Parallel Computation for Positron Emission Tomography with Reduced Processor Communications

A Dissertation

Presented for the

Doctor of Philosophy Degree

The University of Tennessee, Knoxville

Soeren Peter Olesen

May 1996

Acknowledgments

This dissertation is the result of several years of work. I wish like to express my gratitude to a number people who made it possible.

First, I thank my co-advisors Dr. Michael Thomason and Dr. Jens Gregor for providing the opportunity for me to work with them. I am truly honored. I am grateful for their invaluable guidance and for the many hours they have spent discussing research ideas. Dr. Thomason and Dr. Gregor have provided a very productive research environment.

I am thankful to Dr. G. Samuel Jordan, Dr. Michael Berry, and Dr. Jack Dongarra for their time to read this dissertation and for their helpful comments. A special thanks to Dr. Jordan and Dr. Berry for many helpful comments and suggestions along the way. I also thank Dr. Gary Smith, MD (UTMCK) for providing clinical data used in this research.

Finally, a special thank is due to my family and friends for bearing with me while I was working on my research and for making life very enjoyable.

Abstract

Positron Emission Tomography (PET) is a diagnostic technique used to study functionality of human organs. A maximum likelihood reconstruction algorithm for PET, introduced by Shepp and Vardi [32] and Lange and Carson [23], produces images that are statistically superior to other methods, but is an iterative method with high time complexity. Most efforts at parallelization have been for single instruction stream, multiple data stream (SIMD) parallel computers. This research extends the mathematical development of the iterative reconstruction and presents new results in parallel multiple instruction stream, multiple data stream (MIMD) computation with reduced processor communications. We obtain alternative convergence results and new results for rate of convergence of the iterative reconstruction algorithm. We show that the algorithm can be parallelized with reduced communications and show empirically using both simulated and clinical data that the reduced communication algorithm produces reconstructions that are comparable to reconstructions computed with full communication.

Contents

List of Figures

Chapter 1

Introduction and Overview

1.1 Positron Emission Tomography

An introduction to the history and physical basis for PET can be found in [36]. An application of PET for brain imaging is described in [30]. The following description of PET is simplified but includes characteristics necessary to develop a model for a statistical reconstruction algorithm.

A tracer labeled with a positron-emitting nuclide is introduced to the organ under study. A key property used in PET is that when a positron is emitted, it annihilates with a nearby electron and two photons are emitted in approximately opposite directions. The photon pairs are detected by placing the organ under study inside a ring of photon-detectors. The area between a pair of opposing detectors is said to form a tube. An annihilation is recorded to have taken place somewhere within the tube by coincidence detection of the two photons. The recorded data $\boldsymbol{n}^* = \{n_d^* | d = 1, \ldots, D\}$ is a count of coincidence detections for all tubes (detector pairs), where D is the number of tubes. The recorded data $\boldsymbol{n}^*$ is noisy and the sources of distortion include (i) attenuation, (ii) scatter, (iii) random coincidences, and (iv) the inability of actual detectors to respond instantaneously. Figure 1.1 shows some examples of coincidence detection.

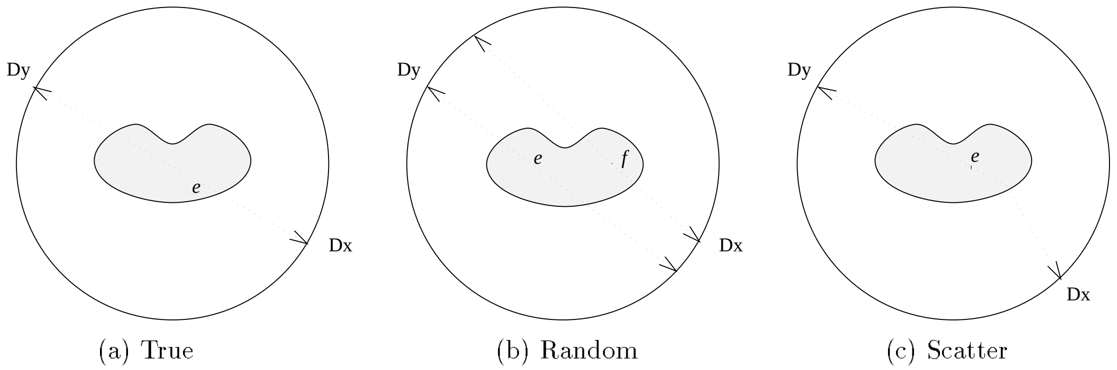


Figure 1.1: Examples of coincidence detection in which the circle represents the gantry of detectors. (a) True detection: Two photons coincidently detected by detector pair (tube) $< D_y, D_x >$, resulting from annihilation at point e in organ. (b) Random detection: Annihilations at e and f result in accidental coincident detection by tube $< D_y, D_x >$. (c) Scatter: Annihilation at point e results in invalid detection by tube $< D_y, D_x >$ because a photon is deflected from its original path of emission.

The reconstruction problem is to estimate a map of positron emissions in the organ. The two main methods for reconstruction in use today are filtered back projection (FBP)[36] and maximum likelihood (ML) parameter estimation[32, 23]. In this study, we focus on the EM-ML (Expectation Maximization, Maximum Likelihood) algorithm described in the following section.

1.2 EM-ML Algorithm

The mathematical model used to develop the ML method makes several assumptions:

1. Positron emissions occur according to a spatial Poisson point process in the region of interest.

2. A grid of B pixels is imposed in the region and the unknown emission density is assumed to be constant within each pixel b.

3. The probability p_{bd} that a photon emission in pixel b is detected by tube d can be computed.

Without loss of generality we normalize $\{p_{bd}\}$ so that $\sum_d p_{bd} = 1$[38]. For mathematical convenience we also assume $n_d^* > 0$ and normalize to have $\sum_d n_d^* = 1$. We refer to the pixels as B-space, to the tubes as D-space, and to $\{p_{bd}\}$ as the P-matrix.

Let N_b be the random variable counting the number of photon emissions at pixel b. There is solid theoretical basis in radiation physics to assume the N_b's are independent and Poisson distributed with unknown means λ_b [36]. Let N_{bd} be the random variable counting detections for tube d from pixel b. It follows [10] that the N_{bd}'s are mutually independent and Poisson distributed with means

$$\lambda_{bd} = \lambda_b p_{bd}.$$

Let N_d^* be the random variable counting the coincident detections of tube d:

$$N_d^* = \sum_b N_{bd}.$$

These random variables are also independent and Poisson distributed[10], so that

$$P(N_d^* = n_d^*) = e^{-\lambda_d^*} \frac{\lambda_d^{* n_d^*}}{n_d^*!}$$

with means

$$\lambda_d^* = \sum_b \lambda_b p_{bd}.$$

The recorded counts $\boldsymbol{n}^* = \{n_d^*\}$ are the sample values of random variables $\{N_d^*\}$. The likelihood of observing $\boldsymbol{n}^*$, given $\boldsymbol{\lambda} = \{\lambda_b\}$, is

$$\begin{aligned} L(\boldsymbol{\lambda}) &= P(\boldsymbol{n}^* | \boldsymbol{\lambda}) \\ &= \prod_d e^{-\lambda_d^*} \frac{\lambda_d^{* n_d^*}}{n_d^*!}. \end{aligned}$$

The maximum likelihood approach is to find the $\boldsymbol{\lambda}$ that maximizes $L(\boldsymbol{\lambda})$ for recorded data $\boldsymbol{n}^*$.

The Expectation Maximization (EM) algorithm is an iterative framework for computing maximum likelihood (ML) estimates originally presented by Dempster *et al.* [8] as a generic tool in statistical estimation. In PET, Shepp and Vardi [32] and Lange and Carson [23] have introduced the same EM algorithm for maximizing $L(\boldsymbol{\lambda})$ and shown that the algorithm converges to a unique limit $\widehat{\boldsymbol{\lambda}}$ that maximizes $L(\boldsymbol{\lambda})$[1]. The algo-

[1]The ML approach and EM algorithm are also used for Single Photon Emission Computed Tomography (SPECT) (cf. [39]) .

rithm gives rise to a multiplicative iteration in which the mean emission at each pixel is re-estimated by the formula

$$\lambda_b^{k+1} = \lambda_b^k \sum_d \frac{n_d^* p_{bd}}{\sum_{b'} \lambda_{b'}^k p_{b'd}}.$$

$\{\lambda_b^0\}$ must be initialized to nonnegative values, e.g., $\lambda_b^0 = \frac{1}{B}$. We will let the function $\ell(\boldsymbol{\lambda}^k) = -\log L(\boldsymbol{\lambda}^k)$ be the negative log-likelihood function and write the EM iteration as a scaled gradient descent algorithm with iteration

$$\lambda_b^{k+1} = \lambda_b^k - \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k).$$

An advantage of the EM-ML method over FBP is that the EM-ML method is based on a model of the actual photon emission process. The EM-ML iteration has the nice property that it ensures nonnegativity of the reconstruction, so $\lambda_b^k \geq 0$ for all b at every iteration k; also, the iteration preserves the total count of detections, i.e.,

$$\sum_d n_d^* = \sum_d \lambda_d^{*k} = \sum_b \lambda_b^k$$

for each iteration k. Furthermore, it has been reported that EM-ML reconstructions produce better estimates of the actual tracer concentration [2, 26] in particular in regions of low activity, and EM-ML reconstruction is recommended for quantitative PET studies [2]. EM-ML is also more tolerant to a primary kind of distortion in projection data.

A comparison of EM-ML and FBP reconstructions computed with distorted projection data [9] reports that EM-ML handles multiplicative distortion better than FBP, whereas FBP is more tolerant to additive distortion. [9] uses EM-ML in clinical studies since, in practice, multiplicative distortion is more important than additive distortion.

A disadvantage of the EM-ML method compared to FBP is that the EM-ML method has a much longer computation time. Several methods have been proposed to reduce the very long computation time for the EM algorithm. A multigrid algorithm is proposed in [31]. Increasing changes at each iteration by an overrelaxation parameter is proposed in [25] and described in more generality in [17]. Minimizing a least-squares function $f(x) = \|\boldsymbol{n}^* - P^T \boldsymbol{\lambda}\|_2^2$ using conjugate gradient algorithm, introduced in [38], is used in [19, 37] and with a preconditioned conjugate gradient algorithm in [18]. Parallel implementations have been reported for hypercube iPSC/2 and shared memory multiprocessor BBN Butterfly GP1000 [5], Alliant FX/8 shared memory multiprocessor [13], vector machine Cray-1 [17], special parallel architecture using two arrays of VLSI chips [15], and massively parallel SIMD computers (AMT-DAP 4096, MasPar 4096) [4, 27, 28].

1.3 Objectives and Overview

This research extends the convergence results for the EM-ML algorithm and presents new results in parallel implementations using both simulated and real clinical data. The dissertation presents the following results:

1. Aspects of sequence and series convergence of the EM-ML algorithm.

2. Convergence of parallel computation with reduced communications.

3. Parallel implementation with reduced processor communications.

4. Experimental results with simulated and clinical data.

Aspects of sequence and series convergence. Convergence of sequences associated with the EM-ML algorithm and optimality is presented in [38]. We present alternative proofs to those and extend the development to include results for series convergence which leads to bounds for the rate of convergence. The negative log-likelihood function $\ell(\boldsymbol{\lambda}^k)$ which is being minimized can be expressed as

$$\ell(\boldsymbol{\lambda}^k) = f(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) + \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}),$$

where $f(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ is a constant function for given vector $\boldsymbol{n}^*$ and $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ is the Kullback discrimination. Thus the convergence analysis will focus on this and other Kullback functions involved in the minimization. Also, to emphasize that the iteration is multiplicative, we write

$$\lambda_b^{k+1} = \lambda_b^k M_b^k$$

where

$$M_b^k = \sum_d \frac{n_d^* p_{bd}}{\sum_{b'} \lambda_{b'}^k p_{b'd}}$$

is the multiplier computed for pixel b at iteration k.

Convergence with reduced communications. A drawback of the parallel computation of the EM-ML algorithm is the amount of communication that is involved in each synchronization. We will show that the EM-ML algorithm can be parallelized with reduced communications while maintaining convergence and optimality. Reducing the communications will lead to processors computing locally with outdated data at times, and in general one cannot assume convergence and optimality. We will show that the EM-ML algorithm belongs to a group of algorithms where convergence to an optimal solution is maintained with reduced communications provided that certain assumptions are met. These assumptions, which are reasonable in practice, include that processors have to communicate *often* enough.

Aspects of parallelization. We have parallelized the EM algorithm for a synchronous, distributed implementation with reduced communications on a Thinking Machines CM-5 multicomputer and a local-area network of workstations; however, the essential contributions to new algorithms are not restricted to these implementations, but are also useful for other distributed memory/message passing machines.

The $\boldsymbol{\lambda}$ - iteration can be written as four sequential computation steps in matrix form:

$$\begin{aligned} \boldsymbol{\lambda}^{*k} &= P^T \boldsymbol{\lambda}^k \\ \forall d : \phi_d^k &= n_d^* / \lambda_d^{*k} \\ \boldsymbol{M}^k &= P \boldsymbol{\phi}^k \end{aligned}$$

$$\forall b : \lambda_b^{k+1} = \lambda_b^k M_b^k.$$

These four steps can be parallelized in various ways depending on how we partition the P matrix. Two possibilities are compressed-block-row or compressed-block-column [22]. Chen *et al.* [5] address aspects of distribution of the P matrix and communication for a synchronous implementation on the Intel iPSC/2 and suggest a partition cyclic-by-row for that hypercube machine. For the reduced communications algorithm we have partitioned the P matrix as compressed-block-row where each node has roughly the same number of nonzero elements and hence about the same amount of work.

Experiments. We present experimental results with both simulated and real clinical data computed using a network of workstations. Creating a model of a PET scanner for the EM-ML algorithm involves estimating the probabilities p_{bd} for every b and d. Recall that p_{bd} is the conditional probability of detection by detector-pair (or tube) d given emission from pixel b. The p_{bd} values, stored as matrix P, depend only on scanner and image geometry. The P matrix is used repeatedly in the algorithm and is computed prior to the reconstruction. We have created P matrices for various scanner geometries and paid special attention to a real scanner, the Siemens/CTI ECAT 921. This scanner has 384 detectors spread evenly on a 82cm diameter gantry and records data for 30720 tubes. If we choose an image size of 128 by 128 and pixel diameter such that the image exactly inscribes the scanner gantry, we get a P matrix for which approximately 8 million of the 500 million entries are nonzero. Results are reported for reconstructions computed with reduced communications and reconstructions computed with full communications. The results compare convergence, difference in reconstructed images and timing.

Chapter 2

Sequence and Series Convergence

2.1 Introduction

We will show that the sequence $\{\boldsymbol{\lambda}^k\}$ generated by the EM-algorithm converges monotonically to a unique limit point $\widehat{\boldsymbol{\lambda}}$ and show that $\widehat{\boldsymbol{\lambda}}$ is a global minimum of the negative log-likelihood function $\ell(\boldsymbol{\lambda}) = -\log L(\boldsymbol{\lambda})$. The proofs of monotonic convergence given here are alternatives to proofs originally presented in [38] and [23]. We will also show that the convergence of certain series leads to a result showing that several sequences are $o(\frac{1}{k})$, i.e., that $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$ at a faster rate than $k \to \infty$.

2.2 Basic Properties and Definitions

This section introduces notation and some properties of the EM-ML algorithm used in subsequent sections. The multiplicative EM-ML iteration is

$$\begin{aligned}\lambda_b^{k+1} &= \lambda_b^k \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}} \\ &= \lambda_b^k (1 - \nabla_b \ell(\boldsymbol{\lambda}^k))\end{aligned}$$

where

$$\lambda_d^{*k} = \sum_b \lambda_b^k p_{bd}.$$

Recall that $\boldsymbol{\lambda}^k = \{\lambda_b^k \mid b = 1, \ldots, B\}$ in B-space is the estimate of the mean emission $\boldsymbol{\lambda}$ at the k^{th} iteration, $\boldsymbol{n}^* = \{n_d^* \mid d = 1, \ldots, D\}$ in D-space is the vector of observed data, and $\boldsymbol{\lambda}^{*k} = \{\lambda_d^{*k} \mid d = 1, \ldots, D\}$ is the projection of $\boldsymbol{\lambda}^k$ into D-space via the P matrix. Without loss of generality we assume normalization for probabilities [38]. Specifically, $\sum_d p_{bd} = \sum_d n_d^* = 1$ with $n_d^* > 0$ for each d, and $\sum_b \lambda_b^0 = 1$ with $\lambda_b^0 > 0$ for each b. It follows from the multiplicative iteration that $\lambda_b^k \geq 0$ and $\lambda_d^{*k} \geq 0$ for each b and d respectively and every iteration k.

Proposition 2.2.1 $\sum_d n_d^* = \sum_d \lambda_d^{*k} = \sum_b \lambda_b^k = 1$ *for* $k = 0, 1, 2, \ldots$

Proof.

$$\sum_b \lambda_b^{k+1} = \sum_b \lambda_b^k \sum_d \frac{n_d^* p_{bd}}{\sum_{b'} \lambda_{b'}^k p_{b'd}}$$

$$= \sum_d n_d^*$$

$$\begin{aligned}
\sum_d \lambda_d^{*k} &= \sum_d \sum_b \lambda_b^k p_{bd} \\
&= \sum_b \lambda_b^k
\end{aligned}$$

□

Proposition 2.2.2 *$\ell(\boldsymbol{\lambda}^k)$ is convex.*

Proof. $\ell(\boldsymbol{\lambda}^k)$ is convex iff the Hessian $\nabla^2 \ell(\boldsymbol{\lambda}^k)$ is nonnegative definite (cf.[3]). The gradient is

$$\nabla_b \ell(\boldsymbol{\lambda}^k) = 1 - \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}}.$$

The Hessian

$$\nabla_{ij}^2 \ell(\boldsymbol{\lambda}^k) = \sum_d \frac{n_d^* p_{id} p_{jd}}{(\lambda_d^{*k})^2}$$

is nonnegative definite if $\boldsymbol{x}^T \nabla^2 \ell(\boldsymbol{\lambda}^k) \boldsymbol{x} \geq 0$ for every nonzero vector $\boldsymbol{x} \in \Re^B$. We have

$$\begin{aligned}
\boldsymbol{x}^T \nabla^2 \ell(\boldsymbol{\lambda}^k) \boldsymbol{x} &= \sum_j \sum_i x_j x_i \nabla_{ij}^2 \ell(\boldsymbol{\lambda}^k) \\
&= \sum_d \frac{n_d^*}{(\lambda_d^{*k})^2} \sum_j x_j p_{jd} \sum_i x_i p_{id} \\
&= \sum_d \frac{n_d^*}{(\lambda_d^{*k})^2} (\sum_b x_b p_{bd})^2 \\
&\geq 0.
\end{aligned}$$

□

Corollary 2.2.1 *If $D > B$, $n_d^* > 0$ for each d, and P has full rank, then $\ell(\boldsymbol{\lambda}^k)$ is strictly convex.*

Proof. From Proposition 2.2.2 we have $\boldsymbol{x}^T \nabla^2 \ell(\boldsymbol{\lambda}^k)\boldsymbol{x} \geq 0$ for every nonzero vector $\boldsymbol{x} \in \Re^B$. To show that $\ell(\boldsymbol{\lambda}^k)$ is strictly convex, we show that the inequality is strictly positive. By contradiction

$$\boldsymbol{x}^T \nabla^2 \ell(\boldsymbol{\lambda}^k)\boldsymbol{x} = 0 \Leftrightarrow \sum_d \sum_b x_b p_{bd} = 0 \Leftrightarrow \sum_d \boldsymbol{x} \cdot \boldsymbol{p}_{+d} = 0$$

where $\boldsymbol{x} \cdot \boldsymbol{p}_{+d}$ is the inner product of vector $\boldsymbol{x}$ and vector $\boldsymbol{p}_{+d}$ which is the dth row of the P matrix. But $\sum_d \boldsymbol{x} \cdot \boldsymbol{p}_{+d} = 0$ iff $\boldsymbol{x}$ is perpendicular to $\boldsymbol{p}_{+d}$ for all d, and since $D > B$ and P has full rank we have $\sum_d \boldsymbol{x} \cdot \boldsymbol{p}_{+d} > 0$. □

Let

$$\mathcal{P}_B = \{(\lambda_1, \ldots, \lambda_B) \mid \lambda_b \geq 0, \ \sum_b \lambda_b = 1\}$$

be the set of candidate solutions in B-space. $\mathcal{P}_B$ is the convex hull of the B extreme points $\{\boldsymbol{\lambda}_b = (\lambda_{b1}, \ldots, \lambda_{bB}) \mid 1 \leq b \leq B\}$, where $\lambda_{bj} = 0$ if $b \neq j, \lambda_{bj} = 1$ if $b = j$, hence, is a *convex set* [1]. The set of candidate solutions in D-space,

$$\mathcal{P}_D = \{(\lambda_1^*, \ldots, \lambda_D^*) \mid \lambda_d^* > 0, \sum_d \lambda_d^* = 1\},$$

is the interior of a convex set, hence, convex [1].

Let

$$\mathcal{Q}_{BD} = \{(q_{11}, q_{12}, \ldots, q_{BD}) \mid q_{bd} = \lambda_b p_{bd}, \lambda_b \geq 0, \sum_b \lambda_b = 1\}$$

and

$$\mathcal{P}_{BD} = \{(\pi_{11}, \pi_{12}, \ldots, \pi_{BD}) \mid \pi_{bd} \geq 0, \sum_b \pi_{bd} = n_d^*\}.$$

Define two probabilities: $\{q_{bd}^k\} \in \mathcal{Q}_{BD}$ where $q_{bd}^k = \lambda_b^k p_{bd}$ and $\{\pi_{bd}^{k+1}\} \in \mathcal{P}_{BD}$ where $\pi_{bd}^{k+1} = q_{bd}^k (n_d^* / \lambda_d^{*k})$.

Note that with this notation the re-estimation of $\boldsymbol{\lambda}^{k+1}$ from $\boldsymbol{\lambda}^k$ in B-space is equivalent to the sequence:

$$\begin{array}{ccccccccc} B & \rightarrow & BD & \rightarrow & D & \rightarrow & BD & \rightarrow & B \\ \lambda_b^k & \rightarrow & q_{bd}^k = \lambda_b^k p_{bd} & \rightarrow & \lambda_d^{*k} = \sum_b q_{bd}^k & \rightarrow & \pi_{bd}^{k+1} = q_{bd}^k \frac{n_d^*}{\lambda_d^{*k}} & \rightarrow & \lambda_b^{k+1} = \sum_d \pi_{bd}^{k+1}. \end{array}$$

2.3 Monotonic Sequence Convergence

Monotonic convergence of certain sequences in D-space and B-space is established in this section. The negative log-likelihood function $\ell(\boldsymbol{\lambda})$ is a convex function for which a minimum is sought given the observed data $\{n_d^*\}$ of positive values and the nonnegativity constraints on $\{\lambda_b\}$. This function is

$$\begin{aligned} \ell(\boldsymbol{\lambda}^k) &= \sum_d \lambda_d^{*k} - \sum_d n_d^* \log \lambda_d^{*k} + \sum_d \log(n_d^*!) \\ &= \sum_d \lambda_d^{*k} + \sum_d \log(n_d^*!) - \sum_d n_d^* \log \lambda_d^{*k} \pm \sum_d n_d^* \log n_d^* \\ &= f(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) + \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \end{aligned}$$

where $f(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ depends only on the observed data $\boldsymbol{n}^*$ and the Kullback discrimination,

$$\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) = \sum_d n_d^* \log \frac{n_d^*}{\lambda_d^{*k}},$$

is nonnegative and equals 0 iff its two arguments are identical [21]; thus, for given $\boldsymbol{n}^*$

$$\ell \textit{ minimized} \quad \Leftrightarrow \quad \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \textit{ minimized},$$

which in turn is true iff the sequence $\{\boldsymbol{\lambda}^k\}$ converges to a fixed point $\widehat{\boldsymbol{\lambda}}$ for which $\mathcal{D}(\boldsymbol{n}^*, \widehat{\boldsymbol{\lambda}}^*) = 0$. This convergence has previously been established by showing $\boldsymbol{\lambda}^k \to \widehat{\boldsymbol{\lambda}}$ where $\widehat{\boldsymbol{\lambda}}$ satisfies the Karush-Kuhn-Tucker conditions for optimality [32, 38, 23].

It can also be convenient and informative to establish that $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$ by direct expansion of certain Kullback discrimination functions together with a result in optimization theory for convex functions and convex sets. The following proposition will establish that $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) \leq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ holds for $k = 0, 1, 2, \ldots$, from which convergence to limit points follows immediately.

Proposition 2.3.1 *The sequence* $\{\boldsymbol{\lambda}^k\}$ *converges monotonically in terms of the Kullback discrimination* $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ *to a fixed point* $\widehat{\boldsymbol{\lambda}}$.

Proof. We first show that $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) \geq \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k)$ for $k = 0, 1, 2, \ldots$. For the Kullback measure $\mathcal{D}(\boldsymbol{\pi}^{k+1}, \boldsymbol{\pi}^{k+2})$, we have

$$0 \quad \leq \quad \mathcal{D}(\boldsymbol{\pi}^{k+1}, \boldsymbol{\pi}^{k+2})$$

$$
\begin{aligned}
&= \sum_{b,d} \pi_{bd}^{k+1} \log \frac{\pi_{bd}^{k+1}}{\pi_{bd}^{k+2}} \\
&= \sum_{b,d} \pi_{bd}^{k+1} \log \frac{\lambda_b^k \lambda_d^{*k+1}}{\lambda_d^{*k} \lambda_b^{k+1}} \\
&= \sum_{b,d} \pi_{bd}^{k+1} (\log \frac{\lambda_d^{*k+1}}{\lambda_d^{*k}} - \log \frac{\lambda_b^{k+1}}{\lambda_b^k}) \\
&= \sum_d (\log \frac{\lambda_d^{*k+1}}{\lambda_d^{*k}}) \frac{n_d^*}{\lambda_d^{*k}} \sum_b \lambda_b^k p_{bd} - \sum_b (\log \frac{\lambda_b^{k+1}}{\lambda_b^k}) \lambda_b^k \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}} \\
&= \sum_d (\log \frac{\lambda_d^{*k+1}}{\lambda_d^{*k}}) \frac{n_d^*}{\lambda_d^{*k}} \lambda_d^{*k} - \sum_b (\log \frac{\lambda_b^{k+1}}{\lambda_b^k}) \lambda_b^k M_b^k \\
&= \sum_d n_d^* \log \frac{\lambda_d^{*k+1}}{\lambda_d^{*k}} - \sum_b \lambda_b^{k+1} \log \frac{\lambda_b^{k+1}}{\lambda_b^k} \\
&= \sum_d n_d^* \log \frac{\lambda_d^{*k+1}}{\lambda_d^{*k}} - \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \\
&= \sum_d n_d^* (\log \frac{\lambda_d^{*k+1}}{\lambda_d^{*k}} \pm \log n_d^*) - \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \\
&= \sum_d n_d^* \log \frac{n_d^*}{\lambda_d^{*k}} - \sum_d n_d^* \log \frac{n_d^*}{\lambda_d^{*k+1}} - \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \\
&= \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) - \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k).
\end{aligned}
$$

Therefore

$$
\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) \geq \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k)
$$

where

$$
\mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \begin{cases} > 0 & \text{if } \boldsymbol{\lambda}^{k+1} \neq \boldsymbol{\lambda}^k \\ = 0 & \text{if } \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k. \end{cases}
$$

This establishes

$$
\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*0}) \geq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \geq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) \geq 0
$$

for $k = 0, 1, \ldots$ In fact, there is strict inequality

$$\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) > 0$$

in the iteration to the limit point $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^*)$. Since $\{\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})\}$ is a monotonically decreasing sequence of real numbers bounded below by 0, it converges [20, p.80]; furthermore,

$$\begin{aligned}
\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) \to 0 \quad &\Rightarrow \quad \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^{k}) \to 0 \\
&\Rightarrow \quad \boldsymbol{\lambda}^{k} \to \widehat{\boldsymbol{\lambda}} \\
&\Rightarrow \quad \boldsymbol{\lambda}^{*k} \to \widehat{\boldsymbol{\lambda}}^{*}
\end{aligned}$$

where $\widehat{\boldsymbol{\lambda}}$ is a limit point of the iteration sequence $\{\boldsymbol{\lambda}^k\}$ in B-space and $\widehat{\boldsymbol{\lambda}}^*$ is the associated limit point of $\{\boldsymbol{\lambda}^{*k}\}$ in D-space. □

Proposition 2.3.2 *Fixed point $\widehat{\boldsymbol{\lambda}}$ is a minimizer of the negative log-likelihood $\ell(\boldsymbol{\lambda})$.*

Proof. A standard result in optimization of convex functions on convex sets is that a necessary and sufficient condition for fixed point $\widehat{\boldsymbol{\lambda}}$ to be a minimizer of $\ell(\boldsymbol{\lambda})$ is that

$$(\boldsymbol{\lambda} - \widehat{\boldsymbol{\lambda}})^T \nabla \ell(\widehat{\boldsymbol{\lambda}}) \geq 0$$

for every $\boldsymbol{\lambda} \in \mathcal{P}_B$ [3]. But since $\boldsymbol{\lambda} \geq 0$ and

$$\begin{aligned}
\widehat{\boldsymbol{\lambda}}^T \nabla \ell(\widehat{\boldsymbol{\lambda}}) &= \sum_b \widehat{\lambda}_b \nabla_b \ell(\widehat{\boldsymbol{\lambda}}) \\
&= \sum_b \widehat{\lambda}_b - \sum_b \widehat{\lambda}_b \sum_d \frac{n_d^*}{\widehat{\lambda}_d^*} p_{bd} \\
&= \sum_b \widehat{\lambda}_b - \sum_d n_d^* \\
&= 0,
\end{aligned}$$

it suffices to show that $\nabla_b \ell(\widehat{\boldsymbol{\lambda}}) \geq 0$ for every b. By contradiction, suppose that there exists a pixel b for which $\nabla_b \ell(\widehat{\boldsymbol{\lambda}}) < 0$. Then $1 - \nabla_b \ell(\boldsymbol{\lambda}^k)$ converges to a value greater than 1 which in turn implies that $\lambda_b^k \to \infty$, but this is a contradiction of the convergence $\lambda_b^k \to \widehat{\lambda}_b$ to a finite $\widehat{\lambda}_b$. □

Corollary 2.3.1 $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$.

Proof. From Proposition 2.3.1, $\boldsymbol{\lambda}^k \to \widehat{\boldsymbol{\lambda}}$ which means that $\boldsymbol{\lambda}^{*k} \to \widehat{\boldsymbol{\lambda}}^*$. The corollary then follows since $\widehat{\boldsymbol{\lambda}}$ is a minimizer of $\ell(\boldsymbol{\lambda})$ which implies that $\widehat{\boldsymbol{\lambda}}^* = \boldsymbol{n}^*$. □

Recall that $\ell(\boldsymbol{\lambda})$ is a convex function and thus all optimal points are global. If $\ell(\boldsymbol{\lambda})$ is strictly convex, then the optimal point will be unique.

It is interesting to consider the b's for which $\widehat{\lambda}_b = 0$. From the iteration

$$\begin{aligned}
\lambda_b^{k+1} &= \lambda_b^k M_b^k \\
&= \lambda_b^0 (M_b^0 M_b^1 \dots M_b^k)
\end{aligned}$$

we have that a pixel can become equal to 0 in two ways: (1) an individual multiplier M_b^k is computed to be 0, or (2) the product of nonzero multipliers is 0 in the limit. For the first case we have from the multiplier equation

$$M_b^k = \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}}$$

that $M_b^k = 0$ iff $n_d^* = 0$ for all the d's where $p_{bd} > 0$. The pixels, whose computed multiplier is zero, can be easily identified and removed from the computation. It is assumed hereafter that they are removed and that no computed multiplier is 0. For the second case all computed multipliers $\{M_b^k\}$ are greater than 0; however, the product of multipliers $(M_b^0 M_b^1 \ldots M_b^k)$ is 0 in the limit and is said to "diverge to 0" [20, p.221]. (For index b where $\widehat{\lambda}_b > 0$, of course, the multiplier product is convergent to a fixed value greater than 0.)

Corollary 2.3.2 *The computed multipliers M_b^k converge to 1.*

Proof. We have previously shown that convergence in D-space is $\boldsymbol{\lambda}^{*k} \to \boldsymbol{n}^*$. It then follows from the multiplier equation $M_b^k = \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}}$ that $M_b^k \to 1$. □

Proposition 2.3.3 *For every d and $k = 0, 1, 2, \ldots$ it follows that $0 < \lambda_d^{*k} \leq 1$.*

Proof. We assume that $n_d^* > 0$ for every d. We have already shown that the convergence in D-space is $\lambda_d^{*k} \to n_d^*$. Thus we need to show that λ_d^{*k} can not take the value 0 while converging to $n_d^* > 0$. Suppose contrary to the proposition that for some d and k we have

$$\lambda_d^{*k} = \sum_{b'} \lambda_{b'}^k p_{b'd} = 0.$$

Then $\lambda_{b'}^k p_{b'd} = 0$ for all b'. The $p_{b'd}$ are unchanging and if $\lambda_{b'}^k = 0$, then by the multiplicative iteration we have $\lambda_{b'}^{k+1} = 0$. Thus $\lambda_d^{*k+m} = 0$ for $m \geq 0$, but this is a contradiction of $\lambda_d^{*k} \to n_d^* > 0$. Finally, since $\sum_d \lambda_d^{*k} = 1$ for $k = 0, 1, 2, \ldots$, we have established that $0 < \lambda_d^{*k} \leq 1$. □

In B-space, [38] gives a direct proof of the inequality $\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k)$. We give an alternate proof using *Jensen's inequality*: if f is a convex function of random variable X, then $E(f(X)) \geq f(E(X))$ where $E(-)$ denotes expected value. Since -log X is convex for positive X, this gives $E(\log X) \leq \log E(X)$.

Proposition 2.3.4 $\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) \geq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ *for* $k = 0, 1, 2, \ldots$.

Proof. We have

$$\begin{aligned}
\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) &= \sum_b \widehat{\lambda}_b \log \frac{\lambda_b^{k+1}}{\lambda_b^k} \\
&= \sum_b \widehat{\lambda}_b \log M_b^k \\
&= E_{\lambda_+}(\log \boldsymbol{M}^k)
\end{aligned}$$

where E_{λ_+} denotes expectation with respect to fixed point $\widehat{\lambda}_b$. Note that

$$\begin{aligned}
M_b^k &= \sum_d p_{bd}\left(\frac{n_d^*}{\lambda_d^{*k}}\right) \\
&= E_{p_{b+}}\left(\frac{\boldsymbol{n}^*}{\boldsymbol{\lambda}^{*k}}\right)
\end{aligned}$$

$$> \quad 0$$

where $E_{p_{b+}}$ denotes expectation with respect to p_{bd}. By Jensen's inequality,

$$\log E_{p_{b+}}(\frac{\boldsymbol{n}^*}{\boldsymbol{\lambda}^{*k}}) \geq E_{p_{b+}}(\log \frac{\boldsymbol{n}^*}{\boldsymbol{\lambda}^{*k}})$$

for every b. Therefore,

$$E_{\lambda_+}\left[\log E_{p_{b+}}(\frac{\boldsymbol{n}^*}{\boldsymbol{\lambda}^{*k}})\right] \geq E_{\lambda_+}\left[E_{p_{b+}}(\log \frac{\boldsymbol{n}^*}{\boldsymbol{\lambda}^{*k}})\right]$$

where the left-hand side is $E_{\lambda_+}(\log \boldsymbol{M}^k)$ and the right-hand side is

$$\begin{aligned}
E_{\lambda_+}\left[E_{p_{b+}}(\log \frac{\boldsymbol{n}^*}{\boldsymbol{\lambda}^{*k}})\right] &= \sum_b \widehat{\lambda}_b \sum_d p_{bd}(\log \frac{n_d^*}{\lambda_d^{*k}}) \\
&= \sum_d (\sum_b \widehat{\lambda}_b p_{bd}) \log \frac{n_d^*}{\lambda_d^{*k}} \\
&= \sum_d n_d^* \log \frac{n_d^*}{\lambda_d^{*k}} \\
&= \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}).
\end{aligned}$$

It follows that

$$\begin{aligned}
\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) &= E_{\lambda_+}(\log \boldsymbol{M}^k) \\
&\geq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}).
\end{aligned}$$

□

Note that if $\widehat{\lambda}_b = 0$, then by definition $\widehat{\lambda}_b \log \frac{\widehat{\lambda}_b}{\lambda_b^k} = 0$ for every k in the evaluation of $\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k)$.

Corollary 2.3.3 *From Propositions 2.3.1 and 2.3.4 we obtain the following inequalities:*

$$
\begin{aligned}
\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) &\geq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) \\
&\geq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \\
&\geq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1}) \\
&\geq \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \\
&\geq 0.
\end{aligned}
$$

2.4 Convergence of series

In this section we show that the convergence of certain series leads to a lower bound on the convergence rate of $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$. We then show the presence of an index b such that $\widehat{\lambda}_b = 0$ plays a role in the convergence of sequences and series.

Proposition 2.4.1 $\sum_{k=0}^{\infty} \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^0)$.

Proof. We have

$$\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}).$$

Since all Kullback values are nonnegative, taking the sum on both sides gives

$$\begin{aligned}\sum_{k=0}^{m}\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k}) &\leq \sum_{k=0}^{m}(\mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^k)-\mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^{k+1}))\\ &= \mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^0)-\mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^{m+1}).\end{aligned}$$

Since $\mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^m)\to 0$, it follows from [20, p.233] that in the limit,

$$\begin{aligned}\sum_{k=0}^{\infty}\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k}) &\leq \mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^0)-\lim_{m\to\infty}\mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^{m})\\ &= \mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^0).\end{aligned}$$

□

Proposition 2.4.2 *For fixed* $m \geq 0$, $k\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k+m})\to 0$.

Proof. Since $\sum_{k=0}^{\infty}\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k})$ is a convergent series (Proposition 2.4.1) of positive monotonically decreasing terms (Proposition 2.3.1), we have [20, p.124] $k\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k})\to 0$ and therefore $k\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k+m})\leq k\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k})\to 0$. □

Proposition 2.4.2 implies that $\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k}) = o(\frac{1}{k})$, i.e., that $\mathcal{D}(\boldsymbol{n}^*,\boldsymbol{\lambda}^{*k})\to 0$ at a faster rate than $k\to\infty$.

Proposition 2.4.3 $\sum_{k=0}^{\infty}k\mathcal{D}(\boldsymbol{\lambda}^{k+1},\boldsymbol{\lambda}^k)\leq\mathcal{D}(\widehat{\boldsymbol{\lambda}},\boldsymbol{\lambda}^0)$.

Proof. From Corollary 2.3.3 we have $\mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \leq \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1})$. Consider the finite sum

$$\begin{aligned} s_m &= \sum_{k=0}^{m} k(\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k+1})) \\ &= \sum_{k=0}^{m} \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) - m\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*m+1}). \end{aligned}$$

From Proposition 2.4.2 $m\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*m+1}) \to 0$ and from Proposition 2.4.1

$$\sum_{k=0}^{\infty} k\mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \leq \lim_{m\to\infty} s_m = \sum_{k=0}^{\infty} \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^0).$$

□

The following propositions link the fixed-point value $\widehat{\lambda}_b = 0$ to the convergence or nonconvergence of certain series.

Proposition 2.4.4 *$\sum_{k=0}^{\infty} \nabla_b \ell(\boldsymbol{\lambda}^k)$ is convergent iff $\widehat{\lambda}_b > 0$.*

Proof. From Proposition 2.3.1 we know that $\nabla_b \ell(\boldsymbol{\lambda}^k) \to 0$; however, this is not sufficient for convergence of the series. For every pixel b we have

$$\begin{aligned} \mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid &= \mid 1 - M_b^k \mid \\ &= \mid \sum_d p_{bd} - \sum_d \frac{n_d^*}{\lambda_d^{*k}} p_{bd} \mid \\ &= \mid \sum_d p_{bd}(1 - \frac{n_d^*}{\lambda_d^{*k}}) \mid \\ &\leq \sum_d p_{bd} \mid 1 - \frac{n_d^*}{\lambda_d^{*k}} \mid \end{aligned}$$

$$\leq \sum_d \mid 1 - \frac{n_d^*}{\lambda_d^{*k}} \mid .$$

By Proposition 2.3.3 there exists a minimum value $\alpha = \min_{d,k} \lambda_d^{*k} > 0$. From above

$$\begin{aligned} \mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid &\leq \sum_d \mid 1 - \frac{n_d^*}{\lambda_d^{*k}} \mid \\ &= \sum_d \frac{1}{\lambda_d^{*k}} \mid \lambda_d^{*k} - n_d^* \mid \\ &\leq \sum_d \frac{1}{\alpha} \mid \lambda_d^{*k} - n_d^* \mid \\ &= \frac{1}{\alpha} \|\boldsymbol{\lambda}^{*k} - \boldsymbol{n}^*\|_1 \\ &\leq \sqrt{D}\frac{1}{\alpha} \|\boldsymbol{\lambda}^{*k} - \boldsymbol{n}^*\|_2 . \end{aligned}$$

Using a property for Kullback discrimination from Csiszár and Körner [6, p.58] we get

$$\|\boldsymbol{\lambda}^{*k} - \boldsymbol{n}^*\|_2^2 \leq 2 \log 2 \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}).$$

Using this in the above inequality we get

$$\nabla_b \ell(\boldsymbol{\lambda}^k)^2 \leq K \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}),$$

where $0 < K < \infty$. Since

$$\sum_{k=0}^{\infty} \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^0),$$

by Proposition 2.4.1, it follows that

$$\sum_{k=0}^{\infty} \nabla_b \ell(\boldsymbol{\lambda}^k)^2 \leq K\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^0).$$

We have previously shown $\widehat{\lambda}_b > 0$ iff the multiplier product

$$\mathcal{M}_b^{\infty} = \prod_{k=0}^{\infty} M_b^k = \prod_{k=0}^{\infty} (1 - \nabla_b \ell(\boldsymbol{\lambda}^k))$$

is convergent. From [20, p.225 Supplementary theorem] we have: *If* $\sum_k \nabla_b \ell(\boldsymbol{\lambda}^k)^2$ *converges, then* $\sum_k -\nabla_b \ell(\boldsymbol{\lambda}^k)$ *and* $\prod_k (1 - \nabla_b \ell(\boldsymbol{\lambda}^k))$ *converge and diverge together.* Since $\sum_{k=0}^{\infty} \nabla_b \ell(\boldsymbol{\lambda}^k)^2$ converges, it follows that $\sum_{k=0}^{\infty} \nabla_b \ell(\boldsymbol{\lambda}^k)$ converges iff $\mathcal{M}_b^{\infty} > 0$. □

Proposition 2.4.5 $\sum_{k=0}^{\infty} \|\boldsymbol{n}^* - \boldsymbol{\lambda}^{*k}\|_2^2$ *is convergent. If there is an index b for which* $\widehat{\lambda}_b = 0$, *then* $\sum_{k=0}^{\infty} \|\boldsymbol{n}^* - \boldsymbol{\lambda}^{*k}\|_2$ *diverges.*

Proof. Convergence is established by the inequalities above; namely,

$$\sum_{k=0}^{\infty} \|\boldsymbol{n}^* - \boldsymbol{\lambda}^{*k}\|_2^2 \leq K \sum_{k=0}^{\infty} \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \leq K\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^0).$$

From Proposition 2.4.4 we have that $\widehat{\lambda}_b = 0$ implies $\sum_{k=0}^{\infty} \nabla_b \ell(\boldsymbol{\lambda}^k)$ is divergent. This implies that $\sum_{k=0}^{\infty} \mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid$ is also divergent, and in this case the inequality

$$\mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid \leq \sqrt{D}\frac{1}{\alpha}\|\boldsymbol{n}^* - \boldsymbol{\lambda}^{*k}\|_2$$

establishes that $\sum_{k=0}^{\infty} \|\boldsymbol{n}^* - \boldsymbol{\lambda}^{*k}\|_2$ is divergent. □

Proposition 2.4.6 *For every b, $\sum_{k=0}^{\infty} \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k)$ is convergent and $\sum_{k=0}^{\infty} \widehat{\lambda}_b \nabla_b \ell(\boldsymbol{\lambda}^k)$ is convergent.*

Proof. From the iteration for pixel b,

$$\lambda_b^{k+1} = \lambda_b^k - \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k)$$

we get that

$$\begin{aligned} \sum_{k=0}^{m} \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k) &= \sum_{k=0}^{m} (\lambda_b^k - \lambda_b^{k+1}) \\ &= \lambda_b^0 - \lambda_b^{m+1}. \end{aligned}$$

Since $\lambda_b^m \to \widehat{\lambda}_b$, it follows from [20, p.233] that in the limit

$$\sum_{k=0}^{\infty} \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k) = \lambda_b^0 - \widehat{\lambda}_b < \infty.$$

For the second series in the Proposition, either $\widehat{\lambda}_b = 0$ or $\widehat{\lambda}_b > 0$ and $\widehat{\lambda}_b \sum_{k=0}^{\infty} \nabla_b \ell(\boldsymbol{\lambda}^k)$ is convergent by Proposition 2.4.4. □

Corollary 2.4.1 *If there is an index b for which $\widehat{\lambda}_b = 0$, then $\sum_{k=0}^{\infty} k^2 \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ diverges.*

Proof. Using Csiszár and Körner [6, p.58] we have $\|\boldsymbol{n}^* - \boldsymbol{\lambda}^{*k}\|_2 \leq \sqrt{2 \log 2 \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})}$; thus,

$$\sum_{k=0}^{\infty} \sqrt{\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})} = \sum_{k=0}^{\infty} \frac{\sqrt{k^2 \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})}}{k}$$

diverges. Following [20, p.125] we have that if

$$\sum_{k=0}^{\infty} \frac{\sqrt{k^2 \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})}}{k}$$

diverges, then $k^2\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) < 0$ for some k or $\sum_{k=0}^{\infty} k^2\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ diverges. Since we have $k^2\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \geq 0$ for all k, the result is shown. $\square$

2.5 Aspects of Sequence and Series Convergence

The preceding results on convergence of nonnegative series allow us to compare rates of convergence and asymptotic ratios of the associated series and sequences. The first proposition shows that the two sequences $\sum_b \lambda_b^{k+1}(\nabla_b \ell(\boldsymbol{\lambda}^k))^2$ and $\sum_b \lambda_b^k(\nabla_b \ell(\boldsymbol{\lambda}^k))^2$ are asymptotically equal.

Proposition 2.5.1 $\lim_{k\to\infty} \dfrac{\sum_b \lambda_b^{k+1}(\nabla_b \ell(\boldsymbol{\lambda}^k))^2}{\sum_b \lambda_b^k(\nabla_b \ell(\boldsymbol{\lambda}^k))^2} = 1.$

Proof.

$$\begin{aligned}
\sum_b \lambda_b^{k+1}(\nabla_b \ell(\boldsymbol{\lambda}^k))^2 &= \sum_b \lambda_b^k M_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \\
&= \sum_b \lambda_b^k (1 - \nabla_b \ell(\boldsymbol{\lambda}^k))(\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \\
&= \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 - \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^3.
\end{aligned}$$

Thus,

$$\frac{\sum_b \lambda_b^{k+1}(\nabla_b \ell(\boldsymbol{\lambda}^k))^2}{\sum_b \lambda_b^k(\nabla_b \ell(\boldsymbol{\lambda}^k))^2} = 1 - \frac{\sum_b \lambda_b^k(\nabla_b \ell(\boldsymbol{\lambda}^k))^3}{\sum_b \lambda_b^k(\nabla_b \ell(\boldsymbol{\lambda}^k))^2},$$

so we want to show

$$\frac{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^3}{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2} \to 0.$$

By definition of null sequence [20, p.45] we have to show that for an arbitrary positive ϵ there is a k_0 such that for all $k > k_0$

$$\left| \frac{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^3}{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2} \right| < \epsilon.$$

We have

$$\begin{aligned} \mid \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^3 \mid &\leq \sum_b \lambda_b^k \mid (\nabla_b \ell(\boldsymbol{\lambda}^k))^3 \mid \\ &= \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid. \end{aligned}$$

Fix $\epsilon > 0$. Since $\mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid \to 0$ for each b, we have $\mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid < \epsilon$ for $k \geq K(b)$. Because there are only finitely many b, the same estimate holds for all b and for all $k \geq \max_b K(b)$. Thus,

$$\begin{aligned} \left| \frac{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^3}{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2} \right| &\leq \frac{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid}{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2} \\ &< \frac{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \epsilon}{\sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2} \\ &= \epsilon. \end{aligned}$$

$\square$

Corollary 2.5.1 $\lim_{k\to\infty} \dfrac{\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))}{\sum_b \lambda_b^{k+1}(\nabla_b \ell(\boldsymbol{\lambda}^k))^2} = 1.$

Proof. We have

$$\begin{aligned}
\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k)) &= \sum_b \lambda_b^k M_b^k(-\nabla_b \ell(\boldsymbol{\lambda}^k)) \\
&= \sum_b \lambda_b^k (1 - \nabla_b \ell(\boldsymbol{\lambda}^k))(-\nabla_b \ell(\boldsymbol{\lambda}^k)) \\
&= -\sum_b \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k) + \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \\
&= \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2.
\end{aligned}$$

Thus,

$$\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k)) = \sum_b \lambda_b^k (\nabla_b \ell(\boldsymbol{\lambda}^k))^2 \cong \sum_b \lambda_b^{k+1} (\nabla_b \ell(\boldsymbol{\lambda}^k))^2$$

where $\cong$ denotes asymptotically equal. $\square$

Proposition 2.5.2 $\lim_{k\to\infty} \dfrac{\sum_b \lambda_b^{k+1} \log M_b^k}{\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))} = \frac{1}{2}.$

Proof. Since $\mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid \to 0$ for each b, we have $\mid \nabla_b \ell(\boldsymbol{\lambda}^k) \mid < 1$ for $k \geq K_1(b)$. This estimate holds for all b and for all $k \geq \max_b K_1(b)$. Thus, using Taylor series expansion of $\log(1 - \nabla_b \ell(\boldsymbol{\lambda}^k))$ gives

$$\begin{aligned}
\log(1 - \nabla_b \ell(\boldsymbol{\lambda}^k)) &= -\nabla_b \ell(\boldsymbol{\lambda}^k) + \nabla_b \ell(\boldsymbol{\lambda}^k)^2 \left[-\frac{1}{2} - \frac{1}{3}\nabla_b \ell(\boldsymbol{\lambda}^k) - \frac{1}{4}\nabla_b \ell(\boldsymbol{\lambda}^k)^2 - \cdots \right] \\
&\equiv -\nabla_b \ell(\boldsymbol{\lambda}^k) + \nu_b^k \nabla_b \ell(\boldsymbol{\lambda}^k)^2.
\end{aligned}$$

Multiplication by nonnegative λ_b^{k+1} and summation over a finite number of b's gives

$$\sum_b \lambda_b^{k+1} \log M_b^k = \sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k)) + \sum_b \lambda_b^{k+1} \nu_b^k \nabla_b \ell(\boldsymbol{\lambda}^k)^2.$$

Note that $\nu_b^k \to -\frac{1}{2}$ as $(-\nabla_b \ell(\boldsymbol{\lambda}^k)) \to 0$ [20]. Fix $\epsilon > 0$. For $k \geq K_2(b)$ we have $-\frac{1}{2} - \epsilon \leq \nu_b^k \leq -\frac{1}{2} + \epsilon$. Because there are only finitely many b, this holds for all b for $k \geq \max_b K_2(b)$. For $k \geq \max\{\max_b K_1(b), \max_b K_2(b)\}$ we have

$$1 + \frac{(-\frac{1}{2} - \epsilon)\sum_b \lambda_b^{k+1} \nabla_b \ell(\boldsymbol{\lambda}^k)^2}{\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))} \leq \frac{\sum_b \lambda_b^{k+1} \log M_b^k}{\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))} \leq 1 + \frac{(-\frac{1}{2} + \epsilon)\sum_b \lambda_b^{k+1} \nabla_b \ell(\boldsymbol{\lambda}^k)^2}{\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))},$$

where we used the fact that $\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k)) > 0$. From Corollary 2.5.1 we have that $\sum_b \lambda_b^{k+1}(\nabla_b \ell(\boldsymbol{\lambda}^k))^2$ and $\sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))$ are asymptotically equal. Hence, the first and last terms in the last equation are close to $+\frac{1}{2}$ for k sufficiently large and ϵ small. Thus, the desired limit of the ratio is $\frac{1}{2}$. $\square$

The two series

$$\sum_{k=0}^{\infty} \sum_b \lambda_b^{k+1} \log M_b^k = \sum_{k=0}^{\infty} \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k)$$

and

$$\sum_{k=0}^{\infty} \sum_b \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^k)^2 = \sum_{k=0}^{\infty} \sum_b \lambda_b^{k+1}(-\nabla_b \ell(\boldsymbol{\lambda}^k))$$

are convergent series of positive terms. Since the corresponding two sequences are asymptotically proportional (by Proposition 2.5.2), the series converge at the same rate.

Proposition 2.5.3 $\nabla\mathcal{D}(\widehat{\boldsymbol{\lambda}}, -)$ *is Lipschitz continuous: for some finite* $L > 0$,

$$\|\nabla\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \bar{\boldsymbol{\lambda}}) - \nabla\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\lambda}})\|_2 \le L\|\bar{\boldsymbol{\lambda}} - \tilde{\boldsymbol{\lambda}}\|_2$$

for all $\bar{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\lambda}}$ *in the convex hull* $\mathcal{C}$ *of iterates* $\boldsymbol{\lambda}^0, \boldsymbol{\lambda}^1, \ldots$.

Proof. We will first show that the norm of the Hessian

$$\nabla^2_{ij}\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) = \begin{cases} \frac{\widehat{\lambda}_i}{(\lambda_i^k)^2} & \text{if } i = j \text{ and } \lambda_i \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

has a finite upper bound. Note first that if $\widehat{\lambda}_b = 0$ we have $\nabla^2_{bb}\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) = 0$ for all k. Next observe that there is a minimum value $\alpha = \min_{b',k} \lambda^k_{b'} > 0$, where the minimum is taken only over those b' for which $\widehat{\lambda}_{b'} > 0$. This result is true, since $\lambda^k_{b'} \to \widehat{\lambda}_{b'}$ ($k \to \infty$, $1 \le b' \le B$) and for $\widehat{\lambda}_b > 0$ we have by the multiplicative iteration that $\lambda^k_b > 0$ for all k. Lipschitz continuity follows from the mean value theorem. □

Proposition 2.5.4 *There holds*

$$\frac{1}{K} \le \frac{\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1})}{\sum_b \widehat{\lambda}_b(-\nabla_b \ell(\boldsymbol{\lambda}^k))} \le 1,$$

where $K > 1$.

Proof. Using convexity of $\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k)$ we establish the upper bound:

$$\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) \quad \le \quad (\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})^T \nabla\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k)$$

$$
\begin{aligned}
&= \sum_b \widehat{\lambda}_b M_b^k - \sum_b \widehat{\lambda}_b \\
&= \sum_b \widehat{\lambda}_b (M_b^k - 1) \\
&= \sum_b \widehat{\lambda}_b (-\nabla_b \ell(\boldsymbol{\lambda}^k)).
\end{aligned}
$$

From Lipschitz continuity of $\nabla \mathcal{D}(\widehat{\boldsymbol{\lambda}}, -)$ (Proposition 2.5.3) we have the *Descent Lemma* [3, p.639]:

$$
\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) + (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^T \nabla \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) + \frac{1}{2} L \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2^2
$$

where L is the positive constant from Lipschitz continuity. From above we have $(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^T \nabla \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) = -\sum_b \widehat{\lambda}_b (-\nabla_b \ell(\boldsymbol{\lambda}^k))$; hence,

$$
\sum_b \widehat{\lambda}_b (-\nabla_b \ell(\boldsymbol{\lambda}^k)) \leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) + \frac{1}{2} L \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|_2^2.
$$

Using an inequality for Kullback discrimination from Csiszár and Körner [6, p.58] we have

$$
\begin{aligned}
\sum_b \widehat{\lambda}_b (-\nabla_b \ell(\boldsymbol{\lambda}^k)) &\leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) + L \log 2 \mathcal{D}(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\lambda}^k) \\
&\leq \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}) + L \log 2 (\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1})) \\
&= (1 + L \log 2)(\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1})).
\end{aligned}
$$

Therefore,

$$\frac{1}{K} \leq \frac{\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1})}{\sum_b \widehat{\lambda}_b(-\nabla_b \ell(\boldsymbol{\lambda}^k))} \leq 1,$$

where $K = 1 + L\log 2$. $\square$

The two series

$$\sum_{k=0}^{\infty} \sum_b \widehat{\lambda}_b \log M_b^k = \sum_{k=0}^{\infty} (\mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^k) - \mathcal{D}(\widehat{\boldsymbol{\lambda}}, \boldsymbol{\lambda}^{k+1}))$$

and

$$\sum_{k=0}^{\infty} \sum_b \widehat{\lambda}_b(-\nabla_b \ell(\boldsymbol{\lambda}^k)),$$

are convergent series of positive terms which, by Proposition 2.5.4, have convergence of the same kind [20, p.279].

Chapter 3

A Reduced Communications Scheme

3.1 Introduction to Reduced Communications

In this section we will show that the EM-ML algorithm belongs to a class of iterative algorithms that, when certain assumptions are met, can be parallelized with reduced communications while still assuring convergence to a fixed point that minimizes the negative log-likelihood function $\ell(\boldsymbol{\lambda})$. The development is based on results presented by Bertsekas and Tsitsiklis [3]. In the following we will show that these assumptions can be satisfied in a parallel computation of a relaxed version of the EM-ML iteration, so that convergence will occur.

The following model of parallel computation is assumed. The pixel vector $\boldsymbol{\lambda}$ is partitioned into disjoint subsets, each of which is assigned to a different processor. We use subscript $[i]$ to denote those pixels that are updated by processor $\mathcal{P}_i$; thus, the vector $\boldsymbol{\lambda}_{[i]}$ is the block of pixels assigned to processor $\mathcal{P}_i$. We also let $B_i \in B$ denote the set of indices b for the pixels that are assigned to processor $\mathcal{P}_i$. The update on each processor is computed by the relaxed iteration

$$\begin{aligned} \boldsymbol{\lambda}_{[i]}^{k+1} &= \boldsymbol{\lambda}_{[i]}^{k} - \gamma \boldsymbol{\lambda}_{[i]}^{k} \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik}) \\ &= \boldsymbol{\lambda}_{[i]}^{k} \widetilde{\boldsymbol{M}}_{[i]}^{ik} \end{aligned}$$

where $0 < \gamma \leq 1$ is a positive step size, $\boldsymbol{\lambda}^{ik}$ is the version of vector $\boldsymbol{\lambda}^{k}$ known to processor $\mathcal{P}_i$ at iteration k (which may contain some outdated values),

$$-\boldsymbol{\lambda}_{[i]}^{k} \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik}) = [-\lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^{ik})] \text{ for } b \in B_i$$

is the update direction, and the vector $\widetilde{\boldsymbol{M}}_{[i]}^{ik} = [1 - \gamma + \gamma M_b^{ik}]$ for $b \in B_i$ where M_b^{ik} is the multiplier corresponding to iteration with $\gamma = 1$. Since $n_d^* > 0$ for each d, each computed multiplier

$$M_b^{ik} = \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*ik}},$$

and hence each vector $\widetilde{\boldsymbol{M}}_{[i]}^{ik}$, is positive (exceeds 0) and, for each b, gradient $\nabla_b \ell(\boldsymbol{\lambda}^{ik}) = 0$ iff $M_b^{ik} = 1$. Pixel vector $\boldsymbol{\lambda}^{ik}$ remains nonnegative.

Specifically, the sufficient conditions for convergence are:

(i) For all vectors in the computational space, the function $\ell(\boldsymbol{\lambda})$ being minimized is nonnegative and its gradient $\nabla\ell(\boldsymbol{\lambda})$ is Lipschitz continuous.

(ii) Each update vector on every processor is in a descent direction, the steps taken by processors are not always zero[1], and convergence of steps to zero implies convergence to an optimal fixed-point solution.

(iii) There is a bound on how out-of-date the data on a processor can be.

3.2 EM-ML Iteration with Reduced Communications

Proposition 3.2.1 *$\ell(\tilde{\boldsymbol{\lambda}}) \geq 0$ for all $\tilde{\boldsymbol{\lambda}}$ such that $\tilde{\lambda}_b \geq 0$ for every b.*

Proof. From the definition of $\ell(\tilde{\boldsymbol{\lambda}})$ we need to show that

$$\sum_d \tilde{\lambda}_d^* - \sum_d n_d^* \log \tilde{\lambda}_d^* + \sum_d \log(n_d^*!) \geq 0$$

or, equivalently,

$$\sum_d \log(n_d^*!) \geq -\sum_d \tilde{\lambda}_d^* + \sum_d n_d^* \log \tilde{\lambda}_d^*.$$

For fixed point solution $\widehat{\boldsymbol{\lambda}}$ we know that $\ell(\widehat{\boldsymbol{\lambda}}) \geq 0$, and since $\widehat{\boldsymbol{\lambda}}^* = \boldsymbol{n}^*$ we also have

$$\begin{aligned} \sum_d \log(n_d^*!) &\geq -\sum_d \widehat{\lambda}_d^* + \sum_d n_d^* \log \widehat{\lambda}_d^* \\ &= -1 + \sum_d n_d^* \log n_d^*. \end{aligned}$$

[1]Except in the exceptional case that the initial uniform image λ^0 is in fact optimal, which is never true with real data or with simulated data having diverse regions-of-interest.

Thus, the proposition will hold for an arbitrary nonnegative $\tilde{\boldsymbol{\lambda}}$ if

$$-1+\sum_d n_d^* \log n_d^* \geq -\sum_d \tilde{\lambda}_d^* + \sum_d n_d^* \log \tilde{\lambda}_d^*$$

or, equivalently, if

$$\sum_d \tilde{\lambda}_d^* - 1 \geq \sum_d n_d^* \log \frac{\tilde{\lambda}_d^*}{n_d^*}.$$

To establish this inequality, we use the standard logarithm inequality

$$\log \frac{\tilde{\lambda}_d^*}{n_d^*} \leq \frac{\tilde{\lambda}_d^*}{n_d^*} - 1$$

for real positive numbers $\frac{\tilde{\lambda}_d^*}{n_d^*}$. Since $n_d^* > 0$ for every d and $\sum_d n_d^* = 1$, this implies

$$\begin{aligned}
\sum_d n_d^* \log \frac{\tilde{\lambda}_d^*}{n_d^*} &\leq \sum_d n_d^* \frac{\tilde{\lambda}_d^*}{n_d^*} - 1 \\
&= \sum_d \tilde{\lambda}_d^* - 1.
\end{aligned}$$

$\square$

Proposition 3.2.2 *Fix $\alpha > 0$ and let S_α be the nonempty, closed and convex set of nonnegative pixel vectors λ with the property that $\lambda_d^* \geq \alpha$ for all d for which $n_d^* > 0$; then $\nabla \ell(\boldsymbol{\lambda})$ is Lipschitz continuous on S_α: that is, for all $\bar{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\lambda}} \in S_\alpha$*

$$\|\nabla \ell(\bar{\boldsymbol{\lambda}}) - \nabla \ell(\tilde{\boldsymbol{\lambda}})\|_2 \leq K_1 \|\bar{\boldsymbol{\lambda}} - \tilde{\boldsymbol{\lambda}}\|_2,$$

where $0 < K_1 < \infty$.

Proof. From the mean value theorem it follows that for $\bar{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\lambda}} \in S_\alpha$, there exists some $\boldsymbol{\lambda} = t\bar{\boldsymbol{\lambda}} + (1-t)\tilde{\boldsymbol{\lambda}}, t \in [0,1]$, such that

$$\begin{aligned} \|\nabla \ell(\bar{\boldsymbol{\lambda}}) - \nabla \ell(\tilde{\boldsymbol{\lambda}})\|_2 &= \|\nabla^2 \ell(\boldsymbol{\lambda})(\bar{\boldsymbol{\lambda}} - \tilde{\boldsymbol{\lambda}})\|_2 \\ &\leq \|\nabla^2 \ell(\boldsymbol{\lambda})\|_2 \|\bar{\boldsymbol{\lambda}} - \tilde{\boldsymbol{\lambda}}\|_2 \end{aligned}$$

by the Schwartz Inequality. Since the Hessian satisfies

$$0 \leq \nabla_{ij}^2 \ell(\boldsymbol{\lambda}) = \sum_d \frac{n_d^* p_{id} p_{jd}}{(\lambda_d^*)^2} \leq \sum_d \frac{n_d^* p_{id} p_{jd}}{\alpha} < \infty,$$

it follows that there exists a finite constant $K_1 > 0$ such that

$$\|\nabla \ell(\bar{\boldsymbol{\lambda}}) - \nabla \ell(\tilde{\boldsymbol{\lambda}})\|_2 \leq K_1 \|\bar{\boldsymbol{\lambda}} - \tilde{\boldsymbol{\lambda}}\|_2.$$

□

To ensure that the step taken by $\mathcal{P}_i$ is not always equal to zero, Assumption 5.5(b) in [3] includes that the norm of the step has a lower bound expressed as positive constant K_2 times the norm of the local gradient. Although this constant K_2 is not actually used in the detailed proof of convergence, this inequality is easy to verify for several standard gradient-like methods such as the Jacobi algorithm [3, p.528]. For the EM-ML algorithm, it is more convenient to show directly that convergence $\|\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2 \to 0$

implies an optimal solution.

Proposition 3.2.3

1. *If* $0 \le \lambda_b^k \le 1$ *for all* $b \in B_i$ *and* k*, then*

$$[-\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})]^T \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik}) \le -\|\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2^2$$

for all i *and all* k*.*

2. *For initial vector* $\boldsymbol{\lambda}_{[i]}^0 > 0$*, if* $\|\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2 \to 0$*, then* $\nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik}) \to 0$*.*

Proof. For property 1, recall B_i is the set of indices for the pixels computed by processor $\mathcal{P}_i$; then

$$\begin{aligned}
[-\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})]^T \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik}) &= -\sum_{b \in B_i} \lambda_b^k \nabla_b \ell(\boldsymbol{\lambda}^{ik})^2 \\
&\le -\sum_{b \in B_i} (\lambda_b^k)^2 \nabla_b \ell(\boldsymbol{\lambda}^{ik})^2 \\
&= -\|\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2^2.
\end{aligned}$$

With respect to Property 2, notice that $\|\boldsymbol{\lambda}_{[i]}^k \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2 \to 0$ implies $\boldsymbol{\lambda}_{[i]}^k \to \widehat{\boldsymbol{\lambda}}_{[i]}$. From Proposition 2.3.2 it follows that fixed point $\widehat{\boldsymbol{\lambda}}$ is an optimal point, i.e., a minimizer of $\ell(\boldsymbol{\lambda})$. From Corollary 2.3.2 we have that the computed multipliers $M_b^{ik} \to 1$ for every index $b \in B_i$ (including any index b for which $\lambda_b^k \to 0$). Hence,

$$\nabla_b \ell(\boldsymbol{\lambda}^{ik}) = \frac{1 - M_b^{ik}}{\gamma}$$

$$\rightarrow \quad 0.$$

$\square$

Proposition 3.2.4 *If $\boldsymbol{\lambda}^{ik} \in S_\alpha$ for each i and all k, then there exists some $\gamma_0 > 0$ such that, if $0 < \gamma < \gamma_0$, then the iterates computed by the EM-ML algorithm using the parallel model with reduced communications described above converges to fixed point $\widehat{\boldsymbol{\lambda}}$ if (1) every pixel is updated at least every $\mathcal{B}$ time units, and (2) every processor $\mathcal{P}_i$ receives at least one update from every other processor $\mathcal{P}_j$ during every time interval of length $\mathcal{B}$.*

Proof. Propositions 3.2.1 and 3.2.2 together with part 1 of Proposition 3.2.3 establish that

$$\|\boldsymbol{\lambda}_{[i]}^{k} \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2 \rightarrow 0$$

which implies that $\boldsymbol{\lambda}_{[i]}^{k} \rightarrow \widehat{\boldsymbol{\lambda}}_{[i]}$. This result follows from the proof of Proposition 5.2 in [3]. Our Propositions 3.2.1 and 3.2.2 above establish that Assumption 5.1 of [3] holds. Part 1 of Proposition 3.2.3 establishes part (a) of Assumption 5.5 in [3], from which convergence to a fixed point follows. Part (b) of Assumption 5.5 in [3] is used to establish that the convergence is to an optimal fixed point; in Part 2 of Proposition 3.2.3, we have established a different, but sufficient, result that the convergence is indeed to an optimal fixed point, and in fact, that $\|\boldsymbol{\lambda}_{[i]}^{k} \nabla_{[i]} \ell(\boldsymbol{\lambda}^{ik})\|_2 \rightarrow 0$ implies $\nabla_b \ell(\boldsymbol{\lambda}^{ik}) \rightarrow 0$. $\square$

The maximum step size γ_0 depends on $K_1, \mathcal{B}$, and B. A sufficient, but not necessary, choice for γ_0 is shown in [3] to be

$$\gamma_0 = \frac{1}{K_1(1 + \mathcal{B} + B\mathcal{B})};$$

however, the value of constant K_1 of Lipschitz continuity of $\nabla\ell(\boldsymbol{\lambda})$ is unknown and presumably dependent on observed data $\boldsymbol{n}^*$. In the following we describe a parallel algorithm with reduced communications where the step size is controlled.

3.3 Aspects of Parallel Implementation

In this section we describe aspects of parallel implementation of the EM-ML iteration with reduced communications. This involves the way in which the processors iterate with outdated data and how we avoid taking too big steps, i.e., avoid changes in pixel values that are too large. Since this is a multiplicative technique, it is convenient to control the step size by creating intervals of allowable multipliers. Thus if a computed multiplier is too big or too small the associated pixel value is not updated. As introduced above, we use $\boldsymbol{\lambda}_{[i]}$ to denote the pixel-vector that is updated by processor $\mathcal{P}_i$, and λ_{id}^{*k} to denote the contribution to λ_d^{*k} from processor $\mathcal{P}_i$. The following describes the tasks a processor performs at iterations when synchronization with other processors is performed, and the tasks performed at iterations between synchronization.

At Synchronization. At iterations where the processors synchronize, each processor performs the following tasks:

(i) Communicate updated λ_{id}^{*k} vector and compute new $\boldsymbol{\lambda}^{*k}$ vector.

(ii) Renormalize λ_b^k and λ_d^{*k}.

(iii) Compute multipliers and update $\boldsymbol{\lambda}_{[i]}^{k+1}$.

(iv) Adjust interval of multipliers $[L_i^{k+1}, U_i^{k+1}]$.

Re.(i): Each processor computes its contribution to the D-space vector $\boldsymbol{\lambda}^{*k}$ as

$$\lambda_{id}^{*k} = \sum_{b \in B_i} \lambda_b^k p_{bd}$$

and communicates this vector to all other processors. All processors compute the same

$$\lambda_d^{*k} = \sum_i \lambda_{id}^{*k}$$

and each processor computes

$$\lambda_{id,other}^{*k} = \lambda_d^{*k} - \lambda_{id}^{*k}.$$

The purpose of $\lambda_{id,other}^{*k}$ will be explained later.

Re.(ii): Since the iterations computed between two synchronizations use outdated values for λ_d^*, the D- and B-space vectors will have lost normalization. Renormalization is carried out such that

$$\sum_d \lambda_d^{*k} = \sum_b \lambda_b^k = \sum_d n_d^*.$$

In D-space the renormalization factor,

$$\alpha_D^k = \frac{\sum_d n_d^*}{\sum_d \lambda_d^{*k}},$$

is computed by every processor and used to restore normalization of each element of λ_d^{*k} and $\lambda_{id,other}^{*k}$, i.e.,

$$\lambda_d^{*k} = \alpha_D^k \lambda_d^{*k} \text{ and } \lambda_{id,other}^{*k} = \alpha_D^k \lambda_{id,other}^{*k}.$$

In B-space a similar factor α_B^k is computed. It turns out that

$$\begin{aligned}
\alpha_B^k &= \frac{\sum_d n_d^*}{\sum_b \lambda_b^k} \\
&= \frac{\sum_d n_d^*}{\sum_i \sum_{b \in B_i} \lambda_b^k} \\
&= \frac{\sum_d n_d^*}{\sum_i \sum_{b \in B_i} \lambda_b^k \sum_d p_{bd}} \\
&= \frac{\sum_d n_d^*}{\sum_d \sum_i \sum_{b \in B_i} \lambda_b^k p_{b_d}} \\
&= \alpha_D^k.
\end{aligned}$$

This is convenient since otherwise the processors would have to communicate $\boldsymbol{\lambda}_{[i]}^k$ in order for each processor to compute $\sum_b \lambda_b^k$. To restore normalization in B-space each element of $\boldsymbol{\lambda}_{[i]}^k$ is multiplied by α_B^k i.e., $\boldsymbol{\lambda}_{[i]}^k = \alpha_B^k \boldsymbol{\lambda}_{[i]}^k$. Note that the $\boldsymbol{\lambda}^{*k}$ vector obtained by forward-projecting the renormalized $\boldsymbol{\lambda}^k$ vector is identical to the $\boldsymbol{\lambda}^{*k}$ computed by renormalization. Since

$$\begin{aligned}\lambda_d^{*k} &= \sum_b \lambda_b^k p_{bd} \\ &= \alpha_D^k \sum_b \lambda_b^k p_{bd} \\ &= \alpha_D^k \lambda_d^{*k}.\end{aligned}$$

Re.(iii): Each multiplier

$$M_b^k = \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*k}}$$

is computed with the renormalized vectors and compared to the interval $[L_i^k, U_i^k]$ of allowed multipliers. If a multiplier is in the interval, then the pixel value

$$\lambda_b^{k+1} = \lambda_b^k M_b^k$$

is updated; otherwise, the pixel value is left unchanged. This mechanism prevents too large changes in pixel value.

Re.(iv): The lower range of multipliers is updated such that

$$L_i^{k+1} = \min(1, \max(L_i^k, \min_{b \in B_i} M_b^k)).$$

This means that the lower range is always less than or equal to 1 and that the lower range is only increased if every computed multiplier M_b^k less than 1 is greater than the current range L_i^k. The sequence $\{L_i^k\}$ is thus nondecreasing and bounded above by 1.

The upper bound is updated in similar fashion, with

$$U_i^{k+1} = \max(1, \min(U_i^k, \max_{b \in B_i} M_b^k)).$$

The upper bound is always greater than 1 and the upper bound is only decreased if every multiplier M_b^k greater than 1, is less than the current range U_i^k. The sequence $\{U_i^k\}$ is consequently nonincreasing and bounded below by 1.

Between Synchronization. At iterations where synchronization is not performed, each processor performs the following tasks:

(i) Update λ_{id}^{*k} locally.

(ii) Compute multipliers and $\boldsymbol{\lambda}_{[i]}^{k+1}$.

Re.(i): Each processor computes its contribution to the $\boldsymbol{\lambda}^{*k}$ vector as

$$\lambda_{id}^{*k} = \sum_{b \in B_i} \lambda_b^k p_{bd}.$$

Processors do not exchange $\boldsymbol{\lambda}_i^{*k}$ vectors so in order to compute the complete $\boldsymbol{\lambda}^{*k}$ vector the processors must estimate the contribution from the other processors. This is where $\lambda_{id,other}^{*k}$ is used. The $\boldsymbol{\lambda}_{i,other}^{*k}$ vector computed at the most recent synchronization is used as an estimate of the contribution from all other processors, we denote

$$\lambda_d^{*ik} = \lambda_{id}^{*k} + \lambda_{id,other}^{*k}$$

as the estimate of λ_d^{*k} at processor $\mathcal{P}_i$.

Re.(ii): Each processor can now compute multipliers

$$M_b^k = \sum_d \frac{n_d^* p_{bd}}{\lambda_d^{*ik}},$$

and update λ_b^{k+1} if M_b^k is in the interval $[L_i^k, U_i^k]$. The interval of multipliers $[L_i^k, U_i^k]$ and the vector $\boldsymbol{\lambda}_{i,other}^{*k}$ are only computed at iterations where processors synchronize and do not change values here.

Chapter 4

Experimental results

4.1 Methods

4.1.1 Computer Resources

The distributed version of the EM–ML algorithm is implemented in C for a multicomputer computer as well as a network of workstations. The multicomputer is a Thinking Machines CM–5 with 32 nodes which communicate over high-speed, internal networks. This implementation configures the CM–5 as a MIMD (multiple-instruction, multiple-data) machine and uses the dedicated CMMD library routines [35] for communication by passing arrays of data among processors. The network of workstations is a virtual MIMD machine composed of Sun Sparc5 workstations connected by Ethernet (10 Mbits/sec). This implementation uses PVM (Parallel Virtual Machine [11, 34]), version 3.3.3, a general purpose software package that facilitates distributed computation on a network of

Unix computers, for communication. Each CM–5 node and Sparc5 workstation has 32 Mbytes local memory. The CM–5 RISC processor has a 32 Mhz clock and a peak execution rate of 4 Mflops, and the Sparc5 processor has a 70 Mhz clock and 13.1 Mflops peak rate. The effective rate of Sparc5 communication on the Ethernet, however, averages at least two orders of magnitude slower than CM–5 inter-processor communication.

4.1.2 Sinograms; P Matrix; Initial Image

The experimental work is based on simulated data as well as scan data obtained from an ECAT 921 PET scanner at the University of Tennessee Medical Center, Knoxville (UTMCK). Each 2-D reconstruction $\boldsymbol{\lambda}$ is a 128×128 image. Observed data $\boldsymbol{n}^*$ is given as a 192×160 sinogram. The P-matrix for the EM algorithm thus has 128×128 rows and 192×160 columns. Each p_{bd} is a double-precision, floating-point value computed as the area of intersection of tube d with pixel b (approximated by its inscribed circle) relative to the sum-total intersection of that pixel with all tubes. This yields approximately 8×10^6 nonzero entries in P; these are the only values stored. If the image is centered in the ring of detectors, the number of values that need to be stored can be reduced even further by applying certain symmetry considerations [16, 5]. However, since the P-matrix in this research directly models the ECAT 921 scanner and that scanner allows the image to be located anywhere within the ring of detectors, these symmetry considerations are not exploited here.

Three sinograms are used. The first is obtained by forward projecting a 128×128 simulated phantom [32] through the P-matrix. This sinogram is free of noise and should result in $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$. The second and third are real ECAT 921 sinograms produced by FDG brain scan and ^{13}N-ammonia cardiac scan respectively; the patient scans follow routine clinical protocols and data processing approved by the UTMCK Institutional Review Board. The ECAT 921 sinograms are normalized for detector variability, corrected for attenuation using transmission scans obtained prior to radiopharmaceutical injection, and corrected for estimated randoms. For these two sinograms, $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ is expected to converge to a positive value since the overall number of emissions is not constant across the view angles. This situation arises as follows.

The observed data is obtained by systematically recording emission activity for a fixed number of view angles. We can thus partition the D-space into V such view angles and write

$$D(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) = \sum_{i=1}^{V} \sum_{d \in D_i} n_d^* \log \frac{n_d^*}{\lambda_d^{*k}}$$

where D_i is the set of tubes at the ith view angle. Assuming that emissions are equally likely to be detected at any view angle, it follows that

$$\sum_{d \in D_i} \lambda_d^{*k} = \frac{1}{V} \sum_{i=1}^{V} \sum_{d \in D_i} n_d^*$$

for all i; that is, the activity forward projected by the EM–ML algorithm is the same for all view angles and remains constant throughout the computation. A necessary, but not sufficient, condition for $D(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) = 0$ is therefore that

$$\sum_{d \in D_i} n_d^* = \sum_{d \in D_j} n_d^*$$

for all i and j. But for real scanner data there will most likely be view angles for which this is not true, in which case convergence of the Kullback discrimination to a positive value rather than 0 is expected.

Initial image $\boldsymbol{\lambda}^0$ is a uniform image in which each pixel has the same positive value. The EM–ML iterations are carried out without employing regularization, filtering, or prior probabilities (cf., [33, 24, 12, 7, 14, 29, 39] for some of these methods in PET or SPECT).

4.1.3 Network and Algorithm Parameters

The empirical results concern convergence, global and local (region-of-interest) error evaluation, and time of computation for distributed EM–ML iteration under the following conditions. All runs on the CM–5 and the workstations are made in dedicated modes which prohibit user processes not associated with the EM–ML computation. Systems traffic on the Ethernet is suppressed as far as possible.

Results are given for 8 and 16 processors. As the number of processors is reduced, each processor must deal with larger portions of the P-matrix; this begins to cause considerable memory paging on processors with 32 Mbytes local memory, which slows the computation severely. As the number of processors is increased, the Ethernet tends to saturate with messages even when synchronizations are reduced, and this also severely impacts the time of computation. The number of processors is a practical compromise between processing power and communication for the workstations, the network, and the ECAT 921 scanner in these experiments.

Let C denote the maximum number of local iterations that take place on each processor between synchronization with the other processors. A run with $C = 1$ is full synchronization and corresponds to the standard EM-ML algorithm. The number of synchronizations between processors is reduced in the following manner. Large changes in numerous pixels occur in the first few iterations, so full synchronization is maintained for the first 16 iterations; thereafter, the number of local iterations is linearly increased until reaching C which is then used for the rest of the run. Thus for a given value of C, the number of local iterations at each processor between synchronizations with the other processors can be described by the sequence

$$\underbrace{1, 1, \ldots, 1}_{16\ iterations}, 2, 3, \ldots, C-2, C-1, C, C, C, \ldots, C$$

All runs reported below consist of 512 iterations. Reconstructions are computed for $C = 1$, $C = 4$, and $C = 8$. It follows that $C = 4$ results in 140 synchronizations and $C = 8$ in 80 synchronizations, respectively 27.3% and 15.6% of the 512 synchronizations for $C = 1$.

Recall that in the synchronization step each processor $\mathcal{P}_i$ computes

$$\lambda_d^{*k} = \sum_i \lambda_{id}^{*k}$$

for each d. For the CM–5 implementation this is accomplished with a call to the `CMMD_reduce_v` function of the CMMD library [35]. This function handles communication and summation using internal hardware of the CM-5 [35]. For the network of workstations implementation using PVM we use a *host-node* programming model. A synchronization is performed in two steps. Initially, the *host* processor receives the $\boldsymbol{\lambda}_i^{*k}$ vector from all other processors $\mathcal{P}_i$ and computes $\boldsymbol{\lambda}^{*k}$ as above; this operation is performed with the PVM function `pvm_reduce`. In the second step, the *host* processor broadcasts $\boldsymbol{\lambda}^{*k}$ to all processors using a call to the PVM function `pvm_bcast` [11].

4.2 Experimental Results

4.2.1 Reconstructed Images

Images in Figs. 4.1-4.4 are displayed as integer pixels of one byte each. Figure 4.1 is the simulated phantom $\boldsymbol{\lambda}^{phan}$. Figure 4.2 shows its reconstructions at iterations 32 and 512 for the three values of C. Figure 4.3 shows the reconstructed brain scan (pixel size 2.275 mm) and Fig. 4.4 the reconstructed cardiac scan (pixel size 3.033 mm) for the same parameters. These are reconstructions using 8 processors. Visually, the reconstructions on 16 processors for corresponding parameters are virtually indistinguishable.

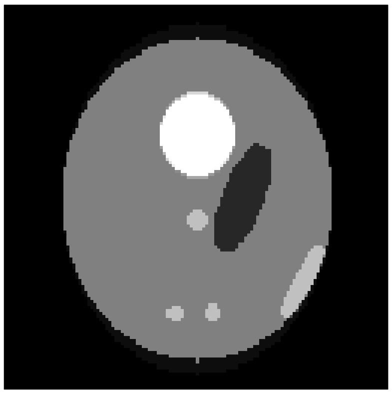


Figure 4.1: 128x128 pixel simulated phantom.

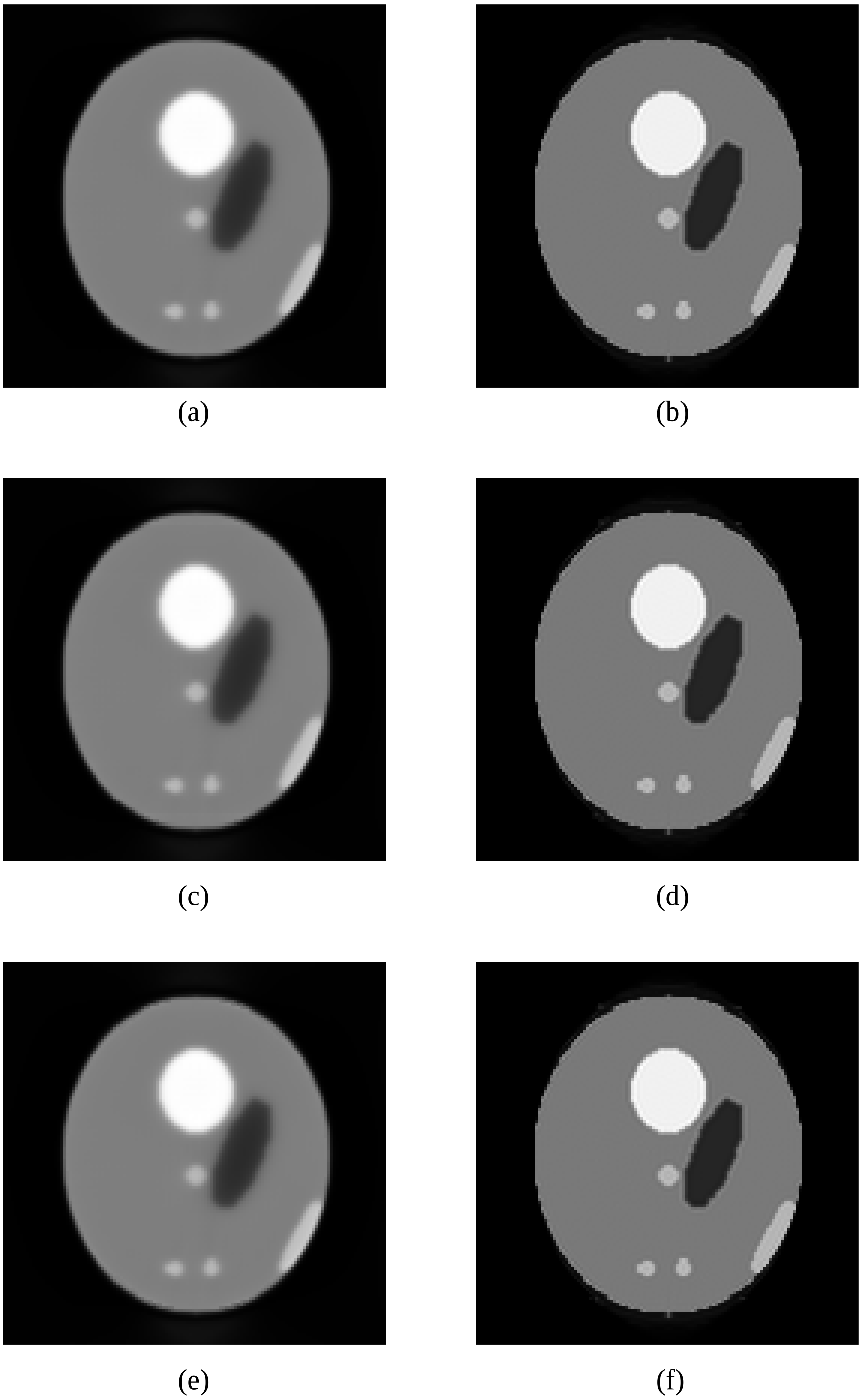
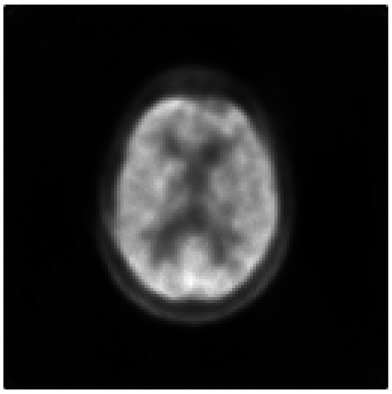
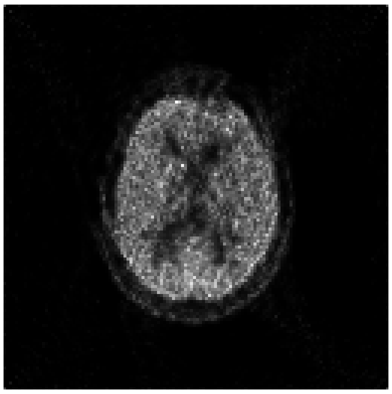


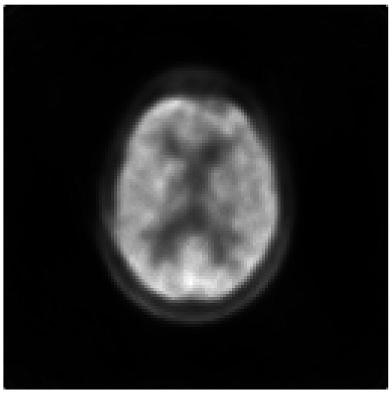
Figure 4.2: Reconstructions of phantom with 8 processors. C=1, C=4, and C=8 from top to bottom. (a), (c), and (e) are 32 iterations;(b), (d), and (f) are 512 iterations.



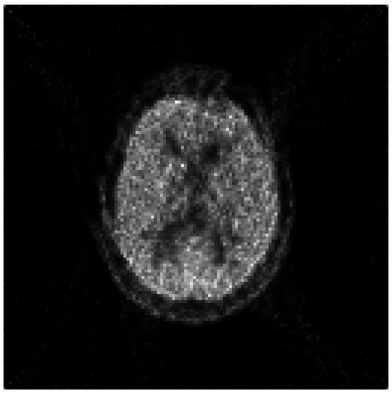
(a)



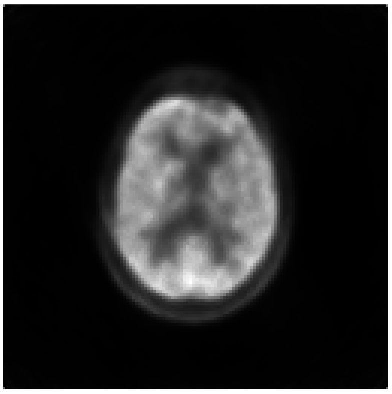
(b)



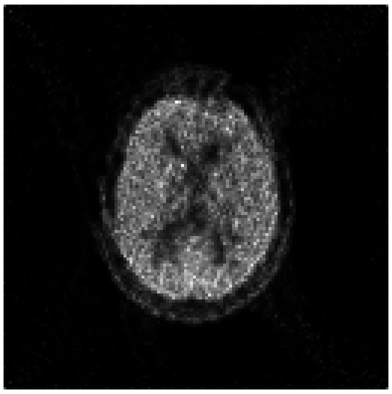
(c)



(d)

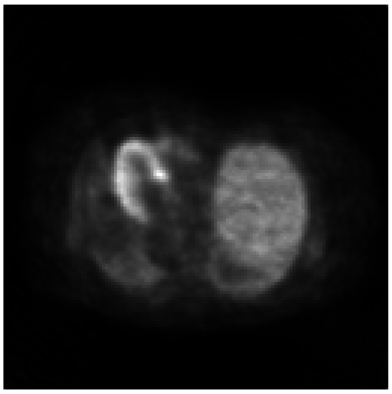


(e)

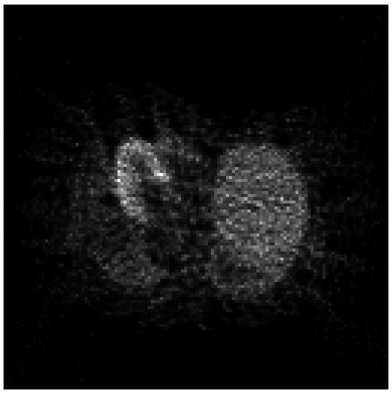


(f)

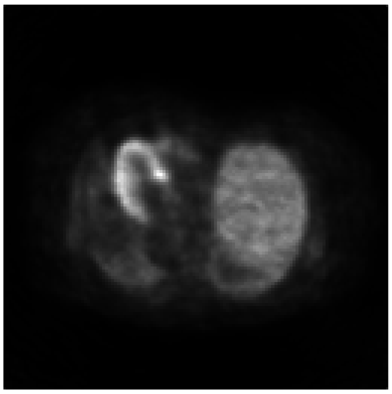
Figure 4.3: Reconstructions of brain scan with 8 processors. C=1, C=4, and C=8 from top to bottom. (a), (c), and (e) are 32 iterations;(b), (d), and (f) are 512 iterations.



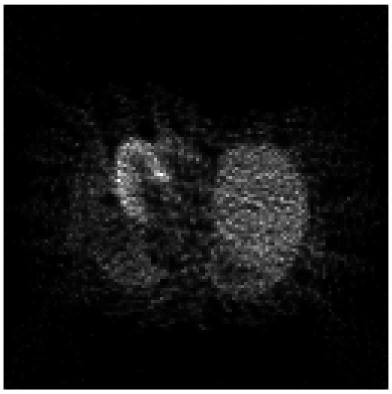
(a)



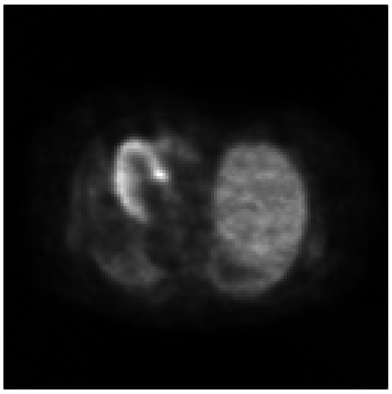
(b)



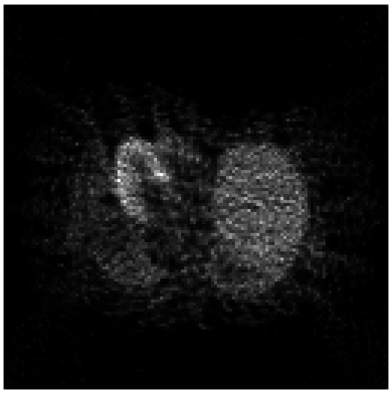
(c)



(d)



(e)



(f)

Figure 4.4: Reconstructions of cardiac scan with 8 processors. C=1, C=4, and C=8 from left to right. (a), (c), and (e) are 32 iterations;(b), (d), and (f) are 512 iterations.

4.2.2 Convergence and Error Evaluation

Figures 4.5 and 4.6 illustrates convergence of the fully synchronized EM–ML algorithm in terms of $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$; the values obtained for $C = 4$ and $C = 8$ essentially coincide with those shown here for $C = 1$. The explicit data points in the plots mark the iterations between 8 and 512 that are powers of 2. Figures 4.5 and 4.6 show no difference between running on 8 and 16 processors. We also note that the Kullback measure converges essentially to 0 for the phantom image and converges to small positive values for the scanner-based brain and cardiac images as predicted.

For 8 processors and the simulated phantom, Fig. 4.7 gives the percentage error $\mathcal{E}^k$ in the reconstructed pixel vector $\boldsymbol{\lambda}^k$ referenced to the original vector $\boldsymbol{\lambda}^{phan}$, computed as

$$\mathcal{E}^k = \frac{\sum_b (\lambda_b^k - \lambda_b^{phan})^2}{\sum_b (\lambda_b^{phan})^2} \times 100.$$

Differences in $\mathcal{E}^k$ for full and reduced synchronizations at corresponding iterations are negligible. At 512 iterations, all values in Fig. 4.7 are less than 0.15%. For 16 processors, Fig. 4.8 shows slightly different values because the finer partitioning of the P-matrix among processors induces different patterns of small-scale numerical inaccuracies during distributed iteration. At 512 iterations, all values in Fig. 4.8 are less than 0.25%.

Fully synchronized reconstructions also serve as references with which to measure errors associated with reduced synchronization. This percentage error is computed for each sinogram, for $C = 4$ and $C = 8$ and for 8 and 16 processors by using the reconstructions for $C = 1$ at corresponding iterations as the references. For the simulated

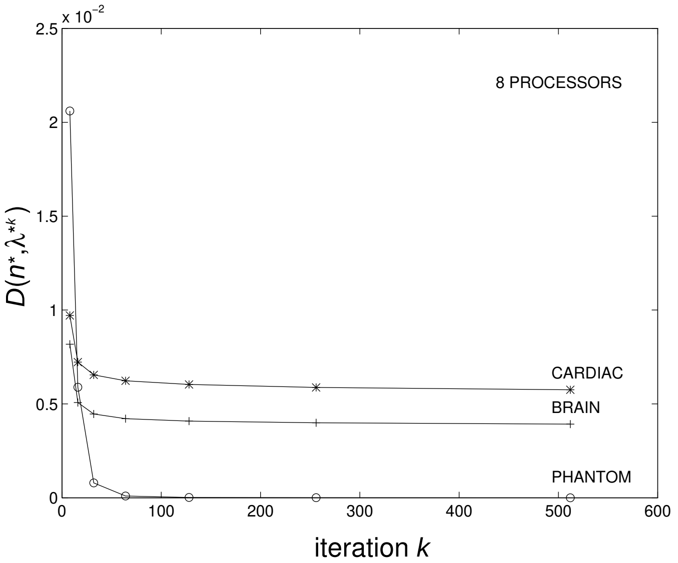


Figure 4.5: Kullback measure using natural logarithms, computed with 8 processors.

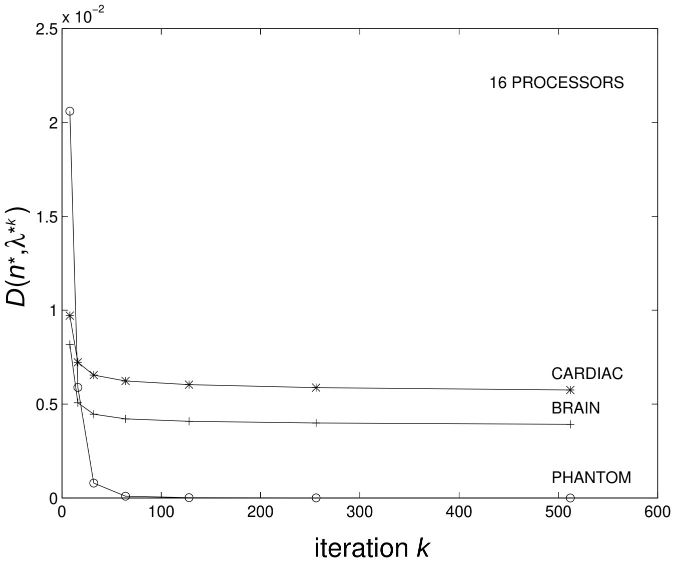


Figure 4.6: Kullback measure using natural logarithms, computed with 16 processors.

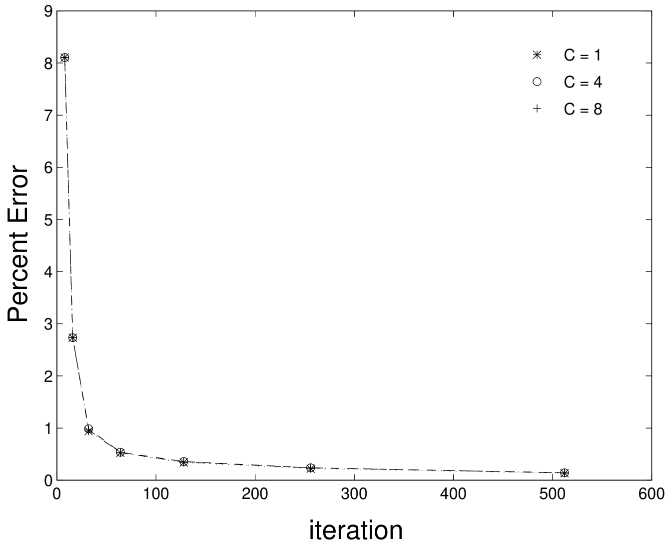


Figure 4.7: Percentage error in phantom reconstruction with respect to original, computed with 8 processors.

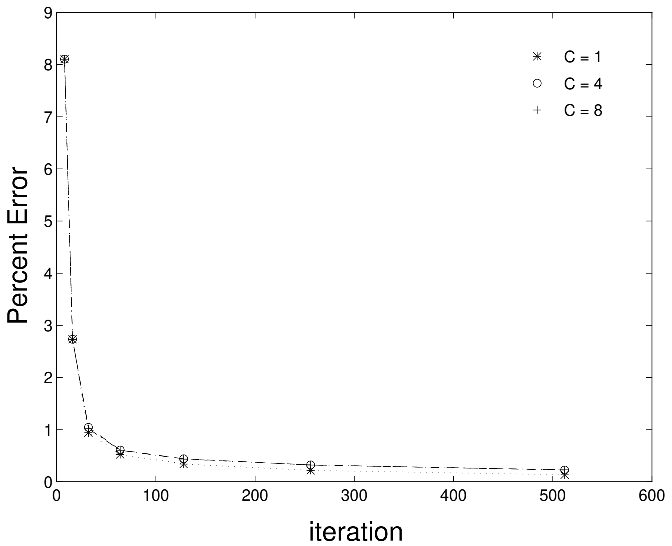
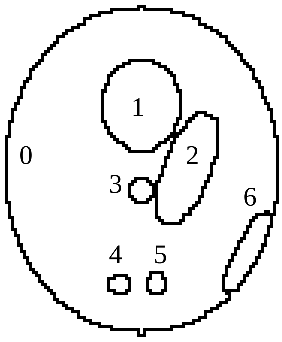


Figure 4.8: Percentage error in phantom reconstruction with respect to original, computed with 16 processors.



ROI	Pixels
0	6287
1	521
2	425
3	37
4	24
5	24
6	140

Figure 4.9: Regions-of-interest (ROIs) and their size in pixels.

phantom, this error in all cases is less than 0.02% throughout the iteration; for the brain scan, less than 0.02%; and for the cardiac scan, less than 0.8%.

The above error measure is global and does not quantify local characteristics of the reconstructions. One way to do so is to partition the simulated phantom into uniform regions-of-interest (ROIs) and compare mean pixel value, standard deviation, and regional bias for different parameter settings [26]; regional bias is the relative difference between the observed and the true mean pixel values of a ROI.

We give values for the seven uniform ROIs of the simulated phantom outlined in Fig. 4.9. Figure 4.10 plots the mean pixel value plus-minus one standard deviation for each of the seven ROIs *vs.* iteration number when 8 processors are used; the results for 16 processors differ only slightly from the results reported here for 8 processors. In Fig. 4.10, for each iteration between 16 and 512 which is a power of 2, there is a grouping of three plots, respectively, for $C = 1$ (left), $C = 4$ (middle) and $C = 8$ (right). The true mean pixel value of each ROI is indicated by a dotted line. The plot for region 5

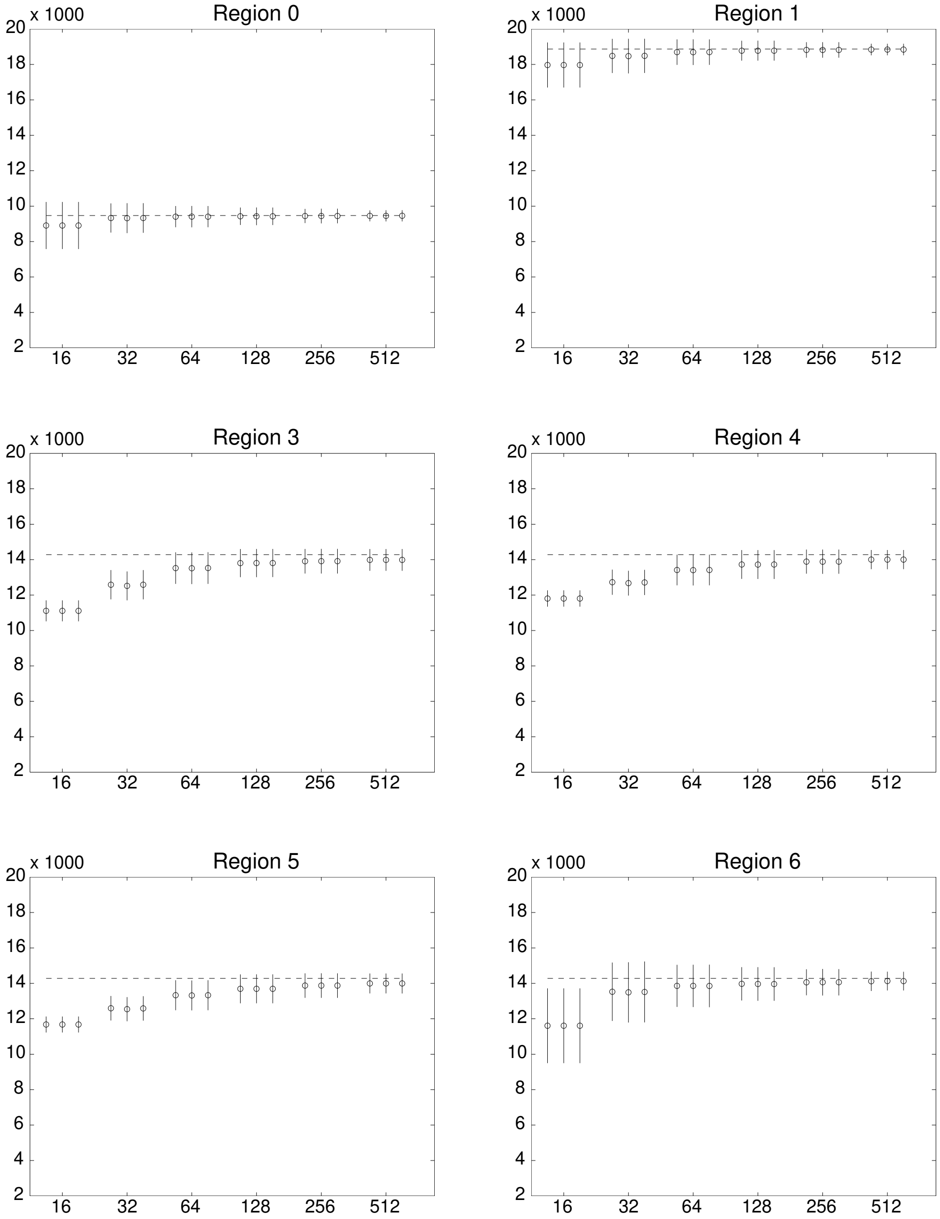


Figure 4.10: Mean pixel value plus-minus one standard deviation for each phantom ROI *vs.* iteration. C=1 (left), C=4 (middle), and C=8 (right).

is almost identical to the plot for region 4 and is therefore not shown.

For each ROI and each value of C, the observed mean pixel value tends to the true value of the simulated phantom; furthermore, the standard deviations are diminishing for ROIs 0, 1, 2, and 6 and nearly constant for ROIs 3, 4, and 5. Over all ROIs and for every iteration, the observed mean pixel values for $C = 4$ and $C = 8$ differ at most by 0.6% from the mean values for $C = 1$. Thus, the regional biases are essentially identical for all three values of C.

An important observation is that the results for $C = 4$ and $C = 8$ are almost indistinguishable from those for $C = 1$ with respect to convergence in terms of the Kullback measure as well as the local and global error measures computed. This indicates that the reduced communications method does not affect the quality of a reconstruction.

4.2.3 Computation Time

The wall-clock time to compute 512 iterations is essentially the same for each of the three sinograms for corresponding settings of network and algorithm parameters. Figure 4.11 shows these times for the sinogram of the brain scan for 8 and 16 processors on the CM–5 and on the Ethernet-connected workstations. Figure 4.11 breaks the wall-clock time down into time spent in computing iterations (i.e., in computing and applying the multipliers), in communicating data, and in idling while waiting to initiate synchronizations.

Wall-clock times on the CM–5 show little dependence on the number of synchronizations because communication is fast among its relatively slower but tightly-coupled

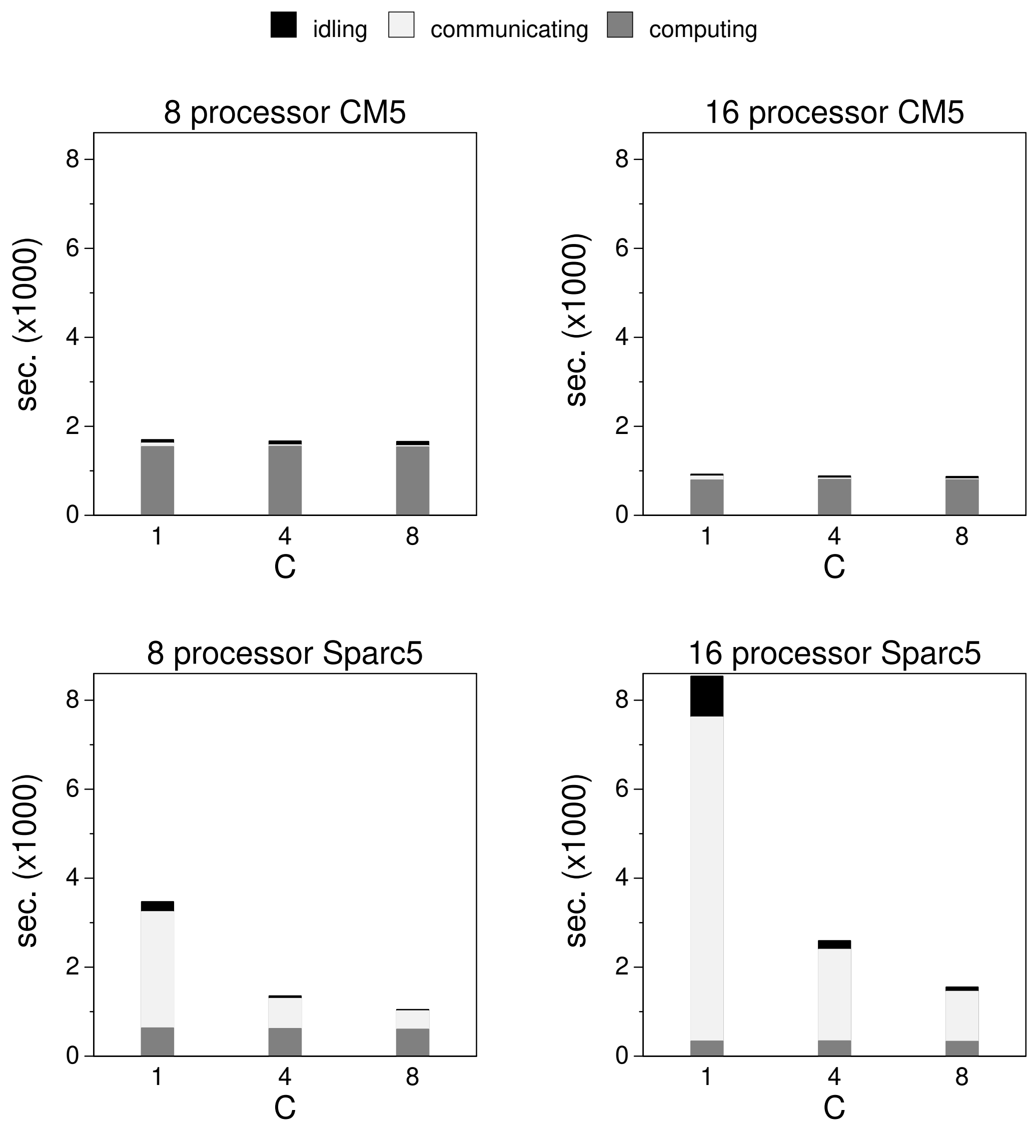


Figure 4.11: Wall-clock time (using brain scan data) for 512 iterations, broken down into time spent idling (waiting to synchronize), communicating, and computing.

processors. 16 CM–5 processors require approximately 55% of the time taken by 8 processors, thus speedup on the CM–5 is near linear. Due to memory constraints we are not able to compute with less that 6 processors.

Wall-clock times for the Sparc5 workstations on the Ethernet are strongly influenced both by the value of C and by the number of processors. Although 16 processors take about 55% the time of 8 to compute and use the multipliers, they also increase the network traffic substantially, and the effective rate of communication deteriorates sharply when the Ethernet is saturated with messages. A contributing factor is that PVM version 3.3.3 uses algorithms that are robust but not tuned for optimality in communications performance on local-area networks [34]. In particular, every call to `pvm_bcast` causes the sending processor to send a copy of the data to each other processor individually. In total, $(\mathcal{P}_n - 1)$ times (size of data vector) bytes are transmitted over the Ethernet for each call to `pvm_bcast` where $\mathcal{P}_n$ is the number of processors involved in the computation.

Communication accounts for about 75%, 50%, and 40% of wall-clock time for 8 Sparc5s for $C = 1$, $C = 4$, and $C = 8$ respectively. Comparable numbers for 16 Sparc5s are 85%, 80%, and 72%. A consequence is that 8 workstations require less wall-clock time than 16 workstations for all three values of C. Using 8 Sparc5s with iteration $C = 8$ yields a wall-clock time comparable to 16 nodes on the CM–5, i.e., an average of less than 2 seconds per iteration over the run of 512 iterations.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This research presents new results concerning convergence and parallel computation with reduced communications for the EM-ML iteration for PET reconstruction introduced by Shepp and Vardi [32] and Lange and Carson [23]. The key results presented in this research include:

- Alternative convergence proofs for the EM-ML algorithm that lead to new results concerning aspects of convergence rate.
- Proof of convergence for EM-ML iteration when computed in parallel with reduced communications.
- Experimental results of a practical implementation of parallel EM-ML iteration with reduced communications showing that communications can be significantly

reduced while maintaining quality of reconstruction.

Convergence. The negative log-likelihood function $\ell(\boldsymbol{\lambda})$ is a convex function for which a minimum is sought given the observed data $\{n_d^*\}$ of positive values and the nonnegativity constraints on $\{\lambda_b\}$. At iteration k this function is

$$\ell(\boldsymbol{\lambda}^k) = f(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) + \mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$$

where $f(\boldsymbol{n}^*, \boldsymbol{\lambda}^*)$ is nonnegative and depends only on the observed data $\boldsymbol{n}^*$. We show that the Kullback function $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ is monotonically decreasing and converges to 0. Thus $\boldsymbol{\lambda}^k$ converges to fixed point $\widehat{\boldsymbol{\lambda}}$ that is a minimizer of $\ell(\boldsymbol{\lambda})$. Furthermore, using a result for convergence of series, we show that the convergence is such that $k\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$. This means that $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) = o(\frac{1}{k})$, i.e., that $\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$ at a faster rate than $k \to \infty$. To further illustrate convergence, we show convergence or nonconvergence of other series associated with the computation.

Parallel computation with reduced communications. We present a result showing that a relaxed version of the EM-ML iteration computed in parallel with reduced communications will converge to an optimal fixed point $\widehat{\boldsymbol{\lambda}}$ if certain assumptions are met. The development is based on results presented by Bertsekas and Tsitsiklis [3]. We show that convergence will occur if the step taken at each iteration is small and if processors communicate often enough.

Experiments. EM–ML iteration with reduced communications on multiple processors can speed up the reconstruction when network, workstation, and algorithm parameters are selected appropriately. For the sinograms and P-matrix in this study, the algorithm for reduced synchronization for $C = 4$ or $C = 8$ distributed over 8 or 16 processors produces reconstructions that are visually close to the fully synchronized reconstructions; this qualitative observation is supported quantitatively by the local and global error measures computed. Convergence as measured by Kullback discrimination is also comparable to fully synchronized computation.

Workstation and network characteristics must be taken into account to prevent adverse trends such as network saturation with messages. When several hundred iterations are computed, a comparatively slow network (the Ethernet) of 8 standard workstations (Sparc5) using generic distributed software (PVM) attains an average time per iteration approaching clinically acceptable rates.

5.2 Future Work

The development of convergence results can be extended. For example, we know that $k\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k}) \to 0$, but we have not concluded whether the series $\sum_k k\mathcal{D}(\boldsymbol{n}^*, \boldsymbol{\lambda}^{*k})$ is convergent or not. The development in section 2.5 of asymptotic ratios of sequences and series gives an introduction to future work concerning convergence results, including further investigation of the impact of $\lambda_b^k \to 0$.

There are several more practical aspects of the parallel computation with reduced communications that can be investigated. One issue is organization of the P-matrix; for example, storing the P-matrix in block-column format will let processors compute in

D-space and communicate the smaller B-space vector. Another aspect is to investigate whether the P-matrix can be reorganized to reduce the dependencies among processors. This could lead to nearest neighbor type communications that exploits network topology.

Bibliography

Bibliography

[1] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M. *Nonlinear Programming Theory and Algorithms.* Wiley, New York, 1993.

[2] Belleman, M. E., Doll, J., Brix, G., Haberkorn, U., Ostertag, H. J., Strauss, L. G., Dimitrakopoulou-Strauss, A., and Lorenz, W. J. PET image reconstruction for pharmacokinetic modeling of $5-[^{18}\mathrm{F}]$fluorouracil in liver tumors. In H. Bergmann, and H. Sinzinger, editor, *Radioactive isotopes in clinical medicine and research*, Basel, 1995. Birkhäuser Verlag.

[3] Bertsekas, D. P. and Tsitsiklis, J. N. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Englewood Cliffs, N.J., 1989.

[4] Butler, C. S. and Miller, M. I. Maximum a posteriori estimation for SPECT using regularization techniques on massively parallel computers. *IEEE Trans. Med. Imaging*, 12:84–89, 1993.

[5] Chen, C. M., Lee, S. Y., and Cho, Z. H. Parallelization of the EM algorithm for 3-D PET image reconstruction. *IEEE Trans. Med. Imaging*, 10:513–522, 1991.

[6] Csiszár, I. and Kórner, J. *Information Theory Coding Theorems for Discrete Memoryless Systems.* Academic Press. Inc, New York, 1981.

[7] De Pierro, A. R. Multiplicative iterative methods in computed tomography. In Herman, G. T., Louis, A. K., and Natterer, F., editors, *Mathematical Methods in Tomography*, pages 167–186. Springer-Verlag, Berlin, 1991.

[8] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc.*, 39:1–38, 1977.

[9] Doll, J., Ostertag, H. J., Bellemann, M. E., Schmidlin, P., Kübler, W. K., Strauss, L. G., and Lorenz, W. J. Effects of distorted PET projection data on the reconstructed image using different reconstruction algorithms. In Bergmann, H. and Sinzinger, H., editors, *Radioactive isotopes in clinical medicine and research*, Basel, 1995. Birkhäuser Verlag.

[10] Feller, W. *An introduction to probability theory and its applications*, volume 1. John Wiley & Sons, Inc, 1968.

[11] Geist, G. A., Beguelin, A., Dongarra, J., Jiang, W., and Manchek, R. PVM 3 user's guide and reference manual. Technical Report ORNL/TM-12187, Oak Ridge National Laboratory, 1994.

[12] Hebert, T. and Leahy, R. A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. *IEEE Trans. Med. Imaging*, 8:194–202, 1989.

[13] Herman, G. T., Odhner, D., Toennies, K. D., and Zenios, S. A. A parallelized algorithm for image reconstruction from noisy projections. In T. Coleman, and Y. Li, editor, *Workshop Large Scale Optimization*, Philadelphia, 1990. SIAM.

[14] Johnson, V. E., Wong, W. H., Hu, X., and Chen, C. T. Bayesian restoration of PET images using Gibbs priors. In *Information processing in medical imaging*, pages 15–28, New York, 1991. IPMI89, Wiley-Liss.

[15] Jones, W., Byars, L. G., and Casey, M. Positron emission tomography images and expectation maximization: a VLSI architecture for multiple iterations per second. *IEEE Trans. Nucl. Sci.*, 35:620–624, 1988.

[16] Kaufman, L. Implementing and accelerating the EM algorithm for positron emission tomography. *IEEE Trans. Med. Imaging*, 6:37–50, 1987.

[17] Kaufman, L. Solving emission tomography problems on vector machines. *Annals Oper. Res.*, 22:325–353, 1990.

[18] Kaufman, L. Maximum likelihood, least squares, and penalized least squares for PET. *IEEE Trans. Med. Imaging*, 12:200–214, 1993.

[19] Kawata, S. and Nalcioglu, P. Constrained reconstruction by the conjugate gradient method. *IEEE Trans. Med. Imaging*, 4:65–71, 1985.

[20] Knopp, K. *Theory and Application of Infinite Series.* Blackie and Son Limited, London and Glasgow, 1951.

[21] Kullback, S. *Information Theory and Statistics.* John Wiley and Sons, New York, 1958.

[22] Kumar, V., Grama, A., Gupta, A., and Karypis, G. *Introduction to Parallel Computing: design and analysis of parallel algorithms.* The Benjamin/Cummings Publishing Company, Inc, Redwood City, CA 94065, 1994.

[23] Lange, K. and Carson, R. EM reconstruction algorithms for emission and transmission tomography. *J. Comput. Assisted Tomography*, 8:306–316, 1984.

[24] Levitan, E. and Herman, G. T. A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography. *IEEE Trans. Med. Imaging*, 6:185–192, 1987.

[25] Lewitt, R. M. and Muehllehner, G. Accelerated iterative reconstruction for positron emission tomography based on the EM algorithm for maximum likelihood estimation. *IEEE Trans. Med. Imaging*, 5:16–22, 1986.

[26] Llacer, J., Veklerov, E., Coakley, K. J., Hoffman, E. J., and Nunez, J. Statistical analysis of maximum likelihood estimator images of human brain FDG PET studies. *IEEE Trans. Med. Imaging*, 12:215–231, 1993.

[27] McCarthy, A. W. and Miller, M. I. Maximum likelihood SPECT in clinical computation times using mesh-connected parallel computers. *IEEE Trans. Med. Imaging*, 10:426–436, 1991.

[28] Miller, M. I. and Roysam, B. Bayesian image reconstruction for emission tomography incorporating Good's roughness prior on massively parallel processors. *Proc. Natl. Acad. Sci. USA.*, 88:3223–3227, 1991.

[29] Miller, T. R. and Wallis, J. W. Clinically important characteristics of maximum-likelihood reconstruction. *J. Nucl. Med.*, 33:1678–1684, 1992.

[30] Raichle, M. E. Visualizing the mind. *Scientific American*, 270(4):58–64, April 1994.

[31] Ranganath, M. V., Dhawan, A. P., and Mullani, N. A multigrid expectation maximazation reconstruction algorithm for positron emission tomography. *IEEE Trans. Med. Imaging*, 7:273–278, 1988.

[32] Shepp, L. A. and Vardi, Y. Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1:113–122, 1982.

[33] Snyder, D. L. and Miller, M. J. The use of sieves to stabilize images produced with the EM algorithm for emission tomography. *IEEE Trans. Nucl. Sci.*, 32:3864–3870, 1985.

[34] Sunderam, V. S., Geist, G. A., Dongarra, J., and Manchek, R. The PVM concurrent computing system: evolution, experiences, and trends. *Parallel Computing*, 20:531–547, 1994.

[35] Thinking Machines Corporation, Cambridge, MA. *CMMD reference manual*, May 1993.

[36] Townsend, D. W. and Defrise, M. Image reconstruction methods in positron tomography. Technical Report 93-02, CERN, 1993. ISBN 92-9083-051-4.

[37] Tsui, B., Zhao, X., Frey, E., and Gullberg, G. Comparison between ML-EM and WLS-CG algorithms for SPECT image reconstruction. *IEEE Trans. Nucl. Sci.*, 38:1766–1772, 1991.

[38] Vardi, Y., Shepp, L. A., and Kaufman, L. A statistical model for positron emission tomography. *J. Amer. Statis. Assoc.*, 80:8–20, 1985.

[39] Wallis, J. W. and Miller, T. R. Rapidly converging iterative reconstruction algorithms in Single-Photon Emission Computed Tomography. *J. Nucl. Med.*, 34:1793–1800, 1993.

Vita

Soeren Peter Olesen was born on July 1, 1965 in Roskilde, Denmark. He graduated from Himmelev Gymnasium (High School) in June of 1984. He then entered the Danish Engineering Academy and graduated in January of 1988 with a Bachelor's degree in Electrical Engineering. He then worked 3 years as software engineer and later as software manager in Danish industry. He entered the Ph.D program in Computer Science at The University of Tennessee, Knoxville in January of 1991, earned the Master of Science in May of 1993 and the Doctor of Philosophy in May of 1996.