Scale Mixture Of Normals (SMiN) Representation

In foreign exchange markets such as the EUR/USD, market moves are widely regarded to be symmetric in nature [**tankov2003financial**]. This motivates the selection of the parameter $\beta = 0$. Instead of considering the general α-stable process, we can instead restrict ourselves to a more analytically tractable sub-class, the a Symmetric α-stable (SαS) process instead.

For $\beta = 0$, we have a convenient Scale Mixture of Normals (SMiN) representation based off the product property of α-stable distributions: If X and Y are independent random variables with $\lambda_t \sim S_{\alpha/2}(1,1,0)$ and $\eta_t \sim S_2(1,0,0) = \mathcal{N}(0,1)$, then $\lambda_t \eta_t \sim S_\alpha(1,0,0)$.

We can then convert the discretised state space model specified in **??** into SMiN form as specified in Equation 1 below.

$$\underbrace{\begin{bmatrix} x_{1,t+\delta t} \\ x_{2,t+\delta_t} \end{bmatrix}}_{\mathbf{x_{t+\delta t}}} = \underbrace{\begin{bmatrix} 1 & \delta t \\ 0 & e^{\theta \delta t} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix}}_{\mathbf{x_t}} + \underbrace{\begin{bmatrix} 0 \\ \sigma_{\delta t} \end{bmatrix}}_{\mathbf{b}} \sqrt{\lambda_{t+\delta t}} \eta_{t+\delta t}; \qquad \eta_{t+\delta t} \sim \mathcal{N}(0,1),\ \lambda_{t+\delta t} \sim S_{\alpha/2}(1,1,0) \tag{1}$$

This means that conditional upon us observing $\lambda_{t+\delta t}$, $\mathbf{x}_{t+\delta t}$ is gaussian, making inference much more tractable. We exploit this fact by designing a Rao-Blackwellised Particle Filter to sample $\lambda_{t+\delta t}$ by using a simple particle filter, and propogating the states using a Kalman Filter conditioned upon the sampled $\lambda_{t+\delta t}$ of the particle.

We take a short detour to properly formulate the inference problem using the SMiN representation here. In the inference problem, we are seeking to infer the state variables $\mathbf{x}_{t+\delta t}$ given the observations $y_{t+\delta t}$. Changing indexes using $k = t + \delta t, k - 1 = t$, to reflect the discretised nature of the problem, we assume model dynamics as follows:

$$\begin{aligned} \mathbf{x}_k &= \mathbf{A}\mathbf{x}_{k-1} + \mathbf{b}\sqrt{\lambda_k}\eta_k \\ y_k &= \mathbf{C}\mathbf{x}_k + d\epsilon_k \\ \lambda_k &\sim S_{\alpha/2}(1,1,0) \\ \eta_k, \epsilon_k &\sim \mathcal{N}(0,1) \end{aligned} \tag{2}$$

It will also be useful to use the following expression for y_k:

$$\begin{aligned} y_k &= \mathbf{CA}\mathbf{x}_{k-1} + \underbrace{\begin{bmatrix} \mathbf{Cb}\sqrt{\lambda_k} & d \end{bmatrix}}_{\mathbf{e}_k} \underbrace{\begin{bmatrix} \eta_k \\ \epsilon_k \end{bmatrix}}_{\mathbf{n_k}} \qquad (3) \\ \mathbf{n}_k &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ \lambda_k &\sim S_{\alpha/2}(1,1,0) \end{aligned}$$

This transforms our state equation form into a (nearly?) α-stable sub-gaussian form. (This is NOT in α-stable sub-gaussian form, see: $<$ α-stable sub-gaussian definition $>$ for reference)

Generic Rao-Blackwellised Particle Filter (RBPF)

For Rao-Blackwellised Particle Filtering, we partition the state vector into gaussian and non-gaussian components. We can then use standard Kalman Filtering to obtain optimal estimates for the gaussian state components, after obtaining estimates for the non-gaussian state components.

At each time step k, the Rao-Blackwellised Particle Filter produces for each time step k a set of weighted samples $\left\{w_k^{(i)}, \lambda_k^{(i)}, \mu_k^{(i)}, \Sigma_k^{(i)} : i = 1, ..., N\right\}$ according to:

1. Draw new latent variables $\lambda_t^{(i)}$ for each particle in $i = 1, ..., N$ from the corresponding importance distribution:

$$\lambda_k^{(i)} \sim \pi(\lambda_k | \lambda_{0:k-1}^{(i)}, y_{1:k}) \qquad (4)$$

For the generic RBPF, we choose the importance distribution:

$$\pi(\lambda_k | \lambda_{0:k-1}^{(i)}, y_{1:k}) = p(\lambda_k | \lambda_{0:k-1}^{(i)}) = S_{\alpha/2}(\lambda_k | 1, 1, 0) \qquad (5)$$

2. Calculate new weights as follows:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(y_k | \lambda_{0:k}^{(i)}, y_{1:k-1}) p(\lambda_k^{(i)} | \lambda_{k-1}^{(i)})}{\pi(\lambda_k^{(i)} | s_{0:k-1}^{(i)}, y_{1:k})} \qquad (6)$$

Here, the likelihood term $p(y_k | u_{0:k}^{(i)}, y_{1:k-1})$ is obtained using the predictive error decomposition from the Kalman Filter:

Kalman Filtering Prediction Step:

$$\begin{aligned} p(\mathbf{x}_{0:k}|\lambda_{0:k}^{(i)}, y_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k|\mu_k^{-(i)}, \Sigma_k^{-(i)}) \\ \text{where:} \qquad \mu_k^{-(i)} &= \mathbf{A}\mu_{k-1}^{(i)} \\ \Sigma_k^{-(i)} &= \mathbf{A}\Sigma_{k-1}^{-(i)}\mathbf{A}^T + \mathbf{b}^T\mathbf{b}\lambda_k^{(i)} \end{aligned} \tag{7}$$

Predictive Error Decomposition:

$$\begin{aligned} p(y_k|\lambda_{0:k}^{(i)}, y_{1:k-1}) &= \int p(y_k|\lambda_{0:k}^{(i)}, \mathbf{x}_{0:k})p(\mathbf{x}_{0:k}|\lambda_{0:k}^{(i)}, y_{1:k-1})d\mathbf{x}_{0:k} \\ &= \mathcal{N}(y_k|\mathbf{C}\mu_k^{-(i)}, \mathbf{C}\Sigma_k^{-(i)}\mathbf{C}^T + d^2) \end{aligned} \tag{8}$$

3. Perform Kalman Filter updates for each of the particles conditional on the drawn latent variables $\lambda_k^{(i)}$.

$$\begin{aligned} p(\mathbf{x}_{0:k}|\lambda_{0:k}^{(i)}, y_{1:k}) &= \mathcal{N}(\mathbf{x}_k|\mu_k, \Sigma_k) \\ \text{where:} \qquad \mathbf{v}_k^{(i)} &= y_k - \mathbf{C}\mu_k^{(i)} \\ \mathbf{S}_k^{(i)} &= \mathbf{C}\Sigma_k^{-(i)}\mathbf{C}^T + d^2 \\ \mathbf{K}_k^{(i)} &= \Sigma_k^{-(i)}\mathbf{C}^T\mathbf{S}_k^{-1} \\ \\ \mu_k^{(i)} &= \mu_k^{-(i)} + \mathbf{K}_k^{(i)}\mathbf{v}_k^{(i)} \\ \Sigma_k^{(i)} &= \Sigma_k^{-(i)} - \mathbf{K}_k^{(i)}\mathbf{S}_k^{(i)}[\mathbf{K}_k^{(i)}]^T \end{aligned} \tag{9}$$

4. Perform multinomial resampling to increase the number of effective particles.

Potential Problems with the RBPF

When $y_k - \mathbf{CA}\mu_k^{(i)}$ is large, the RBPF is often unable to get a good importance sampling estimate for λ_k.

When $y_k - \mathbf{CA}\mu_k^{(i)}$ is large, this implies that λ_k is likely to be large also. (See Figure 2, noting that $p(\lambda_k|y_k - \mathbf{CA}\mu_k^{(i)}) \propto S_{\alpha/2}(\lambda_k)\mathcal{N}(y_k - \mathbf{CA}\mu_k^{(i)}|\lambda_k))$. As a large λ_k lies in the low probability right tail of the particle proposal distribution given by Equation 5, very few particles $\lambda_k^{(i)}$ are generated from the particle proposal distribution which are close to the actual λ_k.

This problem is exacerbated by the fact that the α parameter of the proposal distribution is half of the original α. This causes the tails of the proposal distribution is to decay very slowly, increasing the number of particles needed to give a good importance sampling estimate.

This results in sample impoverishment, whereby there are only a few effective particles with non-negligible weights, which causes the performance of the RBPF to be slightly worse for very low numbers of particles.

One method of quantifying sample impoverishment in a particle filter (whilst adjusting for number of particles) is by measuring the entropy of the particle filter weights given in Equation 10.

$$H(w) = \sum_{i=1}^{N} w_i \log(w_i) \qquad \text{where: } \sum_{i=1}^{N} w_i = 1 \tag{10}$$

In order to compare the entropy of the particle weights across different number of particles, we instead use a normalised entropy measure given in Equation 11.This normalised entropy measures the change in entropy between a set of weights with uniform distribution (an ideal "optimized" set of weights) and a set of weights with a non-uniform distribution, and is scaled to be independent of the number of particles, as well as the base used in calculating the entropy.

$$H_n(w) = \frac{1}{\log_b(N)} \sum_{i=1}^{N} w_i \log_b(w_i) \qquad \text{where: } \sum_{i=1}^{N} w_i = 1 \tag{11}$$

We demonstrate the problem of sample impoverishment by simulating a single time step of the particle filter, for varying values of $y_k - \mathbf{CA}\mu_k^{(i)}$. Fixing $\mu_k^{(i)}$, $\Sigma_k^{(i)}$ whilst varying y_k and N, we simulate one time step of the RBPF update step described above and present the normalised entropy of the particle filter weights obtained in Figure 1. We see that the normalised entropy of the particle filter weights drops rapidly for large $y_k - \mathbf{CA}\mu_k^{(i)}$, and that the as N increases, the normalised entropy of the particle filter weights is more resistant to sample impoverishment at large values of $y_k - \mathbf{CA}\mu_k^{(i)}$ as expected.

Dense Sampling

We note that the cause of the problem of sample impoverishment is due to the low number of particles being drawn from the tails of the proposal distribution, leading to low numbers of particles with effective weights when $y_k - \mathbf{CA}\mu_k^{(i)}$ is large

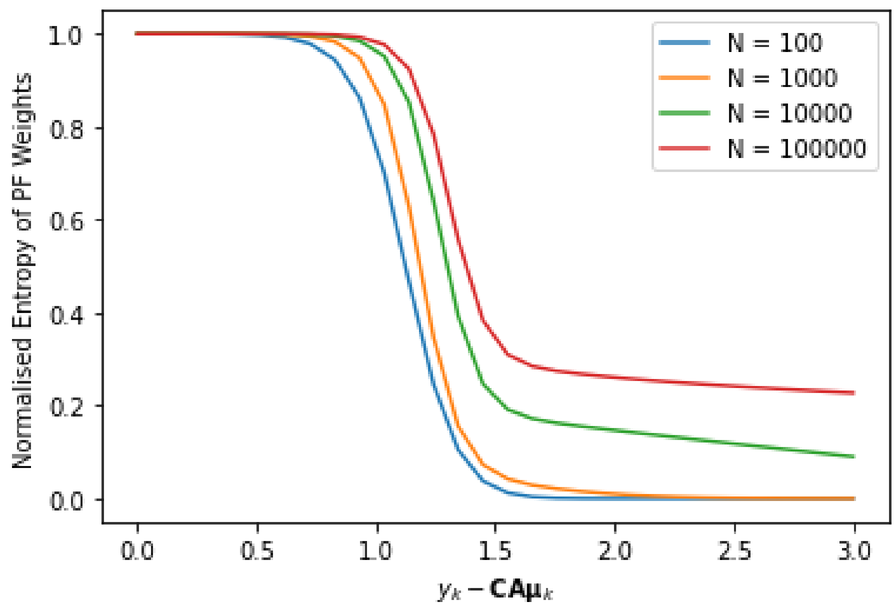

Figure 1: Change in normalised entropy of PF weights as $y_k - \mathbf{CA}\mu_k^{(i)}$ and N are varied

Thus, one method of reducing sample impoverishment in the particle filter would be to simply generate a larger number of particles from the tails of the proposal distribution. In order to preserve the required proposal distribution, we need to then scale the weights of the extra particles generated from the tails appropriately.

This leads to the following algorithm for sampling from the proposal distribution given in Equation 5:

- Define the hyperparameters ϵ_λ and m. In this example, we select $\epsilon_\lambda = 1 \times 10^{-1}$ and $m = 3$.
- Calculate the number of extra particles (N_{extra}) to draw from the right tail using:

$$threshold = \frac{f_{pareto}(\lambda_k|\alpha/2, 1) - S_{\alpha/2}(\lambda_k|1, 1, 0)}{S_{\alpha/2}(\lambda_k|1, 1, 0)} \tag{12}$$

$$N_{extra} = P(\lambda > threshold) \times Nm \tag{13}$$

- Draw N_{extra} particles from the right tail:

$$\lambda_k^{(i)} \sim f_{bounded\ pareto}(\lambda_k|\alpha = \alpha/2, \beta = 1, L = threshold) \qquad i = 1...N_{extra} \tag{14}$$

- Draw $N - N_{extra}$ particles from the truncated alpha stable distribution using a simple rejection sampler:
  - Draw $\lambda_k^{(i)} \sim S_{\alpha/2}(\lambda_k|1,1,0)$
  - Accept if $\lambda_k^{(i)} < threshold$
- Alter the weights of the particles drawn from the tails:

$$w_i = \frac{1}{m} w_i \qquad \text{for: } i \in [1, N_{extra}]$$

Adaptive Sampling

One way of attempting to improve the performance of the RBPF is to form a better importance sampling distribution. In this section, we seek to form a better importance sampling distribution for $\pi(\lambda_t|y_{1:t}, \lambda_{0:t-1})$.

For optimality, we want $\pi(\lambda_t|y_{1:t}\lambda_{0:t-1}^{(i)}) \approx p(\lambda_t|y_{1:t}, \lambda_{0:t-1}^{(i)})$.

It is hard to formulate an analytical form for this density directly. To get around this, we can instead form the analytical joint distribution for $p(y_t, \lambda_t|y_{1:t-1}, \lambda_{0:t-1}^{(i)})$. The complete derivation is given in the appendix, with the key steps highlighted below.

We first augment the probability density given above with $x_{0:t-1}$. This allows us to use results from the previously-performed Kalman Filter step.

$$\begin{aligned}
& p(y_t.\lambda_t, x_{0:t-1}|y_{1:t-1}, \lambda_{0:t-1}^{(i)}) \\
\propto\ & p(y_t|y_{1:t-1}, \lambda_{0:t}^{(i)}, x_{0:t-1}) p(x_{0:t-1}|y_{1:t-1}, \lambda_{0:t}^{(i)}) p(\lambda_t^{(i)}|\lambda_{0:t-1}^{(i)}) \\
\approx\ & p(y_t|y_{1:t-1}, \lambda_{0:t}^{(i)}, x_{0:t-1}) p(x_{0:t-1}|y_{1:t-1}, \lambda_{0:t-1}^{(i)}) p(\lambda_t|\lambda_{0:t-1}) \\
=\ & \mathcal{N}(y_t|\mathbf{CA}\mathbf{x}_{t-1}, \mathbf{e}\mathbf{e}^T) \mathcal{N}(\mathbf{x}_{0:t-1}|\mu_k^{(i)}, \mathbf{\Sigma}_k^{(i)}) S_{\alpha/2}(1,1,0) \\
=\ & \mathcal{N}\left(\begin{bmatrix} y_t \\ \mathbf{x}_{t-1} \end{bmatrix} \middle| \begin{bmatrix} \mathbf{CA}\mu_{k-1}^{(i)} \\ \mu_{k-1}^{(i)} \end{bmatrix}, \begin{bmatrix} \mathbf{e}\mathbf{e}^T + (\mathbf{CA})\mathbf{\Sigma}_k^{(i)}(\mathbf{CA})^T & \mathbf{CA}\mathbf{\Sigma}_k^{(i)} \\ \mathbf{\Sigma}_k^{(i)}(\mathbf{CA})^T & \mathbf{\Sigma}_k^{(i)} \end{bmatrix} \right) S_{\alpha/2}(1,1,0)
\end{aligned}$$

We can then obtain the required density $p(y_t, \lambda_t^{(i)}|y_{1:t-1}, \lambda_{0:t-1})$ by marginalising out $x_{0:t-1}$:

$$
\begin{aligned}
p(y_k, \lambda_k | y_{1:k-1}, \lambda_{0:k-1}^{(i)}) &= \int p(y_k, \lambda_k, x_{0:k-1} | y_{1:k-1}, \lambda_{0:k-1}^{(i)}) dx_{0:k-1} \\
&= \mathcal{N}(y_k | \mathbf{CA}\mu_k^{(i)}, (\mathbf{Cb})(\mathbf{Cb})^T \lambda_k + d^2 + (\mathbf{CA})\mathbf{\Sigma}_k^{(i)}(\mathbf{CA})^T) S_{\alpha/2}(\lambda_t | 1, 1, 0) \\
&= \mathcal{N}(y_k | \mathbf{CA}\mu_k^{(i)}, \sigma_\lambda \lambda_k + \mu_\lambda) S_{\alpha/2}(\lambda_t | 1, 1, 0) \qquad (15)
\end{aligned}
$$

where:

$$
\begin{aligned}
\sigma_\lambda &= (\mathbf{Cb})(\mathbf{Cb})^T \\
\mu_\lambda &= d^2 + (\mathbf{CA})\mathbf{\Sigma}_k^{(i)}(\mathbf{CA})^T
\end{aligned}
$$

By fixing the value of y_k in Equation 15, we can obtain the desired distribution $p(\lambda_k | y_{1:k}, \lambda_{0:k-1})$.

This formulation of the joint distribution also gives us an alternative interpretation for the sampling density $p(\lambda_t | y_{1:t}, \lambda_{0:t-1})$.

We can reinterpret this as finding the posterior density of λ_k given the observations $y_t - \mathbf{CA}\mu_k$. The prior on λ_k is the α-stable proposal distribution $S_{\alpha/2}(\lambda_k | 1, 1, 0)$ while the likelihood is the unknown variance of a normal distribution ($\mathcal{N}(y_t | \mathbf{CA}\mu_k, \sigma_\lambda \lambda_t + \mu_\lambda)$).

Rejection Sampling

In general, there is no closed form expression for this distribution. However, we can utilise the fact that we can draw samples from the prior distribution easily using the Chamber-Mallow-Stuck method [**chambers1976method**]. This can be used as a suitable proposal function for rejection sampling.

We adapt the methods of [**godsill1999bayesian**] to draw samples from this distribution.

The target distribution for the rejection sampling scheme is given by:

$$
\begin{aligned}
f(y_t, \lambda_t | y_{1:t-1}, \lambda_{0:t-1}^{(i)}) &= \mathcal{N}(y_t | \mathbf{CA}\mu_k^{(i)}, \sigma_\lambda \lambda_t + \mu_\lambda) S_{\alpha/2}(\lambda_t | 1, 1, 0) \\
&= \mathcal{N}(y_t - \mathbf{CA}\mu_k^{(i)} |, \sigma_\lambda \lambda_t + \mu_\lambda) S_{\alpha/2}(\lambda_t | 1, 1, 0)
\end{aligned}
$$

Using a proposal distribution $g(\lambda_t | y_{1:t}, \lambda_{0:t-1}) = S_{\alpha/2}(1, 1, 0)$, we can use the likelihood as a valid rejection function as it is bounded from the above:

$$
\begin{aligned}
\mathcal{N}(y_t - \mathbf{C}\mathbf{A}\mu_k^{(i)}|0, \sigma_\lambda \lambda_t + \mu_\lambda) &\leq \frac{f(y_t, \lambda_t | y_{1:t-1}, \lambda_{0:t-1})}{g(y_t, \lambda_t | y_{1:t-1}, \lambda_{0:t-1})} \\
&= M
\end{aligned}
$$

$$
M = \begin{cases} \frac{1}{\sqrt{2\pi (y_t - \mathbf{C}\mathbf{A}\mu_k^{(i)})^2}} \exp(-0.5) & (y_t - \mathbf{C}\mathbf{A}\mu_k^{(i)})^2 \geq \mu_\lambda \\ \frac{1}{\sqrt{2\pi d^2}} \exp\left(-\frac{(y_t - \mathbf{C}\mathbf{A}\mu_k^{(i)})^2}{2d^2}\right) & (y_t - \mathbf{C}\mathbf{A}\mu_k^{(i)})^2 \leq \mu_\lambda \end{cases} \tag{16}
$$

This gives a suitable rejection sampler as:

1. Draw $\lambda_k \sim S_{\alpha_2}(1, 1, 0)$
2. Draw $u \sim U(0, M)$
3. If $u \geq \mathcal{N}\left(y_t - \mathbf{C}\mathbf{A}\mu_k^{(i)}|0, \sigma_\lambda \lambda_k + \mu_\lambda\right)$, reject λ_t and go to Step (a)

Improvements on the rejection sampler:

The rejection sampler suffers from poor acceptance rates when $y_k - \mathbf{C}\mathbf{A}\mu_k^{(i)}$ is very large. Most of the samples generated by the prior fall into the left tail of the likelihood, and have extremely low probabilities of being accepted (Figure 2).

Due to the additional observation noise term (d), we are unable to use the Inverse Gamma approximation to the full posterior as detailed in [**godsill1999bayesian**]. Instead, we attempt to improve the sampling efficiency by truncating the left tail of the likelihood distribution.

Due to the intractability of calculating the Cumulative Distribution Function (CDF) of the likelihood analytically, we left-truncate the likelihood where it is smaller than a chosen epsilon. In this project, this small epsilon was chosen to be $\epsilon_{trunc} = 1 \times 10^{-9}$. Further checks are also performed to ensure that this truncation only occurs well into the right tail of the prior.

This allows us to generate samples from a left-truncated form of the α-stable prior instead of the full prior. These generated samples are much more likely to be in the high probability regions of the likelihood, thus improving our sampling efficiency greatly.

We sample from the left truncated α-stable distribution by applying a paretian tail approximation to the α-stable distribution , then sampling from the paretian tail approximation using **??**.

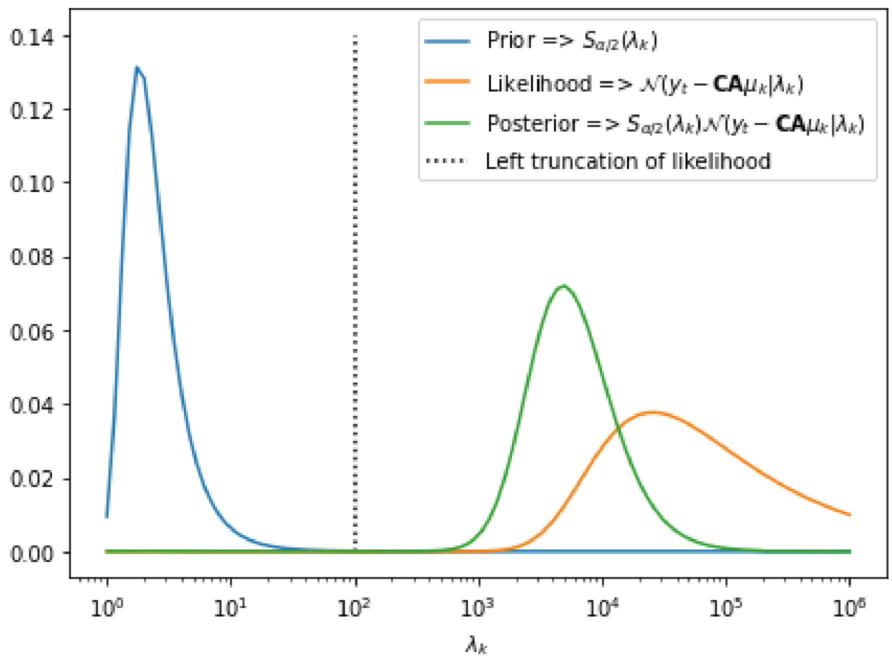

Figure 2: Illustration of the poor acceptance rate regime of prior/likelihood interaction

The improved rejection sampler can be described as:

1. Define hyperparameter ϵ_{trunc}
2. Calculate 95th percentile of the prior ($\lambda_{0.95}$) where: $\int_0^{\lambda_{0.95}} S_{\alpha/2}(\lambda_k|1,1,0) = 0.95$
3. Draw proposals $\lambda_k^{(i)}$
   - Calculate a possible truncation value $\lambda_{k,trunc}$ where $\mathcal{N}(y_k - \mathbf{CA}\mu_k^{(i)}|\lambda_{k,trunc}) = \epsilon_{trunc}$.
   - If this truncation value also lies in the right tail of the prior (i.e. $\lambda_{k,trunc} > \lambda_{0.95}$) then sample particle from the paretian tail approximation

$$\lambda_k^{(i)} \sim f_{bounded\ pareto}(x|\alpha/2, \beta = 1, \lambda_{k,trunc})$$

   - Otherwise, sample particle from the original prior:

$$\lambda_k^{(i)} \sim S_{\alpha/2}(1,1,0)$$

4. Draw $u \sim U(0, M)$

5. If $u \geq \mathcal{N}\left(y_k - \mathbf{CA}\mu_k^{(i)} | 0, \sigma_\lambda \lambda_k + \mu_\lambda\right)$, reject $\lambda_k^{(i)}$ and go to Step (a)

List of Figures

List of Tables