Notes for Don van Ravenzwaaij, Pete Cassey, Scott D. Brown "A simple introduction to Markov Chain Monte–Carlo sampling"

Malte Brinch
University of Copenhagen
Septemper 2019

1. Introduction

MCMC allows us to characterize a distribution without knowing all of the distribution's mathematical properties by randomly sampling values out of the distribution. A strength of MCMC is that we can draw samples from it even though we how to calculate the density for different samples. MCMC stands for Monte-Carlo Markov-Chain and has 2 properties:

- Monte–Carlo is the practice of estimating the properties of a distribution by examining random samples from the distribution, instead of doing it directly from equations.
- The Markov chain property of MCMC is the idea that the random samples are generated by a special sequential process. Each random sample is used as a stepping stone to generate the next random sample, which is why it is called a chain. The current step is only dependant directly on the one before in the chain.

MCMC works well with Bayesian inference due to its focus on posterior distributions, which are hard to derive analytically. The way the data are used to update the prior belief is by examining the likelihood of the data given a certain (set of) value(s) of the parameter(s) of interest. The article describes one way of adding random noise to create proposals, and also an approach to the process of accepting and rejecting called the Metropolis algorithm (see article for full example).

There are some limitations to this method for example when MCMC is applied to Bayesian inference, this means that the values calculated must be posterior likelihoods, or at least be proportional to the posterior likelihood. Also he proposal distribution should be symmetric or the algorithm must be modified to account for asymmetry. Another thing the first steps in the Markov-chain is ignored since the initial guess might be wrong and the answer needs to converge. The width of the proposal distribution is sometimes called a tuning parameter of this MCMC algorithm and can affect whether there is a lot of rejections or maybe the algorithm is stuck in a local maxima (think optimizing chi squared).

for strongly correlated parameters a more advanced method than Metropolis-Hastings is needed since correlations between parameters can lead to extremely slow convergence of sampling chains, and sometimes to nonconvergence (within any reasonable time). Gibbs sampling breaks down the problem by drawing samples for each parameter directly from that parameter's conditional distribution, or the probability distribution of a parameter given a specific value of another parameter. For a combined Metropolis within Gibbs approach The key is that for a multivariate density, each parameter is treated separately: the propose/accept/reject steps are taken parameter by parameter. Example is given in the article.

Differential evolution can help improve the speed of convergence for correlated parameters relative to the Gibbs approach, see figure as to why. The idea is that the chains are not independent – they interact with each other during sampling, and this helps address the problems caused by parameter correlations. It does this by using the difference between other chains to generate new proposal values. The tuning parameters are only two: a multiplier γ and a very small amount of random noise, whereas before each parameter needed one. A good guess for γ is $\frac{2.38}{\sqrt{2K}}$.

There are some examples of code in this paper.

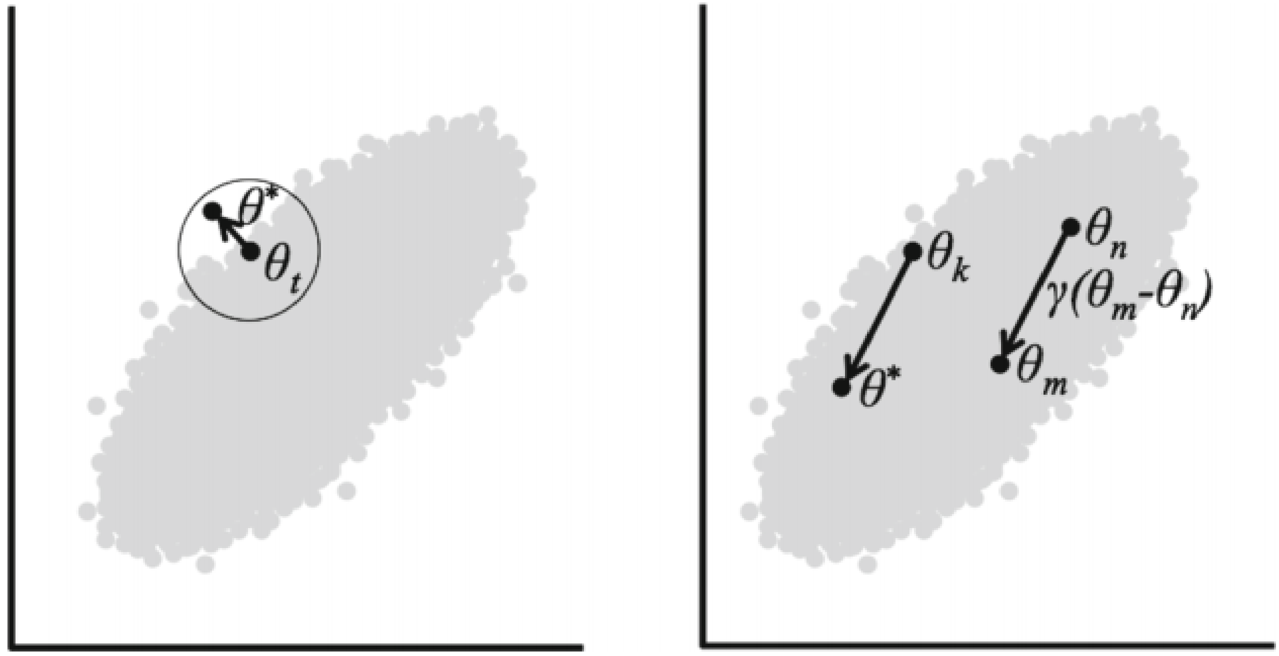

Fig. 3 *Left panel:* MCMC sampling using a conventional symmetrical proposal distribution. *Right panel:* MCMC sampling using the crossover method in Differential Evolution. See text for details