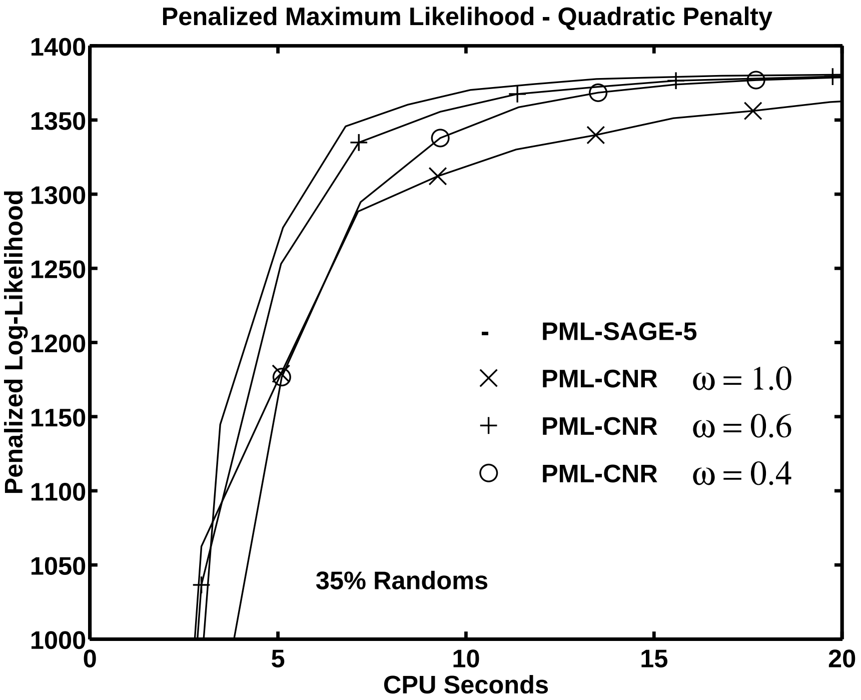

Fig. 21. Penalized likelihood $\Phi(\boldsymbol{\lambda}^i)$ vs. CPU time from data with 35% random coincidences.