Integration

Rachel

2022-12-15

This portfolio covers the intuition behind MCMC methods.

Properties of Markov Chains

Definitions

We define a discrete Markov chain $\mathbf{X} := (X_n)_{n\geq 0}$ on a measurable space $(x, \mathcal{X})$, so that for $A \in \mathcal{X}$ we have

$$P(X_n \in A | X_0 = x_)), \ldots, X_{n-1} = x_{n-1}) = P(X_n \in A | X_{n-1} = x_{n-1}) \tag{1}$$

LLN and CLT

We now define some theorems for the case of Markov Chain

Theorem: LLN For a time-invariant, positive Harris Markov chain $\mathbf{X} = (X_n)_{n\geq 0}$ with stationary distribution π and any $f \in L_1(X, \pi) = \{f : \pi(|f|) < \infty\}$

$$\lim_{n\to\infty} \frac{1}{n} S_n(f) = \pi(f) \tag{2}$$

almost surely for any intial distribution.

Theorem: CLT For a time-invariant, positive Harris and geometrically ergodic Markov chain $\mathbf{X} = (X_n)_{n\geq 0}$ with stationary distribution π and $\pi\left(|f|^{2+\delta}\right)$ for some $\delta > 0$,

$$n^{\frac{1}{2}} \left\{n^{-1} S_n(f) - \pi(f)\right\} \to^{L} \mathcal{N}(0, \sigma^2(f)) \tag{3}$$

as $n \to \infty$, where $\bar{f} = f - \pi(f)$ and

$$\sigma^2(f) = \mathbb{E}_X \left[\bar{f}(X_0)^2\right] + 2 \sum_{k=1}^{\infty} \mathbb{E}_\pi \left[barf(X_0) \bar{f}(X_k)\right] < \infty \tag{4}$$

Metropolis-Hastings

The most common method of constructing Markov chains is the Metropolis-Hastings algorithm.

Algorithm

We assume π has a density w.r.t a measure λ.

We then specify a proposal transition kernel Q, with density q with respect to λ:

$$Q(x, dz) = q(x, z)\lambda(dz) \tag{5}$$

Aiming to obtain $P_{MH}(x, \cdot)$, we perform two steps

1. Simulate $x* \sim Q(x, \cdot)$
2. Take $x = x*$ with probability

$$\alpha_{MH}(x, x*) := \min\left\{1, \frac{\pi(x*)q(x*, x)}{\pi(x)q(x, x*)}\right\} \tag{6}$$

Thus we only need π up to a constant and to simulate from $Q(x, \cdot)$ to obtain sample from P_{MH}.

The code for this is now given below, where the input `Q` is a list containing a `density` function and a `sample` function:

```
make.MH_kernel <- function(pi,Q){
  q <- Q$density
  P <- function(x) {
    x_new <- Q$sample(x)
    alpha <- min(1, pi(x_new)*q(x_new,x)/pi(x)/q(x,x_new))
    ifelse(runif(1) < alpha, x_new, x)
  }
  return(P)
}
```

Normal Distribution Example

We first need to construct a function which produces the `Q` as described above for a normal distribution. For simplicity, we will only consider the univariate proposal

```
make.normal_Q <- function(sigma) {
  Q <- list()
  Q$sample <- function(x) {
    x + sigma*rnorm(1)
  }
  Q$density <- function(x,y) {
    dnorm(y-x, sd=sigma)
  }
  return(Q)
}
```

Can now use these two functions to obtain a `P` we can simulate a Markov chain from

```
Q_normal <- make.normal_Q(1)
P <- make.MH_kernel(dnorm, Q_normal)
```

Now we can obtain a Markov chain:

```r
n <- 100000

xs<- rep(0,n)
x<- 1

for (i in 1:n) {
  x <- P(x)
  xs[i] <- x
}
```