Lecture 10, 11, 12

Numerical Bayesian Techniques

- When the number of parameters in a model, $M(\vec{\theta})$, becomes large, a direct exploration of the posterior pdf becomes infeasible.
  - e.g. for a 2D problem, with θ_1 & θ_2, if we grid each parameter with a resolution of 10^2, we would need to compute $10^2 \times 10^2$ calculations.
  - If the problem now has 5 parameters, $\theta_1, \ldots \theta_5$, each with same resolution, we need $(10^2)^5$ calculations.
  - So you can see how this can become a problem...

- As we work towards a solution to this problem, let's assume we can calculate the posterior:

$$p(\theta) \equiv p(M(\theta)|D,I) \propto p(D|M(\theta),I)\, p(\theta|I)$$

- Thus, in general, our problem is to estimate some integral that is a function only of $\vec{\theta}$,

$$I(\theta) = \int g(\theta)\, p(\theta)\, d\theta$$

- We have two frequent problems:

① Marginalization and parameter estimation:

• We want the posterior for parameters θ_i, with $i = 1, 2, \dots p$, and integral is performed over parameters θ_j, $j = p+1, \dots, k$.

↳ In this case, $g(\theta) = 1$.

• If we want the mean for parameter θ_m, $g(\theta) = \theta_m$.

② Model comparison:

• Here, $g(\theta) = 1$ & integral is performed over all parameters,

$$E(M) \equiv p(D|M,I) = \int p(D|M,\vec{\theta},I)\, p(\vec{\theta}|M,I)\, d\vec{\theta}$$

• Note that here the proper normalization is critical.

Monte Carlo simulation

- The simplest method to evaluate such an integral is Monte Carlo sampling, where we generate a random set of M values of θ, uniformly sampled within the integration volume V_θ:

$$I \approx \frac{V_\theta}{M} \sum_{j=1}^{M} g(\theta_j)\, p(\theta_j)$$

- But this tends to be very inefficient when the integrated function ($p(\theta)$) varies

the integrated function ($p(\vec{\theta})$) varies significantly within the integration volume, which posteriors usually do.
- In effect, we waste time (randomly) sampling areas of low probability.

Markov chain Monte Carlo

- This is the most popular alternative to generic Monte Carlo sampling.
- This method returns a sample of points, or chain, from the k-dimensional parameter space, with a distribution that is asymptotically proportional to $p(\vec{\theta})$.

- Given such a chain, our integral effectively reduces to

$$I = \frac{1}{M} \sum_{j=1}^{M} g(\theta_j)$$

- For example, to estimate the mean of θ_1, we simply take the mean value of all θ_1 from the chain.

- What is a "Markov Chain"?

- A Markov chain is a sequence of random variables where a given value depends (non-trivially) only on its preceding value, i.e.

$$p(\theta_{i+1} | \{\theta_i\}) = p(\theta_{i+1} | \theta_i)$$

- Thus, this is a "memoryless" process.

- In our context, θ can be thought of as a vector in an N-dimensional space, and a realization of the chain represents a path through this space.

- For the chain to reach an equilibrium, or stationarity, it is sufficient (but not necessary) that the transition probability is symmetric:

$$p(\theta_{i+1} | \theta_i) = p(\theta_i | \theta_{i+1})$$

Metropolis-Hastings algorithm:

- There are many algorithms to do MCMC analyses, but the most popular one is the Metropolis-Hastings algorithm,
- Fun fact: most of these methods were generated from physics, in the context of statistical mechanics, thermodynamics, & QFT.

- To get a stationary Markov chain, we need the probability to get a point θ_{i+1} to be proportional to $p(\theta_{i+1})$,

$$p(\theta_{i+1}) = \int T(\theta_{i+1} | \theta_i)\, p(\theta_i)\, d\theta_i$$

where $T(\theta_{i+1}|\theta_i)$ is called the jump kernel or proposal, or transition kernel.

- The stationarity condition can be satisfied by,

$$T(\theta_{i+1}|\theta_i)p(\theta_i) = T(\theta_i|\theta_{i+1})p(\theta_{i+1})$$

- The Metropolis-Hastings algorithm adopts the kernel,

$$T(\theta_{i+1}|\theta_i) = p_{acc}(\theta_i, \theta_{i+1})K(\theta_{i+1}|\theta_i)$$

where the proposed density distribution, $K(\theta_{i+1}|\theta_i)$ is an arbitrary function.

- The proposed point, θ_{i+1}, is randomly accepted with the acceptance probability,

$$p_{acc}(\theta_i, \theta_{i+1}) = \frac{K(\theta_i|\theta_{i+1})p(\theta_{i+1})}{K(\theta_{i+1}|\theta_i)p(\theta_i)}$$

→ When $p_{acc} > 1$, the point is always accepted.
→ When θ_{i+1} is rejected, θ_i is added to the chain instead.

- Usually, a Gaussian distribution centered on θ_i is used for $K(\theta_{i+1}|\theta_i)$.

- The original MH algorithm is based on a symmetric proposal distribution

symmetric proposal distribution,

$$\Rightarrow K(\theta_{i+1}|\theta_i) = K(\theta_i|\theta_{i+1})$$

$$\Rightarrow \boxed{P_{acc} = \frac{p(\theta_{i+1})}{p(\theta_i)}}$$

- Thus, when $p(\theta_{i+1}) > p(\theta_i)$, the point is always accepted.
- If $p(\theta_{i+1}) < p(\theta_i)$, then it is accepted with a probability

$$\frac{p(\theta_{i+1})}{p(\theta_i)}$$

- While $K(\theta_{i+1}|\theta_i)$ above satisfies a Markov chain requirement, it usually takes a number of steps to reach a stationary distribution.

- These initial steps are called the "burn-in" phase,
- There is no general theory for finding the transition from burn-in to stationary, so the burn-in length is determined empirically.
- A popular method to determine if the chain has "converged" or reached "stationarity" is the one proposed by Gelman & Rubin, The R-statistic.

Model Selection with MCMC

- Computing the odds' ratio involves integrating

the un-normalized posterior for a model,

$$\mathcal{L}(M) = \int p(\vec{\theta} | \{x_i\}, I) \, d^k\theta$$

where the integration is over all k model parameters.

- How do we calculate $\mathcal{L}(M)$ from our MCMC chain?

- Recall that the set of points in our MCMC chain is designed to be distributed according to $p(\vec{\theta} | \{x_i\}, I) \equiv p(\vec{\theta})$.

- This means that the local density of points, $g(\vec{\theta})$, for a well-behaved chain is,

$$g(\vec{\theta}) = C N p(\vec{\theta})$$

where,

$C \rightarrow$ an unknown proportionality constant
$N \rightarrow$ number of points in the MCMC chain

- Integrating both sides,

$$\mathcal{L}(M) = \frac{1}{C}$$

since

$$\int p(\vec{\theta}) \, d^k\theta = N$$

- This means that at each point, $\vec{\theta}$, in parameter space, we can estimate the integrand posterior as

$$L(M) = \frac{N p(\theta)}{f(\theta)}$$

- Thus, given an MCMC chain:

→ $p(\vec{\theta}_i)$ is the posterior evaluation at each point

→ $f(\vec{\theta}_i)$ can be estimated from the local distribution of points in the chain.

- The odds ratio problem has thus been reduced to a density estimation problem!

- Choosing an appropriate method to determine $f(\vec{\theta}_i)$, we now have N separate estimators of $L(M)$.
- We can do this for all of our models, and calculate the odds' ratio.

Practical Model Selection

- In practice, the following methods which are computationally efficient are used:

1. Savage-Dickey Density Ratio
2. Product-space sampling
3. Thermodynamic integration
4. Nested sampling

- See class jupyter notebook for details