Lecture 6

Estimation vs. Approximation

There are a few key elements of a Bayesian data analysis: 1. Model Specification:

2. Prior Specification:

Once these are specified and the data have been gathered, the posterior distribution $p(\theta|y_1, \ldots, y_n)$ is fully determined. It is exactly:

$$p(\theta|y_1, \ldots, y_n) = \frac{p(\theta)p(y_1, \ldots, y_n|\theta)}{p(y_1, \ldots, y_n)}$$

Excluding posterior model checking, all that remains is to summarize the posterior.

3. Posterior Summary:

For most models we have discussed thus far, $p(\theta|y_1, \ldots, y_n)$ is known in closed form or easy to sample from using Monte Carlo procedures. However, in more sophisticated settings, $p(\theta|y_1, \ldots, y_n)$ is complicated, and hard to write down or sample from. In these cases, we study $p(\theta|y_1, \ldots, y_n)$ by looking at MCMC samples. Thus, Monte Carlo and MCMC sampling algorithms:

- are
- they
- they

For example, if we have Monte Carlo samples $\theta^{(1)}, \ldots, \theta^{(J)}$ that are approximate draws from $p(\theta|y_1, dots, y_n)$, then these sample help describe $p(\theta|y_1, \ldots, y_n)$, for example:

- $\int$
- $\int$

To keep the distinction between and clear, commonly is used do describe how we use $p(\theta|y_1, \ldots, y_n)$ to make inferences about θ and is used to describe the use of Monte Carlo (including MCMC) procedures to approximate integrals.

MCMC Diagnostics

A useful way to think about an MCMC sampler is that there is a moving through and exploring the parameter space. For each region, or set, A the particle needs to spend time proportional to the target probability, $\int_A p(\phi)d\phi$ Consider an example with three modes and denote these three modes as A_1, A_2, A_3. Assume that A_2 is substantantially less than A_1 and A_3.

Given the weights on the mixture components the particle should spend more time in A_1 and A_3 than A_2. However, if the particle was initialized in A_2 we'd hope that the number of iterations are large enough that:

1. The particle

2. the particle

- The technical term associated with item 1 is which means the chain has converged to the target distribution. For the models we have seen thus far, convergence happens quite rapidly, but we will look at this in more depth later on.

- The second item is focused on the speed the particle moves through the target distribution, this is referred to as . An independent sampler like the Monte Carlo procedures we have seen have perfect mixing as each sample is independently drawn from the target distribution. The MCMC samples can be highly correlated and tend to get stuck in certain regions of the space.

People often quantify mixing properties of MCMC samples using the idea of effective sample size. To understand this, first consider the variance of independent Monte Carlo samples:

$$Var_{MC}[\bar{\theta}] =$$

where $\bar{\phi} = \sum_{j=1}^{J} \phi^{(j)}/J$.

The Monte Carlo variance is controlled by the number of samples obtained from the algorithm. In a MCMC setting, consecutive samples $\theta^{(j)}$ and $\theta^{(j+1)}$ are not independent, rather they are usually positively correlated.

Once stationarity has been acheived, the variance of the MCMC algorithm can be expressed as:

$$Var_{MCMC}[\theta] = \cdots = Var_{MC}[\bar{\theta}] +$$

Now if two consecutive samples are highly correlated the variance of the estimator will be much larger than that of an Monte Carlo procedure with the same number of iterations. This is captured in the idea of the The effective sample size is computed such that:

$$Var_{MCMC}[\bar{\theta}] =$$

where S_{eff} can be interpreted as the number of independent Monte Carlo samples necessary to give the same precision as the MCMC samples. Note that the R function `effectiveSize` in the `coda` package will calculate the effective sample size of MCMC output.

We will talk more about MCMC diagnostics after introducing the Metropolis-Hastings algorithm later in class, but the general procedure is:

0. θ

1. Run

2. Assess

An easy solution, especially in the context of Gibbs Sampling is to look at trace plots and histograms of marginal posterior distributions. In conjunction with ESS (Effective Sample Size) calculations this usually gives a good sense of convergence. In other situations, combining visuals displays (trace plots) with other statisics Gelman's R statistic or QDE is a good strategy.

The big picture idea with MCMC, is that we want to guarantee that our algorithm has:

1. Reached

2. Is efficiently mixing

Exercise

Consider the mixture distribution described on p. 99 (Hoff). This distribution is a joint probability distribution of a discrete variable $\delta = \{1, 2, 3\}$, denoting which mixture component the mass comes from and a continuous variable θ. The target density is $\{Pr(\delta = 1), Pr(\delta = 2), Pr(\delta = 3)\} = (.45, .10, .45)$ and $p(\theta|\delta = i) \sim N(\theta; \mu_i, \sigma_i^2)$ where $\{\mu_1, \mu_2, \mu_3\} = (-3, 0, 3)$ and $\sigma_i^2 = 1/3$ for $i \in \{1, 2, 3\}$.

1. Generate 1000 samples of θ from this distribution using a Monte Carlo procedure. Hint: first generate $\delta^{(i)}$ from the marginal distribution $p(\delta)$ and then generate $\theta^{(i)}$ from $p(\theta|\delta)$ Plot your samples in a histogram form and superimpose a curve of the density function. Comment on your samples, do they closely match the true distribution?

2. Next, generate samples from a Gibbs sampler using the full conditional distributions of θ and δ. You already know the form of the full conditional for θ from above. The full conditional distribution for δ is given below:

$$Pr(\delta = d|\theta) = \frac{Pr(\delta = d) \times p(\theta|\delta = d)}{\sum_{d=1}^{3} Pr(\delta = d) \times p(\theta|\delta = d)}$$

Hint: for $p(\theta|\delta = d)$ evaluate θ from a normal distribution with parameters $\{\mu_d, \sigma_d^2\}$. Intialize θ at 0.

a. Generate 100 samples using this procedure. Plot your samples as a histogram with the true density superimposed on the plot. Also include a plot of your θ value on the y-axis and the iteration number on the x-axis. This is called a trace plot, and allows your to visualize the movement of your MCMC *particle*. Comment on how close your samples match the true density. What does the trace plot reveal about the position of θ over time (the iterations)? Does the proportion of the time the sample spends in each state (δ) match the true probabilities?

b. Repeat for 1000 samples.

c. Repeat for 10000 samples.

3. Now repeat part 2, but instead initialize θ at 100. How does this change the results from part 2? Looking at trace plots, do the chains from the three plots seem to be similar?