Continuous-Detector List-Mode MLEM

All the literature assumes discretized detectors and lots of histograms all over the place. This doesn't make much sense in the case of a continuous LXe detector. Can we avoid binning? If so, how?

1 List-mode MLEM with binned LORs

Equation (13.68) from (Bailey et al., 2014) expresses list-mode MLEM (using slightly different notation) as:

$$\lambda_j^{k+1} = \frac{\lambda_j^k}{\sum\limits_{i\in L} A_{ij}} \sum_{i\in M} A_{ij} \frac{1}{\bar{b}_i + \sum\limits_{j\in V} A_{ij}\lambda_j^k} \tag{1}$$

- λ_j^k: activity in voxel j of reconstructed image after k iterations
- A_{ij}: the *system matrix* whose elements are the probabilities that an annihilation in voxel j is observed in LOR i
- L: the set of all LORs that could potentially be observed by the detector
- M: the set of *measured* LORs, those that *have* been observed in the current data acquisition
- V: the set of voxels in the reconstructed image, or field of view (FOV)
- $\bar{b}_i$: estimation of scatter and random events observed in LOR i

A number of factors contribute to A_{ij}:

- geometry: the intersections of the LORs with the voxels
- attenuation
- detector sensitivity
- etc.
- etc.

The system matrix can also be viewed as a function

$$A(i,j) : L \times V \to \text{Probability} \tag{2}$$

quantifying the coupling between voxels (image space) and LORs (observation space).

Equation (1) can be broken down into the following components:

- The *forward projection* of the image onto LOR i

  $$F_i = \bar{b}_i + \sum_{j\in V} A_{ij}\lambda_j^k \tag{3}$$

  is the expectation value of observing LOR i, given the current (after iteration k) reconstructed numbers of annihilations having taken place in each voxel in the FOV, and taking into account an estimate of additive contributions from scatters and randoms, $\bar{b}_i$.
- The *backward projection* of LORs into image space

  $$B_j = \sum_{i\in M} \frac{A_{ij}}{F_i} \tag{4}$$

  distributes the observed signal among the image voxels in which the signal is most likely to have originated.
- The *sensitivity image*

  $$S_j = \sum_{i\in L} A_{ij} \tag{5}$$

  is the probability that an annihilation in voxel j is detected at all.

It is important to note that the sum in the backprojection, B_j, considers only the observed LORs, $i \in M$, while the sum in the sensitivity image, S_j, takes into account all potentially observable LORs, $i \in L$.

Equation (1) can be expressed in terms of these components as

$$\lambda_j^{k+1} = \frac{\lambda_j^k}{S_j} B_j \tag{6}$$

However, all that precedes assumes that there is a finite number of observable LORs, and that each of these LORs occupies a finite volume in LOR-space.

2 Continuous detector

In our LXe detector the LOR-segment endpoints are not binned into regions covered by finitely-sized detector crystals, instead they can fall on continuously distributed points within the LXe. This means that the number of potential LORs is infinite, and that each LOR occupies an infinitesimal volume in LOR-space.

Consequently, it no longer makes sense to think of A_{ij} as a matrix, leaving the functional perspective as the only sensible one:

$$A(l,j) : L \times V \to \text{Probability density} \tag{7}$$

The crucial differences between functions (2) and (7) are that, in the latter

- L contains an infinite number of infinitesimally-sized LORs, as opposed to a finite number of finitely-sized LORs.
- The codomain is probability density (PD), rather than probability.

Equations (3–5) need to be adapted to account for these differences, becoming

$$F(l) = \bar{b}(l) + \sum_{j \in V} A(l,j)\lambda_j \tag{8}$$

$$B_j = \sum_{l \in M} \frac{A(l,j)}{F(l)} \tag{9}$$

$$S_j = \int_L A(l,v)\, \mathrm{d}L \tag{10}$$

In doing so, a number of components which were originally expressed in terms of a finite number of discrete values are replaced with analogues with continuous domains:

$$\begin{aligned} A_{ij} &\to A(l,j) \\ F_i &\to F(l) \\ \bar{b}_i &\to \bar{b}(l) \\ S_j : \sum_{i \in L} A_{ij} &\to \int_L A(l,v)\, \mathrm{d}L \end{aligned}$$

We need to understand how these continuous versions can be calculated in the computer code that implements the reconstruction.

3 The core problem

How can we express the $A(l,j)$ and $\bar{b}(l)$ terms in equation (8) as probability (or expectation) density functions (PDFs)?

In the code we have tried to histogram trues and scatters, in order to deduce the scatter-to-true ratio for LORs in various bins. The hope is that this PDF varies gently, and can thus be approximated adequately with fairly coarse-grained (and therefore cheap) histograms. Given this binning, the absolute counts are trivially turned into densities by dividing by the bin size.

If we can somehow turn A into a PDF, then, surely, the problem is solved!

Components of A such as the attenuation factors, are multiplicative contributions applied to the fundamental geometric part of $A(l,j)$. The latter is calculated by measuring the length of the segment of LOR l lying within voxel j. **How can this length be converted into a probability density?** Some ideas are explored in Section 7.

4 Automatic normalization and sensitivity image estimation

4.1 Back-projection normalizes

The dimensions of A and F cancel in equations (4) and (9), so the back-projection eliminates any arbitrary scale factors and removes the need for correct normalization of A …were it not for the presence of $\bar{b}$ in equations (3) and (8). So correct normalization is not necessary if A and $\bar{b}$ are expressed on compatible scales (see Section 9), or if $\bar{b}$ is entirely absent …were it not for the presence of A in the sensitivity image, (5) and (10). But …

4.2 Sensitivity image is not evaluated directly

…the sensitivity image, S_j, is not calculated directly as implied by (5) and (10). It is estimated by other means. Which, again absolves us from the need to normalize A correctly, **as long as it is compatible with $\bar{b}$**. Which is why we could ignore this whole issue as long as we didn't try to correct for scatter or randoms.

4.3 Sensitivity estimation

At present, the code completely ignores S_j (it is assumed to be 1 everywhere), unless the attenuation correction option is enabled, in which case a pre-computed sensitivity image is used. This image is constructed by selecting random LORs which have both ends on the detection surface and which pass through the FOV. We forward-project the attenuation image onto these LORs and back-project them onto a blank image, to build the sensitivity image.

If the sensitivity image is constructed with few LORs, it is very noisy, and this noise is introduced into the image reconstructed by MLEM. It takes 10^9 LORs to get a smooth image for a FOV with $235 \times 235 \times 215 \approx 10^7$ voxels, which requires about 2 hours of computation, in serial mode. My guesstimate is that parallelization would speed this up by a factor of 3-ish.

Quantity	Dimension	Interpretation
λ_j	1	Number of annihilations
$A(l,j)$	$\mathrm{d}L^{-1}$	Probability density
$\bar{b}(l)$	$\mathrm{d}L^{-1}$	Expectation density
$F(l)$	$\mathrm{d}L^{-1}$	Expectation density
B_j	1	Inverse expectation
S_j	1	Probability

Table 1: *Dimensions for continuous LOR-space*

Quantity	Dimension	Interpretation
A_{ij}	1	Probability
$\bar{b}_i$	1	Expectation
F_i	1	Expectation

Table 2: *Dimensions for discrete LOR-space*

Can we do better by generating N LORs, perhaps uniformly distributed across solid angle, from the centre of each voxel? Some of these LORs will not hit the detector at both ends: can this be used to incorporate the angular acceptance part of the sensitivity image?

5 Dimensional Analysis

Ignoring time of flight (TOF) for the time being (until Section 6), processing the data acquired by the detector yields pairs of 3-dimensional spatial coordinates which define segments of the (infinitely long) LORs being observed. Knowledge of where a segment starts and ends, adds no useful information over that conveyed by the whole LOR, meaning that two of these six degrees of freedom are superfluous[1]. Thus the set of all observable LORs, L, is a subset of $\mathbb{R}^4$. Consequently the $\mathrm{d}L$ appearing in (10) is 4-dimensional[2].

The dimensions of various quantities appearing in equations (3–5) are set out in table 1, which should be contrasted with the analogous quantities appearing in equations (8–10), whose dimensions are shown in table 2.

6 TOF

No need to bin TOF info. This is an advantage of list mode wrt sinogram-based methods. But the TOF gaussian returns a probability density with units Length^{-1} (or Time^{-1}). It's not immediately obvious where to get a Length (or Time) which would turn this back into a ratio.

7 Consider a spherical cow

The geometrical components of the system matrix, A, are found by a Siddon-like algorithm (Siddon, 1985), which calculates the lengths of LOR-segments within each voxel, so these have units such as millimetres. But table 1 shows that A should have units of probability density in 4-dimensional LOR-space. Therefore the key question is, **how can we convert these millimetres into probability densities in LOR-space?** In the rest of this section I grope around in the dark, hoping to get closer to some sort of answer.

Consider an hermetic detector with a 100% efficient, infinitesimally thin detecting surface. LOR l is defined by the two points l_1 and l_2 at which it crosses the detection surface. By considering single points, rather than entire voxels, we relate $A(l,j)$ to $A(l,p) = A(l_1,l_2,p)$, by

$$A(l,j) = \int_{p\in j} A(l,p)\,\mathrm{d}p \qquad (11)$$

where p is a single point lying within the FOV, and $A(\ldots,p)$ is the probability that an annihilation at point p is detected in LOR l.

The hermeticity and 100% efficiency[3] of the detector guarantee that

$$N(p) = \int A(l_1,l_2,p)\,\mathrm{d}l_1\,\mathrm{d}l_2 = 1$$

where $N(p)$ (the pointwise equivalent of the voxel-oriented S_j in equation (10)) is the probability that an annihilation at point p be detected by the detector. We note that $A(l_1,l_2,p)$ is zero wherever l_1, l_2 and p are not collinear, so A can be factored into

$$A(l_1,l_2,p) = c(l_1,l_2,p)\cdot a(l_1,l_2,p) \qquad (12)$$

where c is a Dirac delta-like collinearity function, and a takes care of any positional dependence there might me . . . is there any? There must be some in the dls, at least, so a must compensate for that.

The $\mathrm{d}l_{1,2}$ are 2-dimensional. It might be helpful to try to build intuition in a simplified, 2-dimensional model where the detecting surface is a circle, centred around a square FOV made up of square voxels. In which case the $\mathrm{d}l_{1,2}$ would be 1-dimensional.

What's the point of all this? We're trying to understand the A in (8) as a probability density function, specifically how the length of the LOR segment in the voxel gives rise to a probability density.

[1] So we should probably take advantage of this to reduce the size of the HDF5 files containing LORs.

[2] How should we parametrize these 4 dimensions? Sinograms would typically be parametrized with something like

- z: z-coordinate of LOR's point of nearest approach to z-axis.
- $|\vec{s}|$: distance between LOR and z-axis.
- ϕ: orientation of $\vec{s}$
- $\tan\theta$: obliqueness of LOR.

This would then probably match the binning scheme we would use when estimating additive corrections.

[3] Deviations from these ideal conditions are encoded in the other components of A: these ideals are correct from the perspective of the purely geometric part of A.

The colinearity component of A makes it possible to evaluate the inegral in equation (11) as a line integral along LOR l inside voxel v. For a fixed LOR l, $A(\ldots, p \in l)$ should be a constant (really?[4]), so this integral amounts to multiplication of A by the aforementioned length of the LOR segment in v.

What are the dimensions of A, c and a in equation (12)? Do these combine with the length-multiplication in the last paragraph, to give us the density we want?

8 Can we discretize arbitrarily?

Discretized 3D detectors with long axes, have a *staggering* number of potential LORs (especially if highly oblique LORs are not discarded, as is our aim). This number is so large that it significantly exceeds the number of observed LORs in a data acquisition. Consequently, sinograms (histograms) into which the observed LORs would be binned, would be very sparse. For such large detectors, list mode (among other advantages) is a much more memory-efficient way of storing the LORs, than sinograms.

Why can't we store our LORs in list mode, and pretend that they occupy some finite, discretized space, in order to solve the problem of the probably density, discussed in earlier sections?

I have various half-formed arguments in my mind ...watch this space.

8.1 Aside

The histograms of scatters and trues which we use to estimate the scatter contribution, must be very coarse-grained; otherwise they would be sparse and therefore completely useless for the purpose of calculating scatter-to-true ratios. These histograms have known bin sizes, and therefore could give us true and scatter counts per $\mathrm{d}L$.

9 What's wrong with the scatter-per-trues trick?

$\overline{b}$ has contributions from scatter, $\overline{s}$, and random, $\overline{r}$, events:

$$\overline{b}(l) = \overline{s}(l) + \overline{r}(l)$$
$$\overline{b}_i = \overline{s}_i + \overline{r}_i$$

Ignoring randoms, for the time being, we generate histograms of scatters, $\overline{s}_i$, and trues, $\overline{t}_i$, in order to estimate the scatter contributions in arbitrary LOR bins i. These bins are arbitrary in the sense that they bear no relation to any binning imposed by the detector itself.

[4]Whatever variation there is in the non-colinearity component of A, should be depend on the angle between l and the detection surface, and this does not vary for a fixed LOR.

We can turn these two histograms of absolute counts, into an estimate of the relative additive contribution *per observed true event*:

$$\overline{\beta}_i = \frac{\overline{b}_i + \overline{t}_i}{\overline{t}_i} = \frac{\overline{b}_i}{\overline{t}_i} + 1 \tag{13}$$

for observations in the region of LOR-space occupied by bin i. The division in equation (13) cancels the units of $\overline{b}$ and $\overline{t}$ ensuring that $\overline{\beta}$ is dimensionless. So, if there is an adequate equivalence between $\overline{t}_i$ and $\sum_{j \in V} A(l,j)\lambda_j$, equation (8) can be rewritten as

$$F(l) = \overline{\beta}_{i(l)} \sum_{j \in V} A(l,j)\lambda_j$$

where $i(l)$ is the bin containing LOR l (or we could smooth by interpolating $\overline{\beta}$ between neighbouring bins). In which case, according to the arguments in Section 4.1 the problem should be solved.

Are there insurmountable problems around the link between $\overline{t}_i$ and $\sum_{j \in V} A(l,j)\lambda_j$?

- $\sum_{j \in V} A(l,j)\lambda_j$ is not a count of observations: it's a projection, an expectation value based on the—initially completely incorrect—reconstructed distribution of activity, which will be very different from 1 in the early iterations, but should converge to 1 in later iterations. In list mode, by definition, every observation is in its own LOR 'bin' that has a count of 1. So perhaps the correct version is

  $$F(l) = \overline{\beta}_{i(l)} + \sum_{j \in V} A(l,j)\lambda_j$$

  where β multiplies the implicit count of 1 observation in this 'bin'. In which case, the implicit 1 would need to have the same units as $A(l,j)\lambda_j$: expectation density. This seems plausible.
- The implicit count of 1, in the above point, occurs for every coincidence recorded by the detector. This includes scatters and randoms, not only trues: these 1s coming from observations in nearby LORs sum to the total count of trues plus additive noise, not just trues. So multiplying the β from equation (13) by these 1s, effectively multiplies it by $t + b$, not just by t, giving $(t + b)^2/t$ rather than the desired $t + b$. So there's something not quite right here. But this seems to built-in to the original formulation of the forward-projection, in any case.

References

Bailey, D.L. et al. (2014). *Nuclear Medicine Physics, A Handbook for Teachers and Students*.

Jacobs, Filip et al. (1998). "A Fast Algorithm to Calculate the Exact Radiological Path through a Pixel or Voxel Space". In: *Journal of Computing and Information Technology*.

Siddon, R.L. (1985). “Fast calculation of the exact radiological path for a three-dimensional CT”. In: *Journal of Medical Physics*.