Pseudocode

Max Thannheimer & Marco Oesting

July 2024

Algorithm 0.1: MCMC Algorithm

input: observations $(x_i(s_0), \dots, x_i(s_M))_{i=1,\dots,n_{\text{obs}}}$, risk functional r, threshold u, parametric model,
prior π, proposal $q(\cdot,\cdot)$, chain length n_{MCMC}, fine grid points $t_0, \dots, t_N$
output: Markov chain of parameters of length n_{MCMC}
begin
    initialize parameter (θ_0, α_0) from prior π
    for $n \in \{1, \dots, n_{\text{MCMC}}\}$
        sample $r(X_i) \mid X_i(s_0) = x_i(s_0), \dots, X_i(s_M) = x_i(s_M)$ under parameter $(\theta_{n-1}, \alpha_{n-1})$
        using Algorithm 2
        set $I = \{i \mid r(X_i) > u\}$
        sample $(\theta_{\text{prop}}, \alpha_{\text{prop}}) \sim q((\theta_{n-1}, \alpha_{n-1}), \cdot)$
        sample $v \sim \text{Unif}[0, 1]$
        if $v < a((\theta_{n-1}, \alpha_{n-1}), (\theta_{\text{prop}}, \alpha_{\text{prop}}))$
            set $(\theta_n, \alpha_n) = (\theta_{\text{prop}}, \alpha_{\text{prop}})$
        else
            set $(\theta_n, \alpha_n) = (\theta_{n-1}, \alpha_{n-1})$
        end if
    end for
    return $(\theta_0, \alpha_0) \dots, (\theta_{n_{\text{MCMC}}}, \alpha_{n_{\text{MCMC}}})$
end

Algorithm 0.2: Cond-X Algorithm

input: observations $(x_i(s_0), \dots, x_i(s_M))_{i=1,\dots,n_{\text{obs}}}$, risk functional r, fine grid points $t_0, \dots, t_N$,
parametric model with parameter (θ, α), chain length n_{condX}
output: sample $r(X_i) \mid X_i(s_0) = x_i(s_0), \dots, X_i(s_M) = x_i(s_M)$ under parameter (θ, α)
begin
    for $i \in \{1, \dots, n_{\text{obs}}\}$
        initialize $w \sim W \mid W(s_1) = \frac{x_i(s_1)}{x_i(s_0)}, \dots, W(s_M) = \frac{x_i(s_M)}{x_i(s_0)}$ under parameter (θ, α)
        for $n \in \{1, \dots, n_{\text{condX}}\}$
            sample $w_{\text{prop}} \sim W \mid W(s_1) = \frac{x_i(s_1)}{x_i(s_0)}, \dots, W(s_M) = \frac{x_i(s_M)}{x_i(s_0)}$ under parameter (θ, α)
            sample $v \sim \text{Unif}[0, 1]$
            if $v < a(w, w_{\text{prop}}) = \min\left\{\frac{r^\alpha(w_{\text{prop}})}{r^\alpha(w)}, 1\right\}$
                set $w = w_{\text{prop}}$
            end if
        end for
        set $w_i = w$
    end for
    return $r(X_i) = x_i(s_0) \cdot r(w_i)$
end

Algorithm 0.3: CondGaussian Simulation Algorithm

input: observations $(x_i(s_0), \dots, x_i(s_M))_{i=1,\dots,n_{\text{obs}}}$, fine grid points $t_0, \dots, t_N$,
parametric model with parameter (θ, α)
output: sample $W \mid W(s_1) = \frac{x_i(s_1)}{x_i(s_0)}, \dots, W(s_M) = \frac{x_i(s_M)}{x_i(s_0)}$ under parameter (θ, α)
begin
    sample $w_{\text{uncond}}(t_1), \dots, w_{\text{uncond}}(t_N)$ unconditional on fine grid with parameter (θ, α) via alg. 4
    set $\Sigma_{ts} = (\gamma_\theta(t_i - s_0) + \gamma_\theta(s_j - s_0) - \gamma_\theta(t_i - s_j))_{i \in \{1,\dots N\}, j \in \{1,\dots M\}}$

```
    set  $\Sigma_{ss} = (\gamma_\theta(s_i - s_0) + \gamma_\theta(s_j - s_0) - \gamma_\theta(s_i - s_j))_{i\in\{1,\ldots M\}, j\in\{1,\ldots M\}}$
    return  exp(
    $\log w_{\text{uncond}}(t_1, \ldots, t_N) + \Sigma_{ts}\Sigma_{ss}^{-1}(\log\left(\frac{x_i(s_1)}{x_i(s_0)}, \ldots \frac{x_i(s_M)}{x_i(s_0)}\right) - \log w_{\text{uncond}}(s_1, \ldots, s_M))$
    )
end
```

Algorithm 0.4: log-Gaussian Simulation Algorithm

```
input:   fine grid points t_0, ..., t_N, parametric model with parameter (θ, α)
output:  sample from W under parameter (θ, α) on fine grid
begin
    sample g(t_1), ..., g(t_N) ~ standard FBM under parameter (θ, α)
    return w_j = exp( (1/α)(g(t_j) − g(s_0) − γ_θ(t_j, s_0)) )
end
```

Algorithm 0.5: Acceptance Rate MCMC Algorithm

```
input:   exceedance observations (x_i(s_0), ..., x_i(s_M))_{i ∈ I_exceed}, risk functional r,
         prior π, proposal q(·,·), parametric model with parameter (θ_prop, α_prop) and (θ_old, α_old),
         sample size n_CondGauss, sample size n_Gauss, fine grid points t_0, ..., t_N
output:  acceptance rate a((θ_old, α_old), (θ_prop, α_prop))
begin
    for θ ∈ {(θ_prop, α_prop), (θ_old, α_old)}
        for i ∈ I_exceed
            set a = GaussianDensity((θ, α), α · log( x_i(s_1)/x_i(s_0), ..., x_i(s_M)/x_i(s_0) )) · α^M
            set b = α (x_i(s_0)/u)^(−α−1)
            estimate c = E(l(W)^α) under parameter (θ, α) via n_Gauss times algorithm 4
            set log likelihood((θ, α), i) = log a + log b − log c
        end for
        set log likelihood(θ, α) = Σ_{i ∈ I_exceed} log likelihood((θ, α), i)
    end for
    return min{1, π((θ_prop, α_prop)) q((θ_prop, α_prop), (θ_old, α_old)) likelihood((θ_prop, α_prop)) / (π((θ_old, α_old)) q((θ_old, α_old), (θ_prop, α_prop)) likelihood((θ_old, α_old)))}
end
```