Chapter 7: Monte Carlo Markov Chain

- MCMC have allowed us to do bayesian analysis on realistic problems
- Used to approximate a distribution using a large sample rather than finding out the exact parameters
- If we have `n` parameters and each parameter can take `m` values, then we will have a total combination of n^m values. These values cannot be represented via a grid.
- Assumptions of MCMC
    - Assumes that prior distribution can be easily evaluated i.e for any value of θ, $p(\theta)$ can be evaluated easily.
    - Assumes likelihood $p(D|\theta)$ can be computed easily for any value of θ
    - No need to evaluate the denominator of bayes theorm.
- MCMC gives approximation of posterior distribution $p(\theta|D)$.

Metropolis Algorithm

- How can we take a large enough sample from a posterior distribution?
- Skipped the derivation for the metropolis Hastings algorithm. Try it later.

Generalised Metropolis Hastings Algorithm

- Politician problem had the following characteristics
    - Discrete positions
    - One dimensional
    - Moves were just one of left or right
- Generalised metropolis hastings algorithm has the following characteristics:
    - Continous positions

- Multidimensional
- More general proposal distribution

Metropolis algorithm to Bernoulli likelihood and beta prior

- We make the following assumptions for the travelling politician problem to use metropolis hastings algorithm
  - Assume we have an infinite distribution of islands
  - Population on each island is relative to the posterior distribution
  - We can jump to any island available(instead of just the adjacent island as described before). We will use the normal distribution for this proposal
- Let the proposal distribution be a normal distribution

$$\Delta\theta = normal(0, \sigma)$$

- Because of the above distribution, we will get the proposed position as:

$$\theta_{pro} = \theta_{cur} + \Delta\theta$$

Steps for MH algorithm

- Start with an initial value of θ. Denote this as θ_{cur}
- Generate proposed jump i.e $\Delta\theta$. Find the value of of the proposed position as:

$$\theta_{pro} = \theta_{cur} + \Delta\theta$$

- Find the probability of moving to the proposed position

$$p_{move} = min(1, \frac{P(\theta_{pro})}{P(\theta_{cur})})$$

$$= min(1, \frac{p(D|\theta)P(\theta_{pro})}{p(D|\theta)P(\theta_{cur})})$$

- Sample a random value from the uniform distribution $uniform(0, 1)$. If this value is less than p_{move} accept the move else reject and compute the current value again.
- Repeat for many samples

Gibbs Sampling

- Applied for models with multiple parameters
- We will apply it to estimate the bias in two coins.
- We will label the coin `j` (j=1 for first coin, j=2 for 2nd coin) with the bias in each coin as θ_j
- We assume independence between attributes for the two coins. Therefore,

$$p(\theta_1, \theta_2) = p(\theta_1)p(\theta_2)$$

- Flips of one coin are indepdendent form the flips of the other coin.
- If y_1 is the output out a flip on the first coin and y_2 is the output of a flip on the second coin

$$p(y_1|\theta_1, \theta_2) = p(y_1|\theta_1)$$

$$p(y_2|\theta_1, \theta_2) = p(y_2|\theta_2)$$

- We observe D_1 flips from first coin with z_1 heads in N_1 flips
- We observe D_2 flips from second coin with z_2 heads in N_2 flips
- We can denote D as the total dataset i.e

$$D = z_1, N_1, z_2, N_2$$

- Because we have assumed independence of events between the coins, the total probability is just the product of the individual bernoulli distributions

 9c91392e.png

- By applying bayes rule, we can obtain the posterior distribution as:

$$p(\theta_1, \theta_2|D) = \frac{P(D|\theta_1, \theta_2) * p(\theta_1, \theta_2)}{\int\int d\theta_1 d\theta_2 P(D|\theta_1, \theta_2) p(\theta_1, \theta_2)}$$

- We can derive the posterior quickly if we assume the priors are beta distributions. Product of bernoulli and beta distributions will be a beta distribution.

Problems with MH

- Proposal distribution should be properly tuned with posterior distribution for this this to work well.

Gibbs Sampling

- Special case of the MH algorithm
- MH algorithm is a generalized version of the Metropolis algorithm

Steps:

- Start walk at a random point
- Next step is only dependent on the current position
- For taking the step:
  - Select one of the componenent parameter $\theta_1, \theta_2, \theta_2, etc.$
  - Cycle through these selected parameters than using random parameters(with many complex models, it will take too many random

steps)
    - Generate new value for selected parameter θ_i as:

      $P(\theta_i | \theta_{j\ !=\ i}, D)$
    - New value for θ_i + unchanged values for $\theta_{j!=i}$
- Accept all proposals(i.e no rejecting proposals based on some random coin toss)

Disadvantage of Gibbs sampling

- Because it changes one parameter value at a time, the process can be slow for highly correlated parameters.

MCMC

- Any simulation that samples a lot of random values from a distribution is called a Monte Carlo simulation
- MH and Gibbs sampling predict the next move based on the current move only. Such random walks are called first order Markov Chains
- The main goals of MCMC are:
    - Values should be *representative* of the posterior distribution.
    - Chains should be of sufficient size so that estimates are *accurate* and *stable*.
    - Chains should be generated *efficiently*.

Representatitveness

- Check using visual examination of the trajectory. Graph of samples as a function of step in the chain is called *traceplot*.
    - Superimpose two or more chains. If they are both representative of the data, they should overlap.

    - In the traceplot, the graph will suggest that the first few iterations should be ignored if the chains don not overlap as they are not representative of the data. This is called the **burn in period**.
    - When the chains overlap each other, they will look like a hairy headgehog. This shows that the chains are representative of the data.
    - If any chain is isolated it => model has not converged
    - If any chain moves around a single value for a long time => model has not converged
- Check smoothed histograms of the sampled values(also called *density plot*)
    - Overlap is an indicator of convergence here.
    - Measure variance between chains vs variance within each chain. This is called *Gelman-Rubin statistic*
        - If = 1.0, chains have converged
        - If > 1.0, chains have not converged

Accuracy

- **Autocorelation**
    - Relation of chain values with chain values k steps ahead.
    - Get values from a chain. Superimpore this with the same chain that is k values ahead. Number of steps k is called **lag**.
    - autocorelation at k denoted as $ACF(k)$
    - Higher ACF => higher corelation => Changes slowly(gradually)
- **Effective sample size**
    - Measure how much independent info is there in the chains. i.e what would be the sample size for a non-corelated chain to give the same information?
    - ESS is denoted by

$$ESS = \frac{N}{1 + 2\sum_{k=1}^{\infty} ACF(k)}$$

    - Can be stopped when $ACF(k) < 0.5$
    - Standard Deviation(SD) of HDI
        - Increases with skewed tails for distributions
        - Increases with smaller ESS
- **Monte Carlo Standard Error(MCSE)**
    - Standard error is computed as

$$SE = \frac{SD}{\sqrt{N}}$$

    - For markov chains, we can substitute the value for N with ESS. Thus, we get

$$MCSE = \frac{SD}{\sqrt{E}SS}$$

Efficiency

- Run applications on parallel hardware.
- Change sampling method
    - Need background knowledge of samples and how they work.
    - Generally HMC is good
- Change parameterization of the model. Combine multiple parameters to a single parameter to reduce the complexity of the model
-