Markov chain Monte Carlo

Peter Beerli

October 10, 2005

[this chapter is highly influenced by chapter 1 in *Markov chain Monte Carlo in Practice*, eds Gilks W. R. et al. Chapman and Hall/CRC, 1996]

1 Short history

Many problems can not be solved analytically, but can be solved using statistical sampling. This idea is certainly old and was first used in a question by Georges-Louis Leclerc, Comte de Buffon (Buffon's needle experiment) and William Gosset. Although these early applications were typically used to simulate data on a understood analytical problem. In 1945 and the following years Nicolas Metropolis and others, including Stanislaw (Stan) Ulam developed statistical sampling method to test the ENIAC computer. Metropolis coined the term Monte Carlo methods (the famous casino town in Monaco in Southern France) [influenced by the fondness of poker of Ulam who had an uncle who once borrowed money to go gambling in Monte Carlo]. Enrico Fermi was using statistical sampling for many problems in the 1930 and later, but he never published his way but used it to impress others about the accuracy of results. In 1953 Metropolis et al. described the now famous Metropolis algorithm and so the first Markov chain Monte Carlo method.

2 Monte Carlo methods

Monte Carlo methods are methods that perform statistical sampling to get the expectation of a function. We want to approximate

$$\mu = \mathrm{E}(f(X_i))$$

with independently identically distributed (iid) samples $X_1, X_2, ...$ using the sample mean

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(f(X_i))$$

If we sample long enough we approximate the original expectation, it is important to note that we always should supply the standard deviation of this sampling process because, that should converge to the standard deviation of a Normal distribution. Note that n is under the control of the researcher and is data-independent, we always can run the analysis longer (increase n) and get a more accurate result.

Monte Carlo simulation is an important tool for integration in almost any field of research.

3 Markov chain (MC)

If a process is producing points and the future is independent from the past, for example

$$\text{Prob}(X_n = a_n | X_0 = a_0, X_1 = a_1, ..., X_{n-2} = a_{n-2}, X_{n-1} = a_{n-1}) = \text{Prob}(X_n = a_n | X_{n-1} = a_{n-1})$$

The values $a0, a_1, ...a_n$ form a Markov chain. A random walk or a sequence of mutations are examples of a Markov chain.

4 Markov chain Monte Carlo (MCMC)

Markov chain Monte Carlo methods are methods that perform statistical sampling to get the expectation of a function. We want to approximate

$$\mu = \text{E}_\pi(f(X_i))$$

where π is the equilibrium distribution or stationary distribution, with samples from a Markov chain $X_1, X_2, ...$ from the distribution $f(\cdot)$ using the sample mean

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(f(X_i))$$

The only difference to MC is that instead of iid samples we draw dependent samples. We would prefer MC over MCMC because the samples $X_0, X_1, ...$ are independent of each other but generating independent samples from the distribution $\pi(\cdot)$ is often infeasible and we resort to MCMC. The

Markov chain of random variables $X_1, X_2, ...$ is sampled from a distribution $\text{Prob}(X_t|X_{t-1})$ where X_t is only dependent on X_{t-1} no other state. This distribution $\text{Prob}(\cdot|\cdot)$ is called the *transition kernel* of the chain. Typically, we assume that that the transition kernel is not dependent on the time t ($\text{Prob}(\cdot|\cdot)$ is time-homogenous).

Although we just stated that the transition kernel is independent of all but the last state, we often recognize dependencies. The transition kernel is only independent from the past in the choices if the next state. Depending on the method how we move from one state to the next the states are not independent draws. Running the chain for a long time will guarantee that it forgets the start conditions and achieves an equilibrium, hence equilibrium or stationary distribution. So we can say that $\text{Prob}^{(t)}(\cdot|X_0)$ will converge to the stationary distribution $\phi(\cdot)$. Often, the initial states in a MCMC analysis are discarded as *burn-in b*. The magnitude of the burn-in depends on several things, such as data, size of problem, and choice of transition kernel.

$$\overline{\mu} = \frac{1}{n-b} \sum_{i=b+1}^{n} (f(X_i))$$

5 Metropolis-Hastings algorithm

A Markov chain can be used to approximate the expectation of $f(X)$ using the stationary distribution $\phi(\cdot)$, but we need to discover how to construct a Markov chain such that $\phi(\cdot)$ is the same as the distribution of interest $\pi(\cdot)$. Metropolis et al (1953) developed the original method that then was generalized by Hastings (1970). At time t we propose the next state X_{t+1} that is chosen by first sampling a candidate Y from the proposal distribution $g(\cdot|X_t)$. Note that the proposal distribution can but does not need to depend on the last state X_t. The candidate Y is accepted with $\alpha(X_t, Y)$ where

$$\alpha(X_t, Y) = min(1, \frac{\pi(Y)g(X_t|Y)}{\pi(X_t)g(Y|X_t)}.$$

If the candidate is accepted then $X_{t+1} = Y$, otherwise $X_{t+1} = X_t$. We can construct a simple algorithm (1). The proposal distribution can have any form and the stationary distribution will be $\pi(\cdot)$. The choice of the proposal distribution $g(\cdot)$ is crucial as we will see later, but we are quite free to choose a convenient distribution because one can show that if we can sample X_t using $g(\cdot)$ from $\pi(\cdot)$, this is also true for X_{t+1} (see Gilks et al 1996 for details).

There are 3 major requirements so that MCMC really works

- *Irreducibility*: the Markov chain must be able to reach all interesting parts of the distribution.

Algorithm 1 Metropolis-Hastings algorithm (Gilks, W. R. et al 1996)

```
Initialize $X_0$; set $t = 0$
loop
  Sample a point $Y$ from $g(\cdot|X_t)$
  Sample a Uniform(0,1) random variable $U$
  if $U \leq \alpha(X_t, Y)$ then
    $X_{t+1} \leftarrow Y$
  else
    $X_{t+1} \leftarrow X_t$
  end if
  increment $t$
end loop
```

- *Recurrence*: all interesting parts must be reached (in principle) infinitely often if the chain is run infinitely long.
- *Convergence*: the sample mean must converge to the expectation.

In reality only the first tenet is crucial and if all states can be reached often (recurrence) then the convergence criteria follows.

Implementations of the Metropolis-Hastings algorithm are not that difficult, once one has found a good proposal distribution, but what is good?

Bad proposal distribution have the property that they propose values that are not in line with the distribution we want to approximate resulting into many proposal that are not accepted. the Markov chain is not *mixing* well. Typically, we would like to accept new values in a range of 0.1 to 0.5. the Markov chain should concentrate the sampling on interesting regions, low acceptance ratios suggest that the proposal distribution is suggesting moves that are to different from the current location and so loose the current context (for example if the parameter space os very large and only a small range is contributing to the observed data proposals that jump out of that range will be typically rejected).

Figure 1: Examples of signatures of (a) good and (b) bad mixing

6 Gibbs sampler

Metropolis-Hastings sampler often have the problem that their acceptance ratio is low. For some distributions it is not all that difficult to find better sampling strategies, so that one can guarantee acceptance. Gibbs sampling (a misnomer as it was applied to Gibbs distribution on lattices by Geman and Geman (1984) but has much larger general use; the method is identical to the *heat-bath algorithm* in statistical physics). We can think of the Gibbs sampler as a special case of the Metropolis-Hastings sampler because it uses the proposal distribution

$$g(Y|X_t) = \pi(Y)$$

this results in the acceptance/rejection ration of

$$\alpha(X_t, Y) = min(1, \frac{\pi(Y)g(X_t|Y)}{\pi(X_t)g(Y|X_t)} = min(1, \frac{\pi(Y)\pi(X_t)}{\pi(X_t)\pi(Y)} = 1.$$

The Gibbs proposal is not always easy to achieve and often needs more computation as one needs to sample from $\pi(\cdot)$. this is is sometimes impossible or needs much more computational effort than doing Metropolis-Hastings.

7 Convergence

These algorithm depend on the choice if the proposal distribution and on the sample mean (to approximate the expectation). In the best of all worlds we would choose the proposal distribution as close as possible to the actual (true) distribution and we also would run the sampler very very very very very long. Several measures were proposed to estimate whether a chain has converged or not, but this is still an active research topic. Current best practise is to run several chains from overdispersed starting points and check the variance between and within the chains is accetable or not. This results in comparison of several runs and whether we can trust the convergence diagnostic still depends on several things: is some parameter space easier to reach than other? What if we always start from a similar region instead of being really over dispersed?

8 Combining multiple chains

Geyer (1994) described a scheme that allows the combination of multiple chains using different proposal distributions $g(\cdot|X_0^{(i)})$. His *reverse logistic regression* reweights the chains so that they can be combined to estimate an overall value. [An example of this approach will follow in the chapter about population genetic inference using the coalescent].

9 Metropolis-coupled MCMC

A major problem with simple proposal distributions is that they often propose lunatic states that get rejected, if the highly dimensional parameter landscape is steep in most directions then a chain will most often not move. Geyer and Thompson (1995) suggested to use multiple chains with different proposal distributions. The chains "heat" the acceptance/rejection ratio so that that hot chains accept more often than cold chains. Therefore the "hot" chains bounce around in the parameter space more vividly than the "cold" chain. The hot/cold analogy correlates to hot and cold gas molecules. the acceptance/rejection ratio changes to

$$\alpha(X_t, Y) = min \left(1, \left[\frac{\pi(Y)g(X_t|Y)}{\pi(X_t)g(Y|X_t)}\right]^{1/T}\right).$$

where T is the temperature. It is easy to see that if the temperature T is 1 nothing changes and this cold chain behaves like the simple MCMC chain. If T increases a change is more readily accepted. With $T \to \infty$ ever change will be accepted: the chain will be really hot.

Every once and so often a comparison between the different heated chains is made (typically ever step) and two chains are picked at random (or two chains with adjacent temperatures are picked) and a Metropolis-Hastings step is preformed using

$$min(1, \frac{\pi_i(X_t^{(j)})\pi_j(X_t^{(i)})}{\pi_i(X_t^{(i)})\pi_j(X_t^{(j)})}.$$

When accepted the chains swap their temperatures i and j (or swap their state). Swapping temperatures forces to record the results on all chains (desirable in a cluster environment) whereas swapping states allows to keep records on a single node. The results of the "hot" chains are typically discarded. This approach has proven important for problems that mix slowly or not at all.

10 Study questions

- Why is the Gibbs sampler not always the best strategy?
- The principle of statistical sampling seems rather old although it seems not many have used it. Reason?
- Devise a strategy for using heated chains, what would be best to cope with unknown situations? For example some data sets mix well other do not.
- Given an example of pathological signature of a mixing plot and explain why it is pathological?
- Why is the addition of Hastings to the Metropolis acceptance-ratio so important?