gibbs_sampling_and_random_walk

Yijin Wang

2023-04-30

Given the fact that we can easily identify the distribution that most of the conditional posteriors follows, we choose to use Gibbs sampling. Gibbs sampling is an efficient sampler to create approximated data that follows a target joint distribution with several conditional distributions. It is a sequential approach that updates parameter value one by one with information of other parameters.

Still, we could not find a regular distribution that has the density function that matches with the conditional posterior of σ^2. Therefore, we incorporated Metropolis Hasting Random Walk into Gibbs sampling. Metropolis Hasting Random Walk finds approximated values of σ^2 by accepting and rejecting proposed value with an acceptance probability that helps retain balance of the system. In our algorithm, we worked with log-scaled posterior values instead of the original-scaled posterior because $\frac{1}{\sigma^N}$ dominates the entire posterior value where N is the total number of observations in the dataset.

In our investigation of hurricane wind speed, we used Gibbs sampling and as Metropolis Hasting Random Walk as the following:

Algorithm 1 Gibbs Sampling

function GIBBS(starting_values, V,S,X,D,Y, nitr,ν,H, hurricane_idx)
    create posts results lists to collect results for B, μ, A, γ, σ^2
    B^1 = f(B|start_μ, start_σ^2, start_A, start_γ, Y)
    $\mu^1 \leftarrow$ f(μ|B^1, start_σ^2, start_A, start_γ, Y)
    $A^1 \leftarrow$ f(A|B^1, μ^1, start_σ^2, start_γ, Y)
    $\gamma^1 \leftarrow$ f(γ|B^1, μ^1, start_σ^2, A^1, Y)
    σ^2_chain $\leftarrow$ metropolis_hasting_sigma_squared(start_σ^2, iter = 500, a = 0.8,Y,D,B^1,X,γ^1)
    $(\sigma^2)^1 \leftarrow$ mean(σ^2_chain[400:500])
    for i in 2:nitr **do**
        $B^i \leftarrow$ f(B|μ^{i-1}, $(\sigma^2)^{i-1}$, A^{i-1}, γ^{i-1}, Y)
        $\mu^i \leftarrow$ f(μ|B^i, $(\sigma^2)^{i-1}$, A^{i-1}, γ^{i-1}, Y)
        $A^i \leftarrow$ f(A|B^i, μ^i, $(\sigma^2)^{i-1}$, γ^{i-1}, Y)
        $\gamma^i \leftarrow$ f(γ|B^i, μ^i, $(\sigma^2)^{i-1}$, A^i, Y)
        σ^2_chain $\leftarrow$ metropolis_hasting_sigma_squared($(\sigma^2)^{i-1}$, iter = 500, a = 0.8,Y,D,B^i,X,γ^i)
        $(\sigma^2)^i \leftarrow$ mean(σ^2_chain[401:500])
    end for
    return posts results lists that collect results for B, μ, A, γ, σ^2
end function

Figure 1: Gibbs Sampling

Algorithm 2 Metropolis Hasting sigma squared

```
function METROPOLIS HASTING SIGMA SQUARED(start_sigma_squared, nitr, a, B, X,D,Y, H,
hurricane_idx)
    Initialize chain
    chain[1] ← start_sigma_squared
    for i in 2:itr do
        curr ← chain[i − 1]
        prop ← a sample from Unif(curr-a, curr+a)
        prop_posterior ← sigma_squared_log_posterior(prop,...)
        curr_posterior ← sigma_squared_log_posterior(curr,...)
        posterior_ratio ← exp(prop_posterior - curr_posterior)
        α ← min(posterior_ratio, 1)                              ▷ acceptance probability
        u ← runif(1)
        if u < α then
            chain[i] ← prop                                      ▷ accept proposal
        else
            chain[i] ← curr                                      ▷ reject proposal
        end if
    end for
    return chain
end function
```

Figure 2: Metropolis Hasting Random Walk