Bayesian Machine Learning for Gen AI

Assignment - III

November 11, 2025

NOTE: By the end of this assignment you will have hands-on experience applying Markov chain Monte Carlo (MCMC) methods—using the PyMC library—to solve a nonlinear inference problem drawn from real clinical imaging research.

Quantitative MRI with MCMC vs Least Squares

You are a senior data scientist at MRI Research Center of UChicago Mitchell Hospital. One primary goal of your research group is to infer the tissue microstructure from cleverly designed MRI signals. This is important because you can prevent so many unnecessary biopsies (about a million a year) and the associated stress for patients and their families.

Intensity at each MR image pixel depends on

1. the parameters applied at the imaging, namely b and t,
2. the composition of the tissue under that pixel, i.e. if there's a cyst, it changes one way, if there's cancer it is another way (more on this, in a bit).

MRI Physicists and clinicians at your institution hypothesize that the relationship between the MRI image pixel value when scanned using the imaging parameters (b, t) is as follows:

$$S(b,t) = \sum_{i=1}^{3} x_i exp(-by_i) exp(-t/z_i) \tag{1}$$

Take a moment to look at this equation. For our purposes,

- b and t are imaging parameters (independent variables).
- $S(b,t)$ is the measured and noisy signal.
- x_1, x_2 and x_3 are the relative volumes of epithelium, stroma, and lumen, respectively. They should sum to one. We know that if we have too much epithelium, it is bad. So it is important to measure the volume of epithelium.
- y_1, y_2 and y_3 are diffusivity of epithelium, stroma and lumen. Lumen is like protein-water, so we can take $y_3 = 2.9$. We will try to find diffusivity of the epithelium and stroma components because it gives information about the aggressiveness of cancer.

- z_1, z_2 and z_3 are some other parameter called T2 of epithelium, stroma and lumen. We will take $z_2 = 80$ and $z_3 = 750$ and we will focus on finding T2 of epithelium (z_1).

The measured MRI pixel intensity will be corrupted by **Rician Noise** with an unknown variance.

1. **Deterministic approach (20pts).** Download the `MRI_measurements.csv`. You will see 16 measurements for a pixel with various b and t values. Fit the model in Equation 1 using Least-Squares and report what the algorithm infers. For the unknowns, we have the following constraints:

   (a) $0 \leq x_i \leq 1$ and $\sum_i x_i = 1$

   (b) $0.3 \leq y_1 \leq 0.7$

   (c) $0.7 \leq y_2 \leq 1.7$

   (d) $y_3 = 2.9$

   (e) $20 \leq z_1 \leq 70$

   (f) $z_2 = 80$

   (g) $z_3 = 750$

   You may use the helper functions below:

```
def three_compartment_fit(M, D_ep, D_st, T2_ep, V_ep, V_st):
    D_lu = 2.9
    T2_st = 80
    T2_lu = 750
    b, t = M
    S_ep = V_ep*np.exp(-b*D_ep)*np.exp(-t/T2_ep)
    S_st = V_st*np.exp(-b*D_st)*np.exp(-t/T2_st)
    S_lu =(1 - V_ep - V_st)*np.exp(-b*D_lu)*np.exp(-t/T2_lu)

    return S_ep + S_st + S_lu
```

```
def hybrid_fit(signals, bvals = [0, 0.15, 1, 1.5], t = [0, 13, 93, 143]):

    voxel = signals
    X, Y = np.meshgrid(t, bvals)
    xdata = np.vstack((Y.ravel(), X.ravel()))
    ydata = voxel.ravel()
    v = np.zeros((3))
    try:
        fitdata_, _  = scipy.optimize.curve_fit(three_compartment_fit,
                                    xdata,
                                    ydata,
                                    p0 = [0.55, 1.3, 50, 0.3, 0.4],
                                    check_finite=True,
                                    bounds=([0.3, 0.7, 20, 0, 0],
                                            [0.7,  1.7, 70,1, 1]),
                                    method='trf',
                                    maxfev=5000)
    except RuntimeError:
        fitdata_ = [0.55,  1.3, 50, 0.3, 0.4]
    coeffs = fitdata_
    D_ep, D_st, T2_ep, V_ep, V_st = coeffs
    V_lu = 1 - V_ep - V_st
    return D_ep, D_st, T2_ep, V_ep, V_st, V_lu
```

2. **Bayesian approach (80 pts).**Reformulate the problem in PyMC. Now, instead of treating x_1, x_2, x_3, y_1, y_2 and z_1 as unknowns with constraints as I provided above, you will treat them as random variables, with prior distributions. Do simulations with NUTS sampler (the default) setting of PyMC, and report the posterior means and plot the posterior distributions (use ArviZ for this). (80p)

   Some tips:

   (a) x_1, x_2 and x_3 (volume fractions of epithelial, stromal and lumen cells in the tissue) can be deemed as values from categorical probability distribution. Similar to the fact that a good prior for a binary probability distribution (Binomial) is Beta, **a good prior for a categorical probability distribution is Dirichlet**.

   ```
   v = pm.Dirichlet("v", a=np.array([1, 1, 1]))
   v_ep, v_st, v_lu = v[0], v[1], v[2]
   ```

   (b) Since we know y_1 (diffusivity of epithelium), y_2 (diffusivity of stroma) and z_1 (T2 of epithelium) can only be in certain ranges, a **good prior for them can be uniform**

   ```
   D_ep = pm.Uniform("D_ep", lower=0.3, upper=0.7)
   D_st = pm.Uniform("D_st", lower=0.7, upper=1.7)
   T2_ep = pm.Uniform("T2_ep", lower=20, upper=70)
   ```

(c) Just as we discussed in class, we don't know the variance of the noise that our scanner introduces. This very much depends on the age of the scanner too. Older scanners have more noise. The only thing we know is that the noise variance is nonnegative. So a good prior for this could be HalfNormal.

```
sigma = pm.HalfNormal("sigma", sigma=0.3)  # Ensures positive values
```

(d) Don't forget that the noise is Rician

```
y_obs = pm.Rice('y_obs', nu=S, sigma=sigma, observed=table.signal)
```

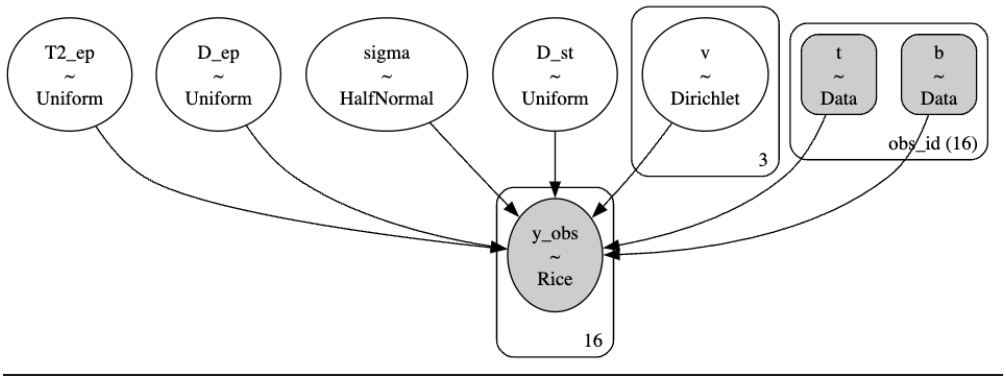

Figure 1: DAG of this Model

3. (BONUS 20p): We learned after biopsy that the true values you we're trying to infer from MRI were:

$$
\begin{aligned}
D_{ep} &= 0.4 \\
D_{st} &= 1.2 \\
T2_{ep} &= 40 \\
v_{ep} &= 0.55 \\
v_{st} &= 0.3 \\
v_{lu} &= 0.15
\end{aligned}
$$

Briefly comment on the performance of Least Squares vs MCMC.(2-3 sentences)