5/27/2008
(HYPER)

Barret & Meyer Book.

the out-of-band ones alone.

Positivity and support constraints cannot be implemented readily on Fourier coefficients, so transformation back to the space domain is still required. The exact transformation rule is

$$\hat{f}(\mathbf{r}) = \sum_{\mathbf{k}} \hat{F}_{\mathbf{k}} \exp(2\pi i \boldsymbol{\rho}_{\mathbf{k}} \cdot \mathbf{r}) \,, \tag{15.295}$$

where the sum is, in principle, infinite in two dimensions. In practice, of course, a finite sum must be used, with the range determined by the amount of superresolution desired. To transform back to Fourier coefficients, the exact equation is

$$\hat{\mathbf{F}}_{\mathbf{k}} = \frac{1}{L^2} \int_{\mathbf{S_f}} d^2r \; \hat{f}(\mathbf{r}) \exp(-2\pi i \boldsymbol{\rho}_{\mathbf{k}} \cdot \mathbf{r}) \,, \tag{15.296}$$

where the object support $\mathbf{S_f}$ is assumed to be a square of side L. In practice the integral would be implemented as a sum, most likely as a DFT (though one with many more than M elements).

15.4.6 MLEM algorithm

To estimate the Coefficients

An important iterative technique is the *maximum-likelihood expectation-maximization* or *MLEM* algorithm, so called because it can be derived by alternating expectation (E) and maximization (M) steps, and because it maximizes the likelihood for a Poisson data model. MLEM has been rediscovered several times. To the authors' knowledge, the earliest paper to present the algorithm was by Metz and Pizer (1971) at the second international conference on Information Processing in Medical Imaging (IPMI). Unfortunately, the untimely death of the conference organizer, Eberhard Jahns, led to the promised Proceedings of IPMI II never appearing and thus the Metz and Pizer paper never being published.

In the optics literature, the MLEM algorithm was presented independently by Richardson (1972) and Lucy (1974), and it is still referred to often as the Richardson-Lucy algorithm. The paper that ignited widespread interest for tomographic applications was by Shepp and Vardi (1982). Another important early contribution to the tomographic literature was Lange and Carson (1984).

MLEM is but one example of a broad class of algorithms that alternate expectation and maximization steps. A rigorous treatment of these more general EM algorithms was given in an important paper by Dempster, Laird and Rubin (1977), and an excellent monograph on the subject is McLachlan and Krishnan (1997). In this section, however, we consider only the MLEM algorithm.

MLEM as a multiplicative algorithm The iteration rule for the basic MLEM algorithm is

$$\hat{\theta}_n^{(k+1)} = \hat{\theta}_n^{(k)} \frac{1}{s_n} \sum_{m=0}^{M} \frac{g_m}{(\mathbf{H}\hat{\boldsymbol{\theta}}^{(k)})_m} H_{mn} \,, \tag{15.297}$$

where s_n is the n^{th} component of the point sensitivity vector, defined in (7.312) as

$$s_n = \sum_{m=0}^{M} H_{mn} \,. \tag{15.298}$$

To interpret s_n, consider the usual voxel description of a source where θ_n is the mean number of photons emitted from the n^{th} voxel and H_{mn} is the probability that a photon from voxel n is detected in detector m. Then $s_n\theta_n$ is the mean number of photons from that voxel detected by all detectors, and s_n is the probability that a photon emitted from voxel n is detected somewhere. We need not worry about dividing by zero in (15.297) since voxels with $s_n = 0$ have zero probability of ever contributing to the data and should not be included in the representation in the first place.

Unlike the linear algorithms discussed in Sec. 15.4.1 and the modified linear algorithms discussed in Sec. 15.4.4, MLEM is a *multiplicative algorithm* where an estimate is modified by multiplying it by a correction factor rather than adding a correction term. Other important examples of multiplicative algorithms include multiplicative ART or MART (Gordon *et al.*, 1970) and its variants (Byrne, 1993; 1995) and the SAGE (space-alternating generalized EM) algorithms (Fessler and Hero, 1994), but we shall not discuss any of these methods further.

The MLEM algorithm preserves positivity; that is, if the initial estimate $\hat{\boldsymbol{\theta}}^{(0)}$ is nonnegative, and if all elements of $\mathbf{g}$ and $\mathbf{H}$ are nonnegative, then all subsequent iterations remain nonnegative since we always multiply by a nonnegative factor. By the same token, however, a component of the estimate will seldom be driven exactly to zero; if H_{mn} is nonzero for *any* m for which g_m is nonzero, then the correction factor for $\hat{\theta}_n^{(k)}$ will not be zero. In this respect, the MLEM algorithm is like maximum entropy in that it tends to drive the estimate towards zero but never quite gets there. The exception would be if $g_m = 0$ for all detectors for which $H_{mn} \neq 0$ for a given n; in that case $\hat{\theta}_n^{(k)}$ would be immediately set to zero.

If we know the support of the object *a priori*, on the other hand, we can set the elements of the estimate (in a voxel representation) outside the support to zero in the initial estimate and they will remain zero for all subsequent iterations.

Finally, note that the algorithm strives for agreement between the actual data and the image of the estimate. If $(\mathbf{H}\hat{\boldsymbol{\theta}}^{(k)})_m = g_m$ for all m, then the correction factor is unity and no further change in the estimate occurs. Of course, it may not be possible to find an estimate such that $(\mathbf{H}\hat{\boldsymbol{\theta}}^{(k)})_m = g_m$ for all m, and in that case it turns out (as we shall see below) that the algorithm converges to an estimate that minimizes the Kullback-Leibler distance (see Sec. 15.3.2) $D_{\mathrm{KL}}(\mathbf{g}, \mathbf{H}\hat{\boldsymbol{\theta}})$ between the data and the image of the estimate.

Poisson likelihood As presented so far, the MLEM algorithm is just a convenient way of finding an estimate that agrees as well as possible (in the Kullback-Leibler sense) with the data. It has no particular relation to the statistics of the data and in fact will work with many different kinds of data. We know from Sec. 15.3.2, however, that the Kullback-Leibler distance is closely related to the log-likelihood for Poisson data, and we shall now explore this relation further.

If we consider an MD Poisson random vector $\mathbf{g}$ with mean $\mathbf{H}\boldsymbol{\theta}$, then

$$\Pr(\mathbf{g}|\boldsymbol{\theta}) = \prod_{m=1}^{M} \exp[-(\mathbf{H}\boldsymbol{\theta})_m] \frac{[(\mathbf{H}\boldsymbol{\theta})_m]^{g_m}}{g_m!} , \tag{15.299}$$

and $\Pr(\mathbf{g}|\boldsymbol{\theta})$ is the likelihood of $\boldsymbol{\theta}$ for a given $\mathbf{g}$. One must view this equation with caution, however, since we are free to choose any representation we like for the object, with any number of components N. It is only when the finite object representation is an adequate representation of the data, in the sense that $\mathbf{H}\boldsymbol{\theta}$ is a good approximation to $\boldsymbol{\mathcal{H}}\mathbf{f}$, that $\Pr(\mathbf{g}|\boldsymbol{\theta})$ is really the likelihood of $\boldsymbol{\theta}$. [See (15.137) and the associated discussion.]

With this caveat, the logarithm of the likelihood is given by (15.149), repeated here for convenience:

$$\ln[\Pr(\mathbf{g}|\boldsymbol{\theta})] = \sum_{m=1}^{M} \{-(\mathbf{H}\boldsymbol{\theta})_m + g_m \ln[(\mathbf{H}\boldsymbol{\theta})_m] - \ln(g_m!)\} . \qquad (15.300)$$

An extremum of this function occurs at a point where the derivative with respect to all components vanishes:

$$\frac{\partial}{\partial\theta_j} \ln[\Pr(\mathbf{g}|\boldsymbol{\theta})] = \sum_{m=0}^{M} \left\{ -H_{mj} + \frac{g_m}{(\mathbf{H}\boldsymbol{\theta})_m} H_{mj} \right\} = 0 , \qquad j = 1, ..., N . \qquad (15.301)$$

To see whether the extremum is a minimum or a maximum, we take another derivative:

$$\frac{\partial^2}{\partial\theta_j \partial\theta_k} \ln[\Pr(\mathbf{g}|\boldsymbol{\theta})] = \sum_{m=0}^{M} \left\{ -\frac{g_m}{[(\mathbf{H}\boldsymbol{\theta})_m]^2} H_{mj} H_{mk} \right\} . \qquad (15.302)$$

All components of $\mathbf{g}$ and $\mathbf{H}\boldsymbol{\theta}$ must be nonnegative for the Poisson law to be applicable; a negative number of counts makes no sense. Moreover, all elements of $\mathbf{H}$ must be nonnegative, since otherwise a negative component of $\mathbf{H}\boldsymbol{\theta}$ could occur for some nonnegative $\boldsymbol{\theta}$. Thus the second derivative is negative everywhere (*i.e.*, the log-likelihood is concave), and any extremum must be a maximum. Maximizing the likelihood is thus equivalent to solving the implicit equation (15.301) for $\boldsymbol{\theta}$. Moreover, from the discussion in Sec. 15.3.2, it is also equivalent to minimizing the Kullback-Leibler distance $D_{\mathrm{KL}}(\mathbf{g}, \mathbf{H}\boldsymbol{\theta})$.

MLEM as a fixed-point iteration We can rewrite (15.301) (with the dummy index j changed to n) as

$$\frac{1}{s_n} \sum_{m=0}^{M} \frac{g_m}{(\mathbf{H}\boldsymbol{\theta})_m} H_{mn} = 1 , \qquad (15.303)$$

where s_n is defined in (15.298). We now multiply both sides of (15.303) by θ_n, yielding

$$\theta_n = \theta_n \frac{1}{s_n} \sum_{m=0}^{M} \frac{g_m}{(\mathbf{H}\boldsymbol{\theta})_m} H_{mn} . \qquad (15.304)$$

To get an iterative algorithm, we replace $\boldsymbol{\theta}$ by a succession of estimates $\hat{\boldsymbol{\theta}}^{(k)}$ and use the fixed-point iteration procedure introduced in Sec. 15.4.4; the result is the MLEM algorithm:

$$\hat{\theta}_n^{(k+1)} = \hat{\theta}_n^{(k)} \frac{1}{s_n} \sum_{m=0}^{M} \frac{g_m}{(\mathbf{H}\hat{\boldsymbol{\theta}}^{(k)})_m} H_{mn} . \qquad (15.305)$$

If this algorithm converges, it must find a maximum of the log-likelihood (and hence of the likelihood itself).

Convergence and stopping rules The mapping defined by (15.305) is not a contraction and hence does not necessarily converge to a fixed point independent of the initial estimate. It can, however, be shown to converge in the sense that the likelihood increases monotonically at each step (Dempster *et al.*, 1977). Of course, just because the algorithm approaches a specified likelihood (the maximum value) does not mean it approaches a specified image. The likelihood is a function of $\mathbf{H}\boldsymbol{\theta}$, not $\boldsymbol{\theta}$ alone. If $\mathbf{H}$ has null functions, many different $\boldsymbol{\theta}$ can give the same $\mathbf{H}\boldsymbol{\theta}$ and hence the same likelihood; which one is obtained by the algorithm depends on the null components of the initial estimate.

Moreover, maximum likelihood is seldom a desirable end point in image reconstruction. As we have stressed repeatedly, forcing agreement with noisy data (in any sense) results in noisy images. In practice, running the MLEM algorithm for a large number of iterations usually results in a virtually useless image, often one consisting of a few bright pixels like the night-sky reconstructions discussed in Sec. 15.3.5. (For an example, see Fig. 17.9.)

The most common way of avoiding these problems is just to stop the algorithm before it gives a poor image in some sense. In this case, the image is not a maximum-likelihood estimate, and it depends on the number of iterations and the initial estimate. The choice of stopping point is usually made purely subjectively, though various statistical stopping rules have been proposed (see, for example, Llacer and Veklerov, 1989) as a means of avoiding excessive noise amplification. The stopping point should ideally be chosen to optimize some objective measure of image quality, such as the ability of a human observer to detect an abnormality (see Sec. 14.2), but in practice it is usually done without regard to task.

15.4.7 Noise propagation in nonlinear algorithms

We have already discussed the noise properties of reconstructed images in several cases in this chapter. In Sec. 15.2.6 we treated the effect of a linear reconstruction operator on noise in the data, and in Sec. 15.3.6 we considered implicit estimates and found that we could get useful analytical forms for the covariance without specifying the algorithm for actually finding the estimate. Then, in Sec. 15.4.2 we studied noise propagation through linear iterative algorithms. Now we shall show how that analysis needs to be modified for nonlinear iterative algorithms. The main difference will turn out to be that constant matrices are replaced by ones that depend on the current estimate.

Differentiable update rules All of the iterative algorithms considered in this chapter have the general form,

$$\hat{\boldsymbol{\theta}}^{(k+1)} = \mathbf{D}^{(k)}\{\hat{\boldsymbol{\theta}}^{(k)}, \mathbf{g}\}, \tag{15.306}$$

where $\mathbf{D}^{(k)}\{\cdot\,,\,\cdot\}$ is a vector-valued functional of its two arguments. The fixed-point iteration of (15.280) is immediately in this form, and POCS, MLEM and conjugate-gradient fit as well.

It will be very useful to assume that $\mathbf{D}^{(k)}\{\cdot\,,\,\cdot\}$ is differentiable with respect to both arguments. That assumption is justified by inspection for MLEM, conjugate-gradient and many other algorithms, but it may not hold for POCS or other algorithms that employ a clipping operator such as $\mathbf{P}_+$ as defined in (15.186). To get around this difficulty, we can redefine $\mathbf{P}_+\{x\}$ as the limit of a differentiable functional, for example,

$$\mathbf{P}_+\{x\} = x\,\mathrm{step}(x) = \lim_{\beta\to\infty} \frac{x}{1-\exp(-\beta x)}\,. \tag{15.307}$$