A detailed reference of MCMC algorithms

Siddharth Bhat (20161105) siddu.druid@gmail.com

April 12, 2020

1 Why do we need MCMC? A practitioner's perspective

Consider that we are plonked down in the `C` programming language, and our only method to generate random numbers is to call `int rand(void)`. However, *the type is a lie*, since we are able to modify global mutable state. So, really, to have a mathematical discussion about the whole state of affairs, we will write the type of `rand` as $\texttt{rand} : S \to S \times \texttt{int}$ — that is, it receives the entire state of the program, and then returns the `int`, along with the new state of the program. $\texttt{uniformbool} : \texttt{S} \to \texttt{S} \times \{0, 1\}$.

When we say that `rand` generates random numbers, we need to be a little more specific: what does it mean to generate random numbers? we need to describe the *distribution* according to which we are receiving the random numbers from the random number generator *rand*. What does that mean? Well, it means that as we generate more numbers, the *empirical distribution* of the list of numbers we get from the successive calls to `uniformbool` tends to some *true distribution*. We will call this *true distribution* succincty as **the** distribution of the random number generator. Formally, let us define $F(t) \equiv \int_0^t P(x)dx$ to be the cumulative distribution of P.

In the case of `uniformbool`, we are receiving random numbers according to the distribution:

$$P_{\texttt{uniformbool}} : \{0, 1\} \to [0, 1]; \qquad P_{\texttt{uniformbool}}(x) = 1/2$$

That is, both 0 and 1 are *equally likely*. However, this is extremely boring. What we are *usually* interested in is to sample $\{0, 1\}$ in some *biased* fashion:

$$P^{bias}_{\texttt{uniformbool}} : \{0, 1\} \to [0, 1]; \qquad P^{bias}_{\texttt{uniformbool}}(0) = bias; \qquad P^{bias}_{\texttt{uniformbool}}(1) = 1 - bias$$

And far more generally, we want to sample from *arbitrary domains* with *arbitrary distributions*:

$$\texttt{sampler}^P_X : S \to S \times X;$$

This function is boring. What we *really* want to sample from are more interesting distributions. For example:

- The normal distribution $P(x : \mathbb{R}) = e^{-x^2}$.
- The poisson distribution $P(x : \mathbb{N}) = e^{-\lambda}\lambda^n/n!$.
- A custom distribution $P(x : \mathbb{R}) = |sin(x)|$.

1.1 Fundamental Problem of MCMC sampling

Given a weak, simple sampler of the form $\texttt{rand} : S \to S \times \texttt{int}$, build a sampler $\texttt{sampler}(P, X) : T \to T \times X$ which returns value distributed according to some distribution of choice $P : X \to [0, 1]$.

1.2 Sampling use case 1. simulation

1.3 Sampling use case 2. gradient free optimisation

2 Whirlwind tour of the underpinnings of MCMC sampling

3 Metropolis-Hastings

Assume we wish to sample from some distribution $P : X \to [0, 1]$, and we have access to a uniform sampler $X_{\texttt{uniform}} : () \to X$.

4 Low discrepancy sequences

5 Gibbs sampling

6 Hamiltonian Monte Carlo

7 No-U-Turn sampling

8 Replica Exchange

9 Discontinuous Hamiltonian monte carlo