THE UNIVERSITY OF HONG KONG

STAT8002 PROJECT

Computational Methods for Statistical Inference of Dynamical Models

Extreme Quantile Estimation of the Cox-Ingersoll-Ross Model

Author:
Yun ZHANG
Uid:
3035802483

Supervisor:
Prof. Wentao LI

Department of Statistics & Actuarial Science

April 30, 2021

Computational Methods for Statistical Inference of Dynamical Models

Extreme Quantile Estimation of the Cox-Ingersoll-Ross Model

Abstract: The sequential Monte Carlo (SMC) algorithm, which is also called Particle filters, is a class of well-established algorithms for estimating the distribution of latent states variable of a state space model given the observations. It has been proven successful on the estimation of posterior mean, but to accurately estimate the extreme quantiles of the filtering distribution is computationally intensive. The main objective of this report is to present a novel strategy based on the generalized (mixture) SMC framework to estimate the extreme quantiles efficiently, along with a comprehensive review of a few particle filter algorithms. In purpose of demonstration, the Cox-Ingersoll-Ross (CIR) model, a famous interest short rate model, is studied as an example. The simulation results show that, compared to other algorithms, the proposed method provides a better estimates about extreme quantiles.

Keywords: *State-space model, sequential Monte Carlo algorithm, CIR model, extreme quantile estimate*

Contents

Chapter 1

Introduction

1.1 State-space model

The state-space model is a kind of probabilistic models where the latent variables (unobserved states) and data (observations) satisfy the Markov property. Such model has found numerous applications in diverse fields, like statistics, especially time series (Aoki, 2013), ecology (Aeberhard et al., 2018 and Buckland et al., 2004), econometric (Geyer and Pichler, 1999), and engineering (Miranda et al., 2009), to model dynamic systems with time-varying inputs and outputs.

Formally, we can define a state-space model by two stochastic processes, $\{X_t\}_{t\geq 0}$ and $\{Y_t\}_{t\geq 0}$. Let x_t be the unobserved state variable and y_t be the observation at the t-th time step. The model is given by the transition density p and observation density f,

$$\begin{aligned} x_0 &\sim p(x_0), \\ x_t \mid x_{t-1} &\sim p\left(x_t \mid x_{t-1}\right), \\ y_t \mid x_t &\sim f\left(y_t \mid x_t\right). \end{aligned} \tag{1.1}$$

For such models, we always assume that the observations y_t are independent, given their corresponding latent states x_t. So we have the following graphical illustration

$$\begin{array}{ccccccccc} \ldots & \rightarrow & \mathbf{X}_{n-1} & \rightarrow & \mathbf{X}_n & \rightarrow & \mathbf{X}_{n+1} & \rightarrow & \ldots \\ & & \downarrow & & \downarrow & & & & \\ \ldots & & \mathbf{Y}_{n-1} & & \mathbf{Y}_n & & \mathbf{Y}_{n+1} & & \ldots \end{array}$$

In a state-space model, the inferences about $X_{0:t}$ given a series of realization of the observations $Y_{0:t} = y_{0:t}$ rely on the posterior distribution, which is also called filtering distribution,

$$p(x_{0:n} \mid y_{0:n}) = \frac{p(x_{0:n},\, y_{0:n})}{p(y_{0:n})}, \tag{1.2}$$

where

$$\begin{aligned} p\left(x_{0:n},\, y_{0:n}\right) &= p\left(x_{0:n}\right) p\left(y_{0:n} \mid x_{0:n}\right), \\ \text{and } p(y_{0:n}) &= \int p\left(x_{0:n},\, y_{0:n}\right) dx_{0:n}. \end{aligned}$$

are the joint probability density function (pdf) and likelihood function. The notation $x_{0:t}$ and $y_{0:t}$ represent the random vector $(x_0, \ldots, x_t)$ and observations $(y_0, \ldots, y_t)$ at time t.

Commonly there are three types of inference tasks for state-space models, according to the given observations and the target posterior distribution. Problems related to the sequential approximation of the filtering distributions $\{p(x_{1:t}|y_{1:t}\}_{t\geq 1}$, or the marginal distributions $\{p(x_t|y_{1:t}\}_{t\geq 1}$ are refereed as the optimal filtering problem, whose objective is to update the knowledge of latent state variables when obtaining new observation. The smoothing problem pursues to estimate previous states with all available observations including those obtained later. The predicting problem is about forecasting the future states with currently available observations. Note that in these problems, the parameters of state-space model are assumed already known, the inference is only about the states x_t. Estimating unknown parameters is a completely different problem. The filtering problem with respect to extreme quantiles is the one this report trying to address.

For a linear Gaussian model, whose states and observations are both linearly Gaussian distributed, the posterior distribution is also Gaussian with easy-to-compute parameters. But for the majority of non-linear and non-Gaussian models, usually we cannot express the posterior distribution in closed-form and obtain analytic solutions to the inference problems of interests. So we need to use numerical methods to make approximation. And the Sequential Monte Carlo (SMC) algorithm was proposed to provide approximate solutions to these intractable inference problems.

The remainder of this report has the following structure, remaining part of this chapter reviews that how the idea of Sequential Monte Carlo algorithms are developed from the basic importance sampling method and briefly introduces the Cox-Ingersoll-Ross model. In Chapter 2, some previous studies on rare events are outlined, along with our novel strategy based on the mixture SMC, where part of the particles are directly generated from the tails of the proposal density. In Chapter 3, as an application, the method is demonstrated through simulation examples and conclusions are drawn.

1.2 Sequential Monte Carlo Methods

The central problem related to the state-space model is to make inference about the unobserved states x_t conditional on the observations y_t, in respect of the filtering density $p(x_t|y_t)$. Since the first introduction in Gordon et al., 1993, the SMC algorithms, also called particle filters, have turned out to be a very popular class of Bayesian numerical methods to solve this problem. They use weighted samples (also called particles) $\left\{\left(\tilde{x}^{(j)}_{0:t-1}; \widetilde{w}^{(j)}_{t-1}\right)\right\}_{j=1}^{N}$ to sequentially approximate the posterior distribution and make inferences based on them, where N is the size of particle samples drawn in each step. The SMC methods have been widely accepted and used in numerous fields, including

navigation and tracking (Stone et al., 2013), estimation of economic models (Jasra and Del Moral, 2011), and signal processing (Wang et al., 2002).

For a state space model of the form (1.1), the best estimator in the sense of Bayesian statistics for the unknown states is the posterior mean,

$$\mu_t \equiv E[x_{0:t}|y_{0:t}] = \int x_{0:t} p(x_{0:t}|y_{0:t}) \mathrm{d}x_{0:t} \tag{1.3}$$

Since the joint probability distribution, which can be regarded as the unnormalized posterior distribution, satisfies

$$\begin{aligned} p\left(x_{0:t}, y_{0:t}\right) &= \prod_{n=0}^{t} f(y_n|x_n) p(x_0) \prod_{n=1}^{t} p(x_n|x_{n-1}) \\ &= p\left(x_{0:t-1},\, y_{0:t-1}\right) f\left(y_t \mid x_t\right) p\left(x_t \mid x_{t-1}\right), \end{aligned} \tag{1.4}$$

therefore the filtering density follows this recursion:

$$\frac{p\left(x_{0:t} \mid y_{0:t}\right)}{p\left(x_{0:t-1} \mid y_{0:t-1}\right)} = \frac{p\left(x_t \mid x_{t-1}\right) f\left(y_t \mid x_t\right)}{p\left(y_t \mid y_{0:t-1}\right)}, \tag{1.5}$$

and

$$p\left(y_t \mid y_{0:t-1}\right) = \int p\left(x_{t-1} \mid y_{0:t-1}\right) f\left(y_t \mid x_t\right) p\left(x_t \mid x_{t-1}\right) dx_{t-1:t}.$$

Usually we are more attracted by the marginal density $p(x_t|y_{1:t})$, derived by integrating out $x_{0:t-1}$ in (1.5),

$$p\left(x_t \mid y_{0:t}\right) = \frac{f\left(y_t \mid x_t\right) p\left(x_t \mid y_{0:t-1}\right)}{p\left(y_t \mid y_{0:t-1}\right)}, \tag{1.6}$$

where

$$p\left(x_t \mid y_{0:t-1}\right) = \int p\left(x_{t-1} \mid y_{0:t-1}\right) p\left(x_t \mid x_{t-1}\right) dx_{t-1}.$$

The SMC algorithms are theoretically based on above two-step recursions, which are called updating and predicting steps in Doucet and Johansen, 2009.

Importance sampling (IS) is a basic Monte Carlo method and foundation of SMC methods. It draws samples from a so-called proposal (importance) density $q_t(x_{0:t})$, to approximate the target distribution. And the proposal is required to be able to envelope the target density, i.e. the filtering density, for all points in the support,

$$p(x_{0:t}|y_{0:t}) > 0 \Rightarrow q_t(x_{0:t}) > 0.$$

In particular, it is easier to generate samples from the proposal we selected than the target distribution. Suppose we have drawn N independent samples $X_{0:t}^i \sim q_t(x_{0:t})$ for $i = 1, \ldots, N$, then by inserting the proposal density in (1.3), we can estimate the posterior mean by

$$\mu_t^{IS} = \int \frac{x_{0:t} p(x_{0:t}|y_{0:t})}{q_t(x_{0:t})} q_t(x_{0:t}) \mathrm{d}x_{0:t} = \sum_{i=1}^{N} W_t^i x_{0:t}^{(i)},$$

where

$$W_t^i = \frac{w_t(x_{0:t}^i)}{\sum_{j=1}^N w_t(x_{0:t}^{(j)})}, \text{ and } w_t(x_{0:t}^{(i)}) = \frac{p(x_{0:t}, y_{0:t})}{q_t(x_{0:t})}$$

are the normalized and unnormalized weights. Meanwhile, the likelihood $Z_t = p(y_{0:t}) = \int w_t(x_{0:t}) q_t(x_{0:t}) \mathrm{d}x_{0:t}$ can be estimated by the average of unnormalized weights, $\hat{Z}_t = \frac{1}{N} \sum_{i=1}^N w_t(x_{0:t}^{(i)})$, which has the variance

$$\mathrm{Var}_{IS}(\hat{Z}_t) = \frac{Z_t^2}{N} \left(\int \frac{p(x_{0:t}|y_{0:t})^2}{q_t(x_{0:t})} \mathrm{d}x_{0:t} - 1 \right).$$

We can see that the variance of likelihood is unbounded as t increases, and consequently the variance of the estimates will increase exponentially. Furthermore, as mentioned in (Doucet and Johansen, 2009), when we try to obtain a sequential version of the algorithm, there will be a conflict between the selection of optimal importance density at time t and the consistency of the marginal distributions of states of fixed time steps, since the optimal importance density for estimating $x_{0:t-1}$ will not be, or even similar to, the marginal density of $x_{0:t-1}$ in the optimal density for estimating $x_{0:t}$. Moreover, the computational complexity of sequentially sampling from $q(x_{0:t})$, for each value of t, would increase at least linearly with t because samples are drawn from distributions with higher and higher dimensions. Therefore, in sequential importance sampling (SIS) algorithm, a proposal distribution with following recursion structure is employed,

$$q_t(x_{0:t}) = q_{t-1}(x_{0:t-1}) q_t(x_t|x_{0:t-1}) = q_0(x_0) \prod_{k=1}^t q_k(x_k \mid x_{0:k-1}), \tag{1.7}$$

so as to admit a definite computational complexity for the sequentially sampling scenarios. In particular, we draw $X_0^{(i)} \sim q_0(x_0)$ at initialization, then $X_t^{(i)} \sim q_t(x_t|X_{0:t-1}^{(i)})$ at step t for $t = 1, 2, \ldots, T$. The corresponding unnormalized weights can also be computed in a recursive way,

$$\begin{aligned}
w_0(x_0) &= \frac{f(y_0|x_0)p(x_0)}{q_0(x_0)}, \\
w_t(x_{0:t}) &= \frac{p(x_{0:t}, y_{0:t})}{q_t(x_{0:t})} \\
&= \frac{p(x_{0:t-1}, y_{0:t-1})}{q_{t-1}(x_{0:t-1})} \frac{p(x_{0:t}, y_{0:t})}{p(x_{0:t-1}, y_{0:t-1}) q_t(x_t|x_{0:t-1})} \\
&= w_{t-1}(x_{0:t-1}) \cdot \alpha_t(x_{0:t}) \\
&= w_0(x_0) \prod_{j=1}^t \alpha_j(x_{0:j}) \text{ for } t > 0,
\end{aligned}$$

where $\alpha_t(x_{0:t})$ is the incremental importance weight. For the selection of proposal, the posterior density $p(x_t|x_{0:t-1}, y_{0:t}) \propto p(x_t|x_{0:t-1}) p(y_{0:t}|x_{0:t})$ is the optimum for reducing the variances of weights $w_t(x_{0:t})$. But generating samples from this density is not

achievable all the time, in which cases we need make approximation of it for the proposal. Since SIS is just a special version of IS, the problem of estimates' exponentially increasing variance also exists. After a few iterations, a few particles (samples) dominate the whole sample and most particles have ignorable weights. Consequently, the algorithm lose its variability and will quickly fall to degeneracy.

To solve this problem, SMC algorithm adds a resampling step into the SIS algorithm. Actually, the weighted samples in SIS are drawn from $q_t(x_{0:t})$ and cannot provide guaranteed approximations of $p(x_{0:t}|y_{0:t})$. So, to obtain approximated samples from the filtering density, we simply sample from its IS empirical approximation,

$$\hat{p}(x_{0:t}|y_{0:t}) = \sum_{i=1}^{N} W_t^i \delta_{X_{0:t}^{(i)}}(x_{0:t}),$$

which is identical to sample from $X_{0:t}^{(i)}$ with replacement and with probability W_t^i, the normalized weights. Meanwhile, the problems related with variances are solved as well. Such operation is called resampling because we do sampling on a sample which itself is obtained by sampling. In this step, samples with tiny weights are dropped and those having large weights are duplicated, so that we can focus our computational efforts on regions of high probability mass. After resampling, equal weights $1/N$ are assigned to each sample. Among various resampling methods, the multinomial resampling is commonly used in the SMC algorithms because of its simplicity, which is based on the core idea of the bootstrap method: draw N samples from the multinomial distribution of parameters $(N, W_t^{1:N})$. Douc and Cappe, 2005 provided a detailed comparison of different resampling schemes. For a SMC algorithm, we can set certain criterion to determine whether resampling is needed in each step, since it's more sensible and efficient to do resampling only if the variance of unnormalized weights exceed a certain threshold. The most popular one is the Effective Sample Size (ESS) which measures the variability of weights,

$$\text{ESS}_t = \left(\sum_{i=1}^{N} (W_t^i)^2 \right)^{-1}.$$

The ESS ranges from 1 to N, and we resample only when it is lower than the threshold value which is usually set to be $N/2$. A larger ESS indicates more information in samples. The posterior mean μ_t is estimated immediately after the weighting step, using particles not yet resampled and the same formula as the importance sampling algorithm.

Chopin and Papaspiliopoulos, 2020 recently introduced the so-called Feynman-Kac models which serves as a mid-layer connecting the state-space model and SMC algorithm. It has been found useful in the programming of state space models and SMC algorithms. Under the three-layer architecture, the transition density, observation density, the proposals as well as their parameters are defined in a state space model object. Then an intermediate Feynman-Kac model layer wraps the state space model and

controls the propagation, weights computation, and all other operations in SMC algorithms, which gives the algorithm a lot flexibility and customization. Finally, the Feynman-Kac model is packed in the SMC object, together with other settings about SMC algorithm including the particle size, resampling method, and resampling conditions.

In practice, almost all SMC type algorithms can be implemented in such architecture. According to the different settings of proposal distribution and weights computation, there are several specifications of SMC algorithms. We will introduce three of them in following: Bootstrap filter, guided particle filter and mixture SMC algorithms.

1.2.1 Bootstrap filter

The Bootstrap filter is the most basic Sequential Monte Carlo algorithm, using the transition density as proposal. And therefore its unnormalized weights are simply the observation densities. It has the following three steps, for $j = 1, \ldots, N$,

1. Propagate: Draw $x_t^{(j)}$ from the proposal $q(x_t|\tilde{x}_{t-1}^{(j)}) = p(x_t|\tilde{x}_{t-1}^{(j)})$.
2. Reweight: Set $w_t^{(j)} \propto W_t^j = f\left(y_t|x_t^{(j)}\right)/N$, where $f\left(y_t|x_t^{(j)}\right)$ is the observation density of y_t given $x^{(j)}$.
3. Resample: If the resampling condition is met, set $\tilde{x}_t^{(j)} = x_t^{(A(j))}$ where $P(A(j) = i) = w_t^{(i)}$ for $j = 1, \ldots, N$.

Interpretation of Bootstrap filter is quite intuitive: at time t, we generate particles from the transition density, then weight them according to their compatibility to the current observation y_t, and resample when needed. The only requirement for this algorithm is the ability to draw samples from the transition density and to compute observation density point-wisely. However, its drawbacks are also obvious, particles X_t are "blindly" generated from transition density (exploration of the state space uses no information from the observations) and we cannot guarantee that they are compatible with the observations, especially when the observations are very informative.

1.2.2 Guided particle filter

As its name suggests, particles are directed by well-designed proposal distributions to the areas where weights are supposed to be high, rather than just following the transition density like the bootstrap filter. A guided particle filter has the following steps: at time step t, suppose we have the particles and their weights $\left\{\left(\tilde{x}_{0:t-1}^{(j)}; \widetilde{w}_{t-1}^{(j)}\right)\right\}_{j=1}^{N}$. Then for $j = 1, \ldots, N$,

1. Propagate: Draw $x_t^{(j)}$ from the proposal distribution $q(x_t|\tilde{x}_{0:t-1}^{(j)})$ and let $x_{0:t}^{(j)} = (\tilde{x}_{0:t-1}^{(j)}, x_t^{(j)})$.
2. Weighting: Set $w_t^{(j)} = \widetilde{w}_{t-1}^{(j)} p(x_t^{(j)} \mid \tilde{x}_{t-1}^{(j)}) p(y_t \mid x_t^{(j)}) / q(x_t^{(j)} \mid \tilde{x}_{0:t-1}^{(j)})$.

3. Resample: When the resampling condition is met, resample $\{x_{0:t}^{(j)}\}_{j=1}^{N}$ by the weights $\left\{w_t^{(j)}\right\}_{j=1}^{N}$ to get new weighted samples $\left\{\left(\tilde{x}_{0:t}^{(j)}; \tilde{w}_t^{(j)}\right)\right\}_{j=1}^{N}$, with $\tilde{w}_t^{(j)} = 1/N$. Otherwise, just let $\left(\tilde{x}_{0:t}^{(j)}; \tilde{w}_t^{(j)}\right) = (x_{0:t}^{(j)}; w_t^{(j)})$.

4. Likelihood estimation: $\hat{p}(y_{1:t}) := \hat{p}(y_0)\prod_{k=1}^{t}\hat{p}(y_k|y_{1:k-1})$, where $\hat{p}(y_k|y_{1:k-1}) = \frac{1}{N}\sum_{j=1}^{N} w_k^{(j)}$ is an estimate of $p(y_k|y_{1:k-1})$ computed at time k.

Just before the resampling step, the posterior mean could be estimated by

$$\mu_t^{SMC} = \frac{\sum_{i=1}^{N} x_{1:t}^{(j)} w_t^{(j)}}{\sum_{i=1}^{N} w_t^{(j)}}.$$

For the selection of proposal density, it is known that the optimal one which reduces variances of weights to zero is conditioned on both states and observations,

$$q(x_t|x_{1:t-1}, y_t) = p(x_t|x_{1:t-1}, y_t) \propto p(y_t|x_t)p(x_t|x_{1:t-1}). \tag{1.8}$$

To utilize the optimal proposal, it needs the capability of sampling from $p(x_t|x_{1:t-1}, y_t)$, which is hard for the state space models whose densities are non-linear and non-Gaussian. There are many approaches to approximate the optimal proposal, for example, if $p(x_t|x_{1:t-1}, y_t)$ is unimodal and twice differentiable with respect to x_t, we can use the local linearization around the mode to construct a normal density. But for more complex cases, it's a better choice to adopt more than one proposals, which leads us to the mixture SMC algorithm.

1.2.3 Mixture SMC

For a SMC algorithm, it's possible and reasonable to include multiple proposals. An intuitive idea is to use both the state equation and a proposal linked to the new observations. For example, (Orton and Fitzgerald, 2002 and Vermaak et al., 2005) used a mixture proposal of transition density and multiple components accounting for particular measurements, in the application of multi-target tracking with multi-dimension observations.

Algorithm of mixture SMC is similar to that of guided particle filters, suppose that the mixture proposal density is $q_{\boldsymbol{\alpha}}(x_t|x_{1:t-1}) = \sum_{k=1}^{p}\alpha_k q_k(x_t|x_{1:t-1})$ given p component proposals $q_k(x_t|x_{1:t-1})$ for $k = 1, \ldots, p$ and the vector of mixture proportions $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_p)$. At time t, assume we have the particles and weights $\left\{\left(\widetilde{x}_{0:t-1}^{(j)}; \tilde{w}_{t-1}^{(j)}\right)\right\}_{j=1}^{N}$ and indicator set $\mathbf{I}_p$ which records the current category of each particle. Then for $j = 1, \ldots, N$,

1. Propagate: Sample $x_t^{(j)}$ from the proposal density $q_k(x_t|\widetilde{x}_{0:t-1}^{(j)})$, $(I_p(j) = k, k = 1, \ldots, p)$ and let $x_{0:t}^{(j)} = (\widetilde{x}_{0:t-1}^{(j)}, x_t^{(j)})$.

2. Weighting: Assign $x_{0:t}^{(j)}$ with weight

$$w_t^{(j)} = \widetilde{w}_{t-1}^{(j)} \frac{p\left(x_t^{(j)} \mid \widetilde{x}_{t-1}^{(j)}\right) p\left(y_t \mid x_t^{(j)}\right)}{q_{\boldsymbol{\alpha}}\left(x_t^{(j)} \mid \widetilde{x}_{0:t-1}^{(j)}\right)}.$$

3. Resampling: If the resampling condition is met, resample $\{x_{0:t}^{(j)}\}_{j=1}^{N}$ according to the $\left\{w_t^{(j)}\right\}_{j=1}^{N}$ to acquire latest particles $\left\{\left(\tilde{x}_{0:t}^{(j)}; \tilde{w}_t^{(j)}\right)\right\}_{j=1}^{N}$ with weight $1/N$, and shuffle the indicator set. If the resampling condition is not satisfied, just let $\left(\tilde{x}_{0:t}^{(j)}; \tilde{w}_t^{(j)}\right) = (x_{0:t}^{(j)}; w_t^{(j)})$.

To implement the mixture SMC algorithm in the Feynman-Kac framework, the following adaptations are made:

- define the p proposals inside the state space model object,
- specify the selection of proposal and the corresponding proportions in the FeynmanKac level object,
- store and shuffle the indicator sets in the SMC level object.

Figure 1.1 presents the architecture of the algorithm's realization under the Feynman-Kac framework. And the new strategy we propose later will rely a lot on the mixture SMC algorithm.

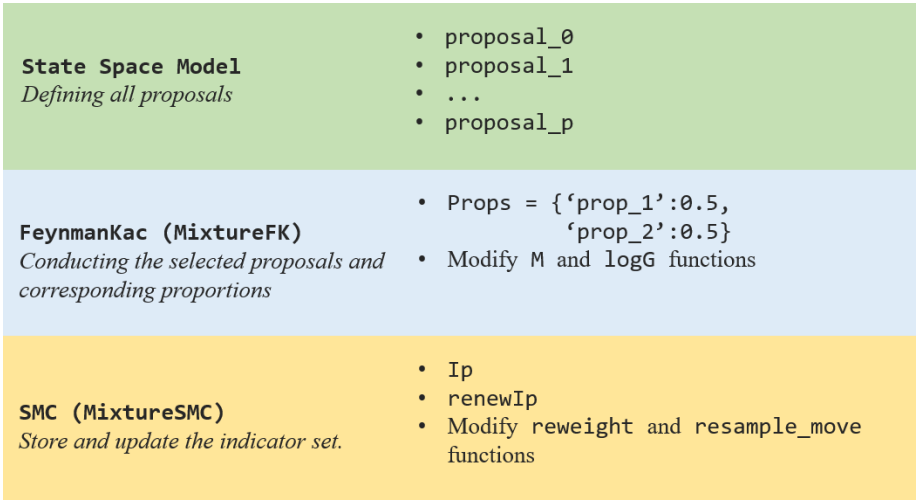

Figure 1.1: Architecture of the mixture SMC under the three-layer Feynman-Kac framework.

There are many other algorithms based on the SMC framework besides the three reviewed above, for example, the Kalman Filters, the auxiliary particle filters, the robust filters, and the hybrid filters, etc. But considering the applicability on the problem we discuss later, the three algorithms are mainly used in this report.

1.3 CIR model

The Cox-Ingersoll-Ross (CIR) model, firstly proposed by Cox et al., 1985, describes the evolution of interest rate (short rate) r_t by a square-root stochastic process. It satisfies the following stochastic differential equation (SDE),

$$\mathrm{d}r_t = \kappa\left(\theta - r_t\right)\mathrm{d}t + \sigma\sqrt{r_t}\mathrm{d}Z_t, \tag{1.9}$$

where θ, κ, and σ are the long-term mean of r_t, the mean reversion parameter, and the volatility parameter respectively, and Z_t is an independent Brownian motion process at all moments. As suggested by the SDE, the CIR model has the mean-reversion property and guarantees a nonnegative short rate (for $2\kappa\theta \geq \sigma^2$, the solution is strictly positive), that's why the model is popular in academic literature and among practitioners. The SDE has no explicit solution but the distribution of the solution is known, with

$$d = \frac{4\kappa\theta}{\sigma^2}, \quad g(t) = \frac{4\kappa e^{-\kappa t}}{\sigma^2\left(1 - e^{-\kappa t}\right)},$$

$e^{\kappa t}g(t)r_t$ is known to be a non-central χ^2 random variable with non-centrality parameter $r_0 g(t)$ and degrees of freedom d.

1.3.1 State-space form of the CIR model

Since the square-root SDE's solution is a random variable and we know its distribution, we can study the CIR model in the discretized state space form given by Geyer and Pichler, 1999. Consider a sequence of points in time, $t_0, t_0 + \Delta, t_0 + 2\Delta, \ldots$ for the time interval Δ of an arbitrary length, we define the sequence $\{x_s\}_{s=1}^{\infty} \equiv \{r_{t_0+(s-1)\Delta}\}_{s=1}^{\infty}$. Then the density of x_t is:

$$\begin{aligned} p\left(x_1\right) &= \text{Gamma}\left(x_1; c - ce^{-\kappa\Delta}, 1 + q\right) \\ &= \frac{\left(c\left(1 - e^{-\kappa\Delta}\right)\right)^{1+q}}{\Gamma(q+1)} x_1^q e^{-x_1 c\left(1 - e^{-\kappa\Delta}\right)}, \end{aligned} \tag{1.10}$$

and

$$\begin{aligned} p\left(x_t \mid x_{t-1}\right) =& 2c \,\text{noncentral}\chi^2\left(2cx_t; 2(q+1), 2ce^{-\kappa\Delta}x_{t-1}\right) \\ =& c\exp\left[-c\left(x_t + e^{-\kappa\Delta}x_{t-1}\right)\right]\left(\frac{x_t}{e^{-\kappa\Delta}x_{t-1}}\right)^{q/2} \\ & \times I_q\left(2c\sqrt{x_t e^{-\kappa\Delta}x_{t-1}}\right), \end{aligned} \tag{1.11}$$

where q and c are constants defined as

$$q \equiv \frac{2\kappa\theta}{\sigma^2} - 1, \quad c \equiv \frac{2\kappa}{\sigma^2\left(1 - e^{-\kappa\Delta}\right)},$$

and $I_q(\cdot)$ is the first kind modified Bessel function with order q. And the equation of the yields observations y_t with duration τ is

$$y_t(\tau) = -A(\tau) + B(\tau)x_t + \epsilon_t, \;\; \epsilon_t \sim N(0, H), \tag{1.12}$$

where $A(\tau)$ and $B(\tau)$ are known functions of θ, κ, σ, the risk premium parameter λ and the time to maturity τ,

$$\begin{aligned} A(\tau) &= \frac{2\kappa\theta}{\tau\sigma^2} \log\left\{ \frac{\gamma \exp(\tau\eta_1)}{\gamma + \eta_1 \left(e^{\gamma\tau} - 1\right)} \right\}, \\ B(\tau) &= \frac{1}{\tau} \frac{e^{\gamma\tau} - 1}{\gamma + \eta_1 \left(e^{\gamma\tau} - 1\right)}, \\ \eta_1 &= (\kappa + \lambda + \eta_2)/2 \\ \eta_2 &= \sqrt{2\sigma^2 + (\kappa + \lambda)^2}. \end{aligned}$$

We can collect N_y yields into the observation vector $\mathbf{y}_t$

$$\begin{aligned} \mathbf{y}_t &= \begin{bmatrix} y_t(\tau_1) \\ \vdots \\ y_t(\tau_{N_y}) \end{bmatrix} = \begin{bmatrix} B(\tau_1) \\ \vdots \\ B(\tau_{N_y}) \end{bmatrix} x_t - \begin{bmatrix} A(\tau_1) \\ \vdots \\ A(\tau_{N_y}) \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,t} \\ \vdots \\ \varepsilon_{N_y,t} \end{bmatrix} \\ &= \mathbf{B}x_t - \mathbf{A} + \boldsymbol{\varepsilon}_t. \end{aligned}$$

The measurement error vector $\boldsymbol{\varepsilon}_t$ follows the multivariate normal distribution with zero mean and diagonal covariance matrix $h\mathbf{I}_{N_y}$, i.e. the $\varepsilon_{i,t}$ are independent. So the observations are multivariate Gaussian distributed and has mean $\mathbf{B}x_t - \mathbf{A}$ and covariance $h\mathbf{I}_{N_y}$. Then we can generate interest rates and observed yields rates data from these distributions for simulation study.

Chapter 2

Estimation of extreme quantiles

Extreme quantile estimation is an important task of the rare-event study. The rare events are defined as those expected to occur infrequently or more technically have extreme low probabilities of happening, say 10^{-8} or less, according to a probability model (Chopin and Papaspiliopoulos, 2020). It can be found typically in applications fields such as the industrial dynamic systems which require high stabilization and reliability (Kuwahara and Mura, 2008), analysis of queueing networks where the loss of traffic packets is rare (Kovalenko, 1994), fraud detection in finance and insurance (Jin et al., 2005), and safety assessment in transportation system (Kaufman and Giras, 2001).

Although there have been some studies of rare events simulation using the Monte Carlo techniques (see Asmussen et al., 2000 and Giardina et al., 2011), the problem is not adequately studied in the framework of state space model and SMC algorithms introduced in chapter 1, where propagating extreme quantiles of the filtering distribution is the main concern. A naive Monte Carlo simulation method requires billions of random samples when evaluating the probability of a magnitude of 1 in 1 million (10^{-6}), let alone the computations in a repetition structure.

Kaynar and Ridder, 2010 proposed a state-dependent IS algorithm for the simulation of rare events from a Markov chain. It first divides the state space and then applies cross-entropy iterations on those smaller sets. But this method is not suitable for the recursive Bayesian computing problems and time consuming algorithms. Mattos et al., 2006 and Gonzalez and Wu, 1999 introduced the idea of probability distortion in decision making under risk-taking situations and the futures hedging, given the fact that most decision makers do not usually treat probabilities in a linear manner. Inspired by these studies, Neslihanoglu and Date, 2019 presented a framework of applying probability distortion technique on weights before resampling, in order to generate more samples from the extreme tails. The weights are distorted in the way

$$w_t^{(i)'} = \frac{\exp\left(-s \times w_t^{(i)}\right) - 1}{\exp(-s) - 1},$$

where s is a coefficient determining the extent of distortion. However, since the particles are already generated before the weighting step, there is little or no particles obtained from the extreme tails at all, when the particle size is not large enough. Besides, a normal approximation of extract transition density, which overly simplified the CIR model, is used as both transition density and proposal (means that nature of the algorithm is bootstrap filer). There exists some nuances between the approximated and exact densities on extreme tails, so the computed theoretical quantiles used as standard are biased themselves. The figure 2.1 displays the differences between two densities on tails.

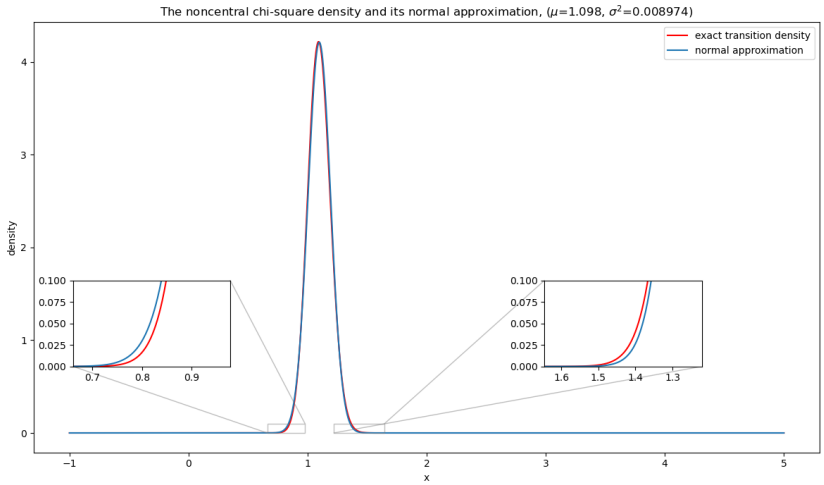

Figure 2.1: Density of noncentral chi-square distribution and the normal approximation.

In the following discussions, we first provide an approximation of the optimal proposal for the CIR model with exact transition density, then outline the strategy of mixing it with its own tails to form a mixture proposal, in order to efficiently propagate samples from the extreme tails and estimate quantiles accurately.

2.1 Optimal proposal for the CIR model

We already know that $q(x_t|x_{t-1}, \mathbf{y}_t) = p(x_t|x_{t-1}, \mathbf{y}_t) \propto p(x_t|x_{t-1})f(\mathbf{y}_t|x_t)$ is the most favorable proposal in the sense of minimizing the variance of weights w_t, which can be regarded as a balance between the information on x_t given by x_{t-1} and those brought by $\mathbf{y}_t$. Since the transition density (1.11) is intractable, we can make a normal approximation to it by preserving the first two moments. For a non-central χ^2 distributed random variable with non-centrality parameter λ and degrees of freedom k, its mean

and variance are $\mu = k + \lambda$, and $\sigma^2 = 2(k + 2\lambda)$ respectively. So the normal approximation will have parameters,

$$\mu(x_{t-1}) = \frac{q+1}{c} + e^{-\kappa\Delta} x_{t-1} \text{ and } \sigma^2(x_{t-1}) = \frac{q+1}{c^2} + \frac{2e^{-\kappa\Delta} x_{t-1}}{c}.$$

Treating x_t as the random term, the observation density (1.12), which is multivariate normal, could be written as

$$\begin{aligned} f(\mathbf{y}_t \mid x_t) &= \prod_{i=1}^{N_y} \frac{1}{\sqrt{2\pi h}} \exp\left[-\frac{(y_t(\tau_i) + A(\tau_i) - B(\tau_i) x_t)^2}{2h} \right] \\ &\propto \exp\left[-\frac{1}{2h} \sum_{i=1}^{N_y} B(\tau_i)^2 x_t^2 + \frac{1}{h} \sum_{i=1}^{N_y} B(\tau_i)(y_t(\tau_i) + A(\tau_i)) x_t \right] \\ &\equiv N\left(\frac{\sum_{i=1}^{N_y} B(\tau_i)(y_t(\tau_i) + A(\tau_i))}{\sum_{i=1}^{N_y} B(\tau_i)^2}, \frac{h}{\sum_{i=1}^{N_y} B(\tau_i)^2} \right) \equiv N\left(\mu(\mathbf{y}_t), \sigma^2(\mathbf{y}_t) \right). \end{aligned}$$

Then product the two densities and normalize, we get the approximate optimal proposal density

$$q_{opt}(x_t | x_{t-1}, \mathbf{y}_t) \equiv N\left(\frac{\mu(x_{t-1})/\sigma^2(x_{t-1}) + \mu(\mathbf{y}_t)/\sigma^2(\mathbf{y}_t)}{1/\sigma^2(x_{t-1}) + 1/\sigma^2(\mathbf{y}_t)}, \frac{\sigma^2(x_{t-1})\sigma^2(\mathbf{y}_t)}{\sigma^2(x_{t-1}) + \sigma^2(\mathbf{y}_t)} \right), \quad (2.1)$$

which is truncated at zero due to the nonnegative property of states. In order to more emphasize the tail part of the posterior distribution, we can replace the normal proposal by a t-distribution proposal with the exactly same first two moments.

2.2 Mixing proposal and its tails

Since we are more interested in the extreme tails of the filtering posterior distribution instead of the region of high density. We can truncate the original optimal proposals at certain quantiles and take the tails to form new proposals, then generate a proportion of particles from them. Mathematically, denote the β-th and $(1-\beta)$-th quantiles of x_t with density $q(x_t|x_{t-1}, \mathbf{y}_t)$ by $\theta_{t,(\beta)}(x_{t-1})$ and $\theta_{t,(1-\beta)}(x_{t-1})$. We have the component proposals: $q(x_t|x_{t-1}, \mathbf{y}_t)$, $(x_t - \theta_{t,(\beta)}(x_{t-1}))^+ q(x_t|x_{t-1}, \mathbf{y}_t)$, and $(x_t - \theta_{t,(1-\beta)}(x_{t-1}))^- q(x_t|x_{t-1}, \mathbf{y}_t)$. As an illustration of this idea, Figure 2.2 displays the optimal proposal density colored in gray, and its two tails in red, where the density curve is not proper scaling for simplicity. Then the new strategy is clear, in each iterative step of the mixture SMC algorithm, α_1 proportions of particles are generated from the optimal proposal, and α_2 and α_3 portions of particles are generated from the two tail proposals $(x_t - \theta_{t,(\beta)}(x_{t-1}))^+ q(x_t|x_{t-1}, \mathbf{y}_t)$, and $(x_t - \theta_{t,(1-\beta)}(x_{t-1}))^- q(x_t|x_{t-1}, \mathbf{y}_t)$ respectively. The $\boldsymbol{\alpha}$ and β are user-chosen algorithm parameters. Li et al., 2016 gave some guidelines on the selection of the mixture proportions $\boldsymbol{\alpha}$ for the optimal filtering task.

Figure 2.3 compares the posterior density, optimal and the mixture proposals, It can

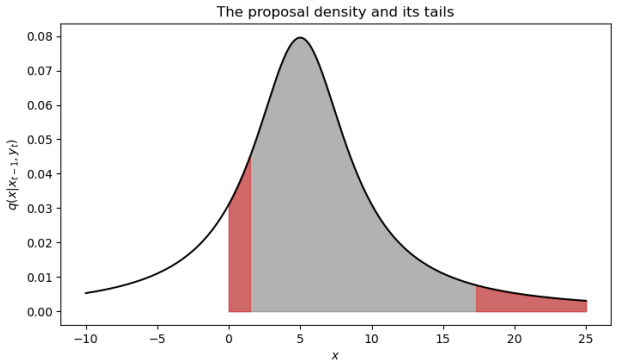

Figure 2.2: Plot of the optimal proposal and its two tail

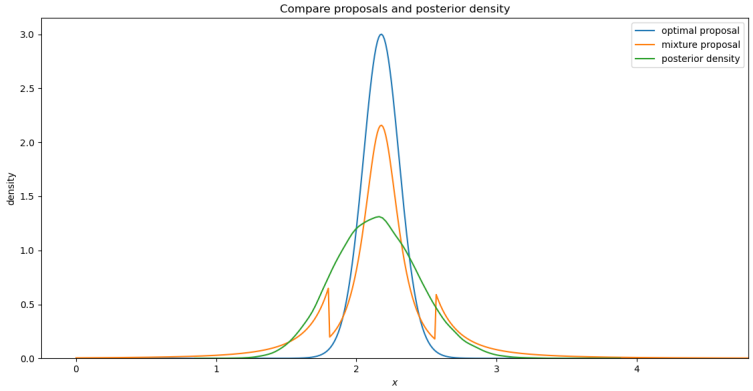

Figure 2.3: The density curves of the proposals and theoretical posterior distribution

be seen that the optimal proposal envelopes the target (posterior) density in the region around the mean and mode but fails on tails, and the mixture proposal could envelopes the target density at tail parts. So it is expected that the mixture SMC algorithm adopted our new strategy would perform better than the bootstrap filter and basic guided particle filters.

Besides the CIR model, this strategy has potential applications in many other fields where the problem can be translated to a state space model and modeling the rare event is of interests. In next chapter, we will conduct some simulations to support above assumption and demonstrate that algorithm using the new strategy can efficiently estimate the extreme quantiles of the short rates driven by the CIR model.

Chapter 3

Simulation study and results

To numerically verify our expectations on the new strategy of mixture SMC, we shall conduct some simulations on generated data. The four methods are compared: bootstrap filter, modified SMC, guided particle filter, and mixture SMC. Among all of them, multinomial resampling is used at every step (threshold of ESS is equal to the sample size), each step has $N = 100$ particles. The simulation is independently repeated $R = 500$ times.

Considering the parameters estimated by Geyer and Pichler, 1999 ($\kappa = 0.169$, $\theta = 6.56$, $\sigma = 0.321$ and $\lambda = -0.201$), we can simulate samples of $T = 100$ monthly observations from the CIR model with these parameters and following equations (1.10) and (1.11). Yields of five zero-coupon bonds with maturities 0.25, 1, 3, 5 and 10 years are generated according to (1.12) as observations, and their variance H are selected so that the signal to noise ratio (SNR) takes the values 10, 5, 1 and 0.1. The formal definition of SNR is the ratio of the variance of states and observations. But since the CIR model is inhomogeneous due to the square root of state multiplied on the Brownian motion term ($\mathrm{d}r_t = \kappa(\theta - r_t)\,\mathrm{d}t + \sigma\sqrt{r_t}\mathrm{d}Z_t$), there is not a uniform variance for the states. So we replace the r_t with its long term mean θ to represent an average level of r_t. Therefore, the SNR is defined as $\mathrm{SNR} = \sigma^2\theta/H$ in our simulations. Figure 3.1 shows the trajectories of observations generated for different SNR, the maturity is labeled on the upper left of the subplots.

In terms of the extreme quantile estimation, performance of these methods are compared with the average mean squared error (MSE) over $T = 100$ steps, which is defined as,

$$\overline{\mathrm{MSE}} = \frac{1}{T}\sum_{t=1}^{T}\mathrm{MSE}_t, \text{ where } \mathrm{MSE}_t = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{x}^{(r)}_{t,(\alpha)} - x_{t,(\alpha)}\right)^2, \tag{3.1}$$

and $\hat{x}^{(r)}_{t,(\alpha)}$ is the estimate of specific quantile obtained from model (i.e. empirical quantile of the propagated particles), at t-th time step and the r-th repeats, $x_{t,(\alpha)}$ is the theoretical quantile of target distribution. The model with the lowest MSE indicates best performance on extreme quantile estimation. Following the definition of rare events given in chapter 2, we estimate the following quantiles on both lower and upper tails: 10^{-8}, 10^{-5}, 10^{-3}. Besides, we also compare the algorithms from filtering aspect using

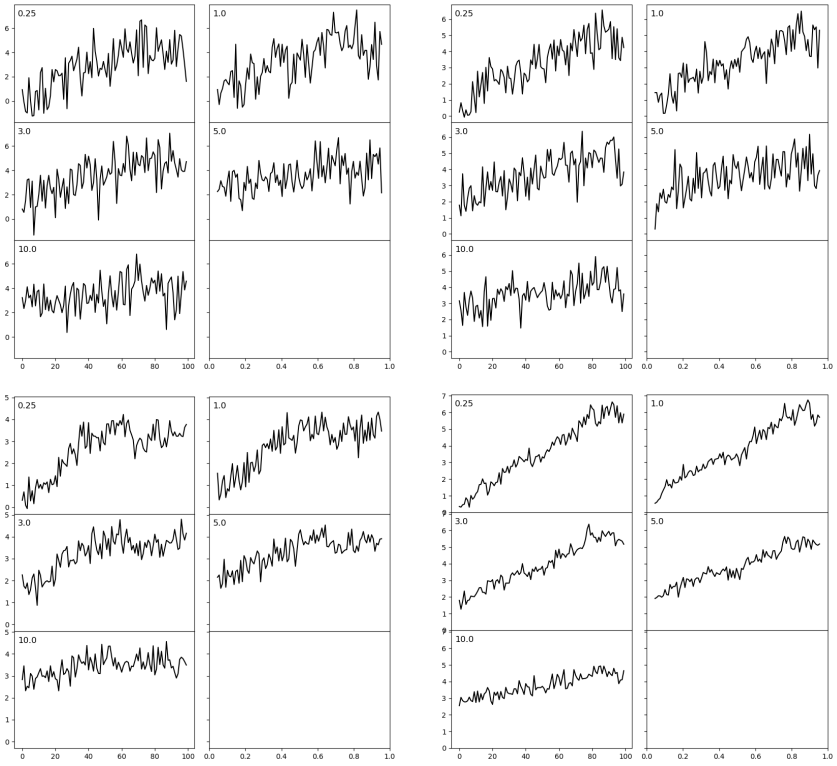

Figure 3.1: Observation trajectories for different SNR (0.5, 1, 5, 10)

the average MSE of the posterior mean's estimates and the the standard errors,

$$\overline{\text{MSE}} = \frac{1}{TR}\sum_{t=1}^{T}\sum_{r=1}^{R}\left(\hat{\mu}_t^{(r)} - \mu_t\right)^2, \text{ and } \text{std}_t = \frac{1}{R}\sqrt{\sum_{r=1}^{R}\hat{\mu}_t^{(r)2} - \left(\sum_{r=1}^{R}\hat{\mu}_t^{(r)}\right)^2},$$

where $\hat{\mu}_t^{(r)}$ is the estimate at the t-th step and r-th replication, μ_t is the theoretical posterior mean of x_t.

In our simulations, the theoretical quantiles and mean of the filtering distribution are approximated by a bootstrap filter with large particle size, for example 100,000. To ensure that such approximation is reasonable, we execute the algorithm for various particle sizes (M) several times and see whether the estimation gets stable as M increases. Figure 3.2 shows the box plots of the estimated standard errors of estimates of various quantities interested, which are computed over 100 replications in each time step. As the particle size increasing, the standard errors of the approximations keep decreasing, and they converge to the true theoretical values. Therefore it's sound to regard these approximations as standards for evaluating and comparing the algorithms.

For mixture SMC, the extra two component proposals are the tail parts of the approximated optimal proposal cut off at certain quantiles. The location and scaling parameter of the optimal proposal are computed according to (2.1). For a Normal proposal, lower

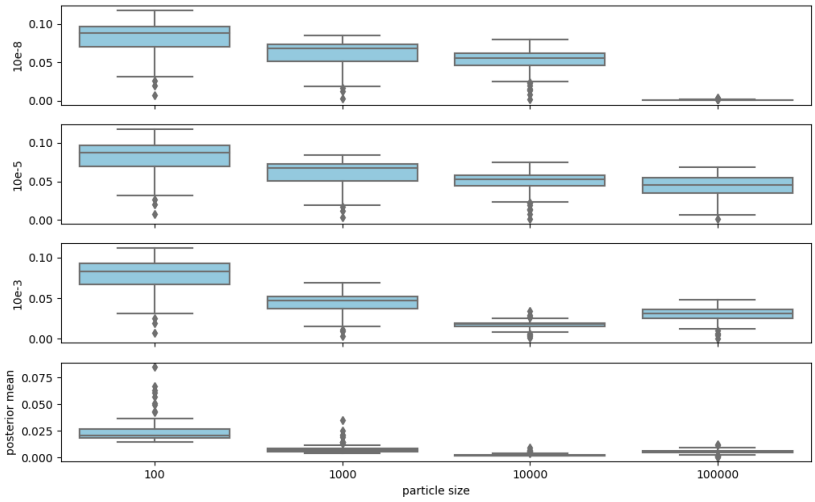

Figure 3.2: Each box plot represents the distribution of standard errors of 100 estimates of certain quantity over T=100 time steps of the same simulated sample. The amount of particles for each group of exercises are, 100, 1000, 10000 and 100000 respectively.

and upper 0.01 quantiles of the importance density are chosen to be the cut off points, while for a t proposal with degrees of freedom 5 we use the 0.05 and 0.95 quantiles. The three proposals are mixed with mixture proportions 0.8, 0.1, 0.1 in both normal and t cases.

Table 3.1 displays the average MSEs defined above for various quantiles estimates as well as the computational times relative to a bootstrap filter with 100,000 particle size, which is about 89.32s. The included algorithms are: bootstrap filter (BF), modified SMC with distorted weights (SMC_mod), guided particle filter with normal and t optimal proposal (SMC and SMCt), mixture SMC algorithms described above (normal+q811.95 and t5+q811.95).

In table 3.1, it is seen that,

1. When the SNR is small, the mixture SMC algorithm, especially the one with t-distribution optimal proposal, outperforms the other methods significantly for estimating extreme quantiles, although it is computationally most expensive.

2. If the SNR is large, simple algorithms like the bootstrap filter are good enough. But the usage of tail proposal can still enhance the uni-proposal guided particle filter.

Lower SNR indicates lower signal strength, and relatively weaker system dynamics. The introduce of information from observations is worthy, which gives guided particle filter and mixture SMC more advantages to approximate the filtering distribution. Since part of particles are directly sampled from tails of proposal, the mixture SMC could further improve the estimates of extreme quantiles. So our new strategy of the

Table 3.1: Average MSEs of the estimates of various quantiles of CIR

SNR=0.5	1.0e-08	1.0e-05	1.0e-03	1-1.0e-03	1-1.0e-05	1-1.0e-08	Time
BF	0.3061	0.2578	0.0645	0.1050	0.5064	0.6471	0.0057
SMC_mod	0.3051	0.2568	0.0637	0.1010	0.4981	0.6379	0.0094
SMC	0.3296	0.2790	0.0738	0.1338	0.5792	0.7275	0.0957
SMCt	0.2415	0.2009	0.0555	0.0994	0.4454	0.5722	0.1057
normal+q811.99	0.2953	0.2475	0.0593	0.1028	0.5050	0.6445	0.4327
t5+q811.95	0.1567	0.1299	0.0587	0.0851	0.3049	0.4020	0.4535
SNR=1	1.0e-08	1.0e-05	1.0e-03	1-1.0e-03	1-1.0e-05	1-1.0e-08	Time
BF	0.2725	0.2134	0.0445	0.0711	0.3530	0.4556	0.0057
SMC_mod	0.2779	0.2181	0.0462	0.0698	0.3493	0.4508	0.0094
SMC	0.3260	0.2628	0.0666	0.1131	0.4501	0.5667	0.0957
SMCt	0.2276	0.1799	0.0525	0.0813	0.3215	0.4148	0.1057
normal+q811.99	0.2810	0.2222	0.0487	0.0875	0.3931	0.5027	0.4327
t5+q811.95	0.1375	0.1083	0.0595	0.0802	0.2170	0.2853	0.4535
SNR=5	1.0e-08	1.0e-05	1.0e-03	1-1.0e-03	1-1.0e-05	1-1.0e-08	Time
BF	0.1358	0.1056	0.0219	0.0320	0.1746	0.2229	0.0057
SMC_mod	0.1355	0.1054	0.0220	0.0316	0.1734	0.2215	0.0094
SMC	0.2154	0.1765	0.0593	0.1148	0.3406	0.4063	0.0957
SMCt	0.1359	0.1091	0.0445	0.0682	0.2199	0.2699	0.1057
normal+q811.99	0.1914	0.1547	0.0482	0.0942	0.3024	0.3644	0.4327
t5+q811.95	0.0829	0.0672	0.0498	0.0575	0.1414	0.1765	0.4535
SNR=10	1.0e-08	1.0e-05	1.0e-03	1-1.0e-03	1-1.0e-05	1-1.0e-08	Time
BF	0.1471	0.1129	0.0220	0.0293	0.1603	0.2063	0.0057
SMC_mod	0.1477	0.1135	0.0224	0.0285	0.1581	0.2039	0.0094
SMC	0.3643	0.3103	0.1359	0.1553	0.3935	0.4634	0.0957
SMCt	0.2330	0.1939	0.0819	0.0928	0.2549	0.3082	0.1057
normal+q811.99	0.3358	0.2842	0.1195	0.1377	0.3634	0.4305	0.4327
t5+q811.95	0.1518	0.1239	0.0597	0.0683	0.1691	0.2082	0.4535

mixture SMC gives improvement in situations where SNR is small and the simple algorithms perform poorly. But when the SNR is large, the observations are relatively less informative, elementary methods are good enough, and there is no need to employ guided particle filters or mixture SMC.

Through more experiments, we found that a bootstrap filter with particle size $N = 50000$ has the computational time close to the mixture SMC algorithms, but its MSEs are also much larger when the SNR is small. So taking the computational time into account, the new strategy still maintains its advantages.

Table 3.2 displays the MSEs for the estimation of filtering mean. For same data set, the MSE values of different algorithms are very close, but considering the computational time, the new method is inappropriate for ordinary filtering task of the central moment.

In conclusion, the mixture SMC strategy presented here provides a new way for estimating the extreme quantiles of the filtering distribution of a state space model, with a desirable accuracy at acceptable computational cost. We have shown through numerical simulations on the CIR model that the new strategy could estimate extreme

Table 3.2: Average MSE of posterior mean's estimates

	SNR			
Algorithm	0.5	1	5	10
bootstrap	0.004404	0.258779	0.727185	0.707971
SMC_mod	0.004063	0.25962	0.725874	0.708934
SMC	0.004106	0.259771	0.728944	0.706787
SMCt	0.004052	0.260954	0.727835	0.706399
normal+q811.99	0.003921	0.259149	0.728918	0.706332
t5+q811.95	0.004426	0.25995	0.729324	0.70585

quantiles with high accuracy for certain situations where the SNR is not too large. It offers a great flexibility and customization ability as well, on the select of component proposals and mixture proportions.The benefits of this strategy will span a large number of applications in various fields. Future research might consider the applications of this strategy within more complex state space models, especially those with multivariate state variable, like the multivariate autoregressive models, multi-factor CIR model, and the source localization models.

Acknowledgements

First of all, I have to express my most sincere gratitude to my supervisor Dr. LI Wentao for his continuous guidance, patience, and motivation throughout the development of the project. He supported and encouraged my selection of this topic and introduced the SMC algorithms to me when he knew I am interested in it. Since the beginning of this project, we have had meetings every week which keeps me on the right track of the study.

I am also grateful to my parents, ZHANG Yugui and LI Huiqing, for their spiritual and dietary supports during the period of studying at home in the last semester. Their support and love means a lot to me. Finally, my sincere thanks also go to my friends and classmates, LAI Daoyuan, LIANG Jun, LUO Qingyun and MENG Chuhan for their useful and constructive advice and kind help. It's they who make my study in Hong Kong a pleasant and memorable one.

Yun ZHANG
The University of Hong Kong
April 26, 2021

Bibliography

Aeberhard, W. H., Mills Flemming, J., & Nielsen, A. (2018). Review of state-space models for fisheries science. *Annual Review of Statistics and Its Application*, *5*, 215–235.

Aoki, M. (2013). *State space modeling of time series*. Springer Science & Business Media.

Asmussen, S., Binswanger, K., Hojgaard, B., et al. (2000). Rare events simulation for heavy-tailed distributions. *Bernoulli*, *6*(2), 303–322.

Buckland, S., Newman, K., Thomas, L., & Koesters, N. (2004). State-space models for the dynamics of wild animal populations. *Ecological modelling*, *171*(1-2), 157–175.

Chopin, N., & Papaspiliopoulos, O. (2020). *An introduction to sequential monte carlo*. Springer.

Cox, J., Ingersoll, J., & Ross, S. (1985). A theory of the term structure of interest rates. *Econometrica*, *53*(2), 385–407.

Douc, R., & Cappe, O. (2005). Comparison of resampling schemes for particle filtering. *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, 64–69.

Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, *12*(656-704), 3.

Geyer, A. L., & Pichler, S. (1999). A state-space approach to estimate and test multifactor cox-ingersoll-ross models of the term structure. *Journal of Financial Research*, *22*(1), 107–130.

Giardina, C., Kurchan, J., Lecomte, V., & Tailleur, J. (2011). Simulating rare events in dynamical processes. *Journal of statistical physics*, *145*(4), 787–811.

Gonzalez, R., & Wu, G. (1999). On the shape of the probability weighting function. *Cognitive psychology*, *38*(1), 129–166.

Gordon, N. J., Salmond, D. J., & Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE proceedings F (radar and signal processing)*, *140*(2), 107–113.

Jasra, A., & Del Moral, P. (2011). Sequential monte carlo methods for option pricing. *Stochastic analysis and applications*, *29*(2), 292–316.

Jin, Y., Rejesus, R. M., & Little, B. B. (2005). Binary choice models for rare events data: A crop insurance fraud application. *Applied Economics*, *37*(7), 841–848.

Kaufman, L. M., & Giras, T. C. (2001). Simulation of rare events in transportation systems. *Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304)*, *2*, 1380–1385.

Kaynar, B., & Ridder, A. (2010). The cross-entropy method with patching for rare-event simulation of large markov chains. *European journal of operational research*, *207*(3), 1380–1397.

Kovalenko, I. N. (1994). Rare events in queueing systems - a survey. *Queueing systems*, *16*(1), 1–49.

Kuwahara, H., & Mura, I. (2008). An efficient and exact stochastic simulation method to analyze rare events in biochemical systems. *The Journal of chemical physics*, *129*(16), 10B619.

Li, W., Chen, R., & Tan, Z. (2016). Efficient sequential monte carlo with multiple proposals and control variates. *Journal of the American Statistical Association*, *111*(513), 298–313.

Mattos, F., Garcia, P., & Pennings, J. M. (2006). *Probability distortion and loss aversion in futures hedging* (tech. rep.).

Miranda, H., Corts, P., Yuz, J. I., & Rodrguez, J. (2009). Predictive torque control of induction machines based on state-space models. *IEEE Transactions on Industrial Electronics*, *56*(6), 1916–1924.

Neslihanoglu, S., & Date, P. (2019). A modified sequential monte carlo procedure for the efficient recursive estimation of extreme quantiles. *Journal of Forecasting*, *38*(5), 390–399.

Orton, M., & Fitzgerald, W. (2002). A bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, *50*(2), 216–223.

Stone, L. D., Streit, R. L., Corwin, T. L., & Bell, K. L. (2013). *Bayesian multiple target tracking*. Artech House.

Vermaak, J., Godsill, S. J., & Perez, P. (2005). Monte carlo filtering for multi target tracking and data association. *IEEE Transactions on Aerospace and Electronic systems*, *41*(1), 309–332.

Wang, X., Chen, R., & Liu, J. S. (2002). Monte carlo bayesian signal processing for wireless communications. *Journal of VLSI signal processing systems for signal, image and video technology*, *30*(1), 89–105.