

A NEW APPROACH TO ADAPTIVE PARTICLE FILTERS FOR
JOINT STATE AND PARAMETER ESTIMATION IN HIDDEN
MARKOV MODELS

A DISSERTATION
SUBMITTED TO THE INSTITUTE FOR COMPUTATIONAL
AND MATHEMATICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Pengfei Gao

August 2019

© 2019 by Pengfei Gao. All Rights Reserved.
Re-distributed by Stanford University under license with the author.

This dissertation is online at: <http://purl.stanford.edu/yd865df3794>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Tse Lai, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Han Hong

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ying Lu

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumpert, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

The landmark paper by Gordon, Salmond and Smith [22] launched the development of sequential Monte Carlo (SMC), also called particle filters, for the estimation of latent states in hidden Markov models (HMM). Liu [39] contains a collection of the techniques that have been developed since then, with examples of applications in computational biology and engineering, and Chan and Lai [9] provide a general theory of particle filters, assuming the model parameters to be pre-specified. This assumption is too restrictive in practice, since the model parameters are usually unknown and also need to be estimated sequentially from the observed data. The obvious approach that treats the parameters as latent states and thereby incorporates them in a larger state vector suffers from severe degeneracy difficulties of the particle filter because the subvector corresponding to the parameters does not undergo Markovian dynamics. Beginning with Liu and West [38] and Storvik [44], there have been many proposals to address this difficulty; see [2]. In particular, Andrieu, Doucet and Holenstein [2] developed the particle MCMC (PMCMC) approach and Chopin, Jacob, and Papaspiliopoulos[11] subsequently introduced the SMC² method. These two approaches have achieved the state-of-the-art performance.

In this thesis, we introduce a new approach to adaptive particle filters for joint parameter and state estimation in HMMs and develop a complete asymptotic theory

showing its computational and statistical advantages over previous methods. This approach also provides consistent estimates of (a) the standard errors for the Monte Carlo estimate and (b) mean squared errors of the adaptive particle filter. We accomplish this by combining the theory of particle filters for state estimation in Chan and Lai [9] when the parameters are known with that of a novel MCMC scheme using sequential state substitutions for parameter estimation (MCMC-SS) in Lai, Zhu and Chan[34]. Chapter 2 describes our new adaptive particle filter, its computational advantages and how it seamlessly combines the aforementioned two components (a) and (b). Applications to nonlinear state space models in automatic navigation and to HMMs in quantitative finance are given in Chapter 3. Concluding remarks are given in Chapter 4, in which we also provide further discussion of our approach and additional related references in the literature.

Acknowledgments

First of all, I would like to thank my advisor Professor Tze Leung Lai for his continuous support during my PhD life. I admire Professor Lai’s deep understanding of statistics and his broad knowledge on extended areas including health care, financial mathematics, econometrics and so many other fields. I first knew Professor Lai when I was still a junior. In the summer of 2013, I attended the SIAM conference in San Diego and then stopped by Stanford University. I was lucky to “catch” Professor Lai during my visit and met him in one of the conference room of statistics department, where we later used that room millions of times for discussion after I became a PhD student here. I expressed my interest in doing research in statistics and Professor Lai encouraged me to continue working towards my goals. Fortunately, two years later, I became a PhD candidate at Institute Computational and Mathematical Engineering (ICME) of Stanford University and was able to work with Professor Lai. I am very grateful to him for teaching me many things, from statistics to real life, and for reading my thesis.

Secondly, I give my deep thanks to my committee members, Professor Han Hong, Professor Ying Lu, Professor Michael Saunders, and Professor Jose H. Blanchet. Their encouragement and advice are precious sources for me to improve my research and thesis. I would also like to thank ICME Director, Professor Gianluca Iaccarino, and

former Director Professor Margot Gerritsen, for their support of me and the students in ICME. Stanford University and ICME have provided me with a great environment to study and work, and I have had a great life here for the past five years.

Finally, I would like to express my great gratitude to all my friends and families, for their love, understanding, and support during these years. They are the driving forces for me to go forward. I will carry their expectations forward and achieve more in the future.

Contents

Abstract	iv
Acknowledgments	vi
1 Introduction	1
1.1 State space models	2
1.2 Sequential Monte Carlo with known parameters	4
1.3 Monte Carlo Markov Chain with known posterior likelihood $\pi(\theta Y_{1:t})$	6
1.3.1 Metropolis-Hastings MCMC	7
1.3.2 MCMC with sequential substitution	8
1.4 Joint estimation of hidden states and unknown parameters	11
1.4.1 Augmenting the state with the parameter	11
1.4.2 Gibbs sampling	13
1.4.3 Particle marginal Metropolis-Hastings and SMC ²	14
1.4.4 Particle Gibbs sampling	17
1.4.5 Markov chain approximation method	19
2 A new adaptive particle filter	21
2.1 Oracle case in which $\pi(\theta Y_{1:T})$ is known and easy to sample from	22

2.2	Combining MCMC-SS with SMC	22
2.3	Group sequential oracle procedure	24
2.4	Recursive MCMC-SS/SMC filter	25
2.5	Computational methods and implementation	29
3	Simulation studies and applications	31
3.1	A linear Gaussian model with unknown model parameters	31
3.2	A nonlinear state space model with unknown parameters	37
3.3	Adaptive particle filters for latent efficient price process with marked point process observations and exogenous parameter jumps	40
3.4	Adaptive particle filters in stochastic volatility models with contemporaneous jumps	46
4	Conclusion	51
A	Numerical experiments supplements	54
A.1	Nonlinear state space model	54
A.2	Latent efficient price process	63
A.2.1	Proposal designs	63
A.2.2	Applications in real world dataset	66
A.3	Stochastic volatility models	69
A.3.1	State space modeling	69
A.3.2	Proposal designs	72
A.3.3	Comparison studies with Eraker et al. [20]	76
A.3.4	More filtering results for S&P 500	80

Chapter 1

Introduction

In this chapter, we give an overview of the research question and related literature. Firstly, Section 1.1 reviews hidden Markov models (HMMs), also known as state space models, and introduces our notations. Then Section 1.2 presents the case where model parameters are known and sequential Monte Carlo are developed to filter latent states. Section 1.3 discusses situations where model parameters are unknown but posterior likelihood can be easily calculated. In this case, Monte Carlo Markov Chain (MCMC), in particular, MCMC with sequential substitution (MCMC-SS) is effective to learn posterior distributions of model parameters, which will be an important building block for our adaptive particle filters in Chapter 2. Finally, Section 1.4 covers methods that have been used for joint estimation of latent states and model parameters. This includes standard MCMC methods, particle-based statistical methods, and numerical methods.

1.1 State space models

In this section, we briefly review the state space models, also known as hidden Markov models and explain the computational difficulties when estimating values of interest for this model.

Suppose two sequences of random variables $X_0, X_1, X_2, \dots, X_t$ and Y_1, Y_2, \dots, Y_t . They satisfy the following transition processes:

$$X_t \sim p_t(\cdot | X_{t-1}, \theta), \quad Y_t \sim g_t(\cdot | X_t, \theta), \quad \text{for } t \geq 1 \quad (1.1)$$

with an initial distribution,

$$X_0 \sim p_0(\cdot | \theta), \quad (1.2)$$

in which p_t and g_t are known density functions with respect to measures ν_X and ν_Y . X_t are called latent or hidden states undergoing a Markov process. Y_t are called observed states and are conditionally independent given latent states X_t . θ are parameters depicting the two parametric distributions p_t and g_t . Without loss of generality, we further impose a prior on θ under the assumption that θ is unknown:

$$\theta \sim \pi(\theta), \quad (1.3)$$

in which π is a known prior density with respect to measure ν_θ . We also define $X_{0:T} = \{X_t\}_{t=0:T}$ and $Y_{1:T} = \{Y_t\}_{t=1:T}$, called latent states path and observed states path respectively, as a store of all past states. Given observed states path $Y_{1:T}$, we can calculate the joint posterior distribution of $X_{0:T}, \theta | Y_{1:T}$ and estimate our variable

of interest, which is usually a posterior expectation of a pre-specified function ψ :

$$\mathbb{E}[\psi(\theta, X_{0:T})|Y_{1:T}] = \int \psi(\theta, X_{0:T})\mathbb{P}(\theta, X_{0:T}|Y_{1:T})d\theta dX_{0:T} \quad (1.4)$$

$$= \int_{\theta} [\int_{X_{0:T}} \psi(\theta, X_{0:T})\mathbb{P}(X_{0:T}|Y_{1:T}, \theta)dX_{0:T}]\mathbb{P}(\theta|Y_{1:T})d\theta \quad (1.5)$$

$$= \int_{\theta} \mathbb{E}[\psi(\theta, X_{0:T})|Y_{1:T}, \theta]\mathbb{P}(\theta|Y_{1:T})d\theta \quad (1.6)$$

The two conditional probabilities $\mathbb{P}(X_{0:T}|Y_{1:T}, \theta)$ and $\mathbb{P}(\theta|Y_{1:T})$ (also denoted as density $\pi(\theta|Y_{1:t})$ with respect to measure ν_{θ}) in equation (1.5) can be calculated as follows:

$$\mathbb{P}(X_{0:T}|Y_{1:T}, \theta) \propto p_0(X_0|\theta) \prod_{t=1}^T p_t(X_t|X_{t-1}, \theta)g_t(Y_t|X_t, \theta) \quad (1.7)$$

$$\pi(\theta|Y_{1:T}) := \mathbb{P}(\theta|Y_{1:T}) \propto \pi(\theta) \int_{X_{0:T}} p_0(X_0|\theta) \prod_{t=1}^T p_t(X_t|X_{t-1}, \theta)g_t(Y_t|X_t, \theta)dX_{0:T} \quad (1.8)$$

As can be seen from equation (1.7), although we can calculate $\mathbb{P}(X_{0:T}|Y_{1:T}, \theta)$ up to some multiplicative constant, sampling from this density can be difficult. Moreover, the integral in equation (1.8) is in high dimensional space, which make it challenging to calculate accurately and efficiently, not mention to sample from it. Therefore, the direct estimation of $\mathbb{E}[\psi(\theta, X_{0:T})|Y_{1:T}]$ is intractable.

1.2 Sequential Monte Carlo with known parameters

The problem defined in the last section contains two unknown components: latent states path $X_{0:t}$ and model parameters θ . In this section, we first look into a simpler problem, where model parameters are known. Dropping out θ in equation (1.1) and (1.2), let $X_t \sim p_t(\cdot|X_{t-1}), Y_t \sim g_t(\cdot|X_t)$. The density function \tilde{p}_T of $X_{0:T} = (X_0, \dots, X_T)$ conditional on $Y_{1:T} = (Y_1, \dots, Y_T)$ is

$$\tilde{p}_T(x_{0:T}|Y_{1:T}) \propto \prod_{t=1}^T [p_t(x_t|x_{t-1})g_t(Y_t|x_t)]. \quad (1.9)$$

This conditional distribution is still difficult to sample from and the normalizing constant is also difficult to compute for high-dimensional or complicated state spaces, and particle filters use sequential Monte Carlo that involves importance sampling and resampling to circumvent this difficulty. The particle filter computes $\mathbb{E}[\psi(X_{0:T})|Y_1, \dots, Y_T]$ by the recursive Monte Carlo scheme summarized in Algorithm 1. Let $X_{0:t-1}^m$ denote the sample path of the m th particle (trajectory), $1 \leq m \leq M$. The algorithm uses importance sampling from a proposal density q_t to circumvent this difficulty and updates not only the particles $X_{0:t-1}^m$ but also the associated weights w_{t-1}^m and ancestor A_{t-1}^m of $X_{0:t}^m$. It is initialized with $A_0^m = m$ and $w_0^m = 1$. The SMC estimate of $\psi_T := \mathbb{E}[\psi(X_{0:T})|Y_{1:T}]$ is

$$\hat{\psi}_T = \sum_{m=1}^M w_T^m \psi(\tilde{X}_{0:T}^m) / \sum_{m=1}^M w_T^m. \quad (1.10)$$

Algorithm 1 SMC with M particles

- *Initialization:* $A_0^i = i, w_0^i = 1$ for $i = 1, \dots, M$
 - *Importance sampling at stage $t \in \{1, \dots, T\}$:* Generate conditionally independent \tilde{X}_t^i from $q_t(\cdot | X_{0:t-1}^i)$ and set $\tilde{X}_{0:t}^i = (X_{0:t-1}^i, \tilde{X}_t^i), w_t^i = p_t(\tilde{X}_t^i | X_{0:t-1}^i)g_t(y_t | \tilde{X}_t^i)/q_t(\tilde{X}_t^i | X_{0:t-1}^i), i = 1, \dots, M$
 - *Bootstrap resampling at stage $t \in \{1, \dots, T-1\}$:* Generate i.i.d random variables B_t^1, \dots, B_t^M such that $P(B_t^i = j) = w_t^j / \sum_{i=1}^M w_t^i, j = 1, \dots, M$. Let $(X_{0:t}^m, A_t^m) = (\tilde{X}_{0:t}^{B_t^m}, A_{t-1}^{B_t^m}), m = 1, \dots, M$.
-

By using martingale theory, Chan and Lai [9] provide a comprehensive theory of the SMC estimate $\hat{\psi}_T$, which includes asymptotic normality and consistent standard error estimation as the following:

Theorem 1.1. *Under certain integrability conditions,*

$$\sqrt{M}(\hat{\psi}_T - \psi_T) \Rightarrow N(0, \sigma^2).$$

Moreover, letting $\bar{w}_t = M^{-1} \sum_{i=1}^M w_t^i$, σ^2 can be consistently estimated by

$$\hat{\sigma}^2 = \frac{1}{M} \sum_{m=1}^M \left(\sum_{i: A_{T-1}^i = m} \frac{w_T^i}{\bar{w}_T} [\psi(\tilde{X}_{0:T}^i) - \hat{\psi}_T]^2 \right).$$

Chan and Lai [9, Lemmas 1 and 4] use the following representation of $\hat{\psi}_T - \psi_T$ to derive Theorem 1.1. Let $w_t(x_{0:t}) = p_t(x_t | x_{t-1})g_t(Y_t | x_t)/q_t(x_t | x_{0:t-1})$, in which the Y_t can be treated as constants since the particle filter is the conditional distribution of $X_{0:t}$ given the observations Y_1, \dots, Y_t . Let $H_t^m = (\bar{w}_1 \cdots \bar{w}_t) / \prod_{j=1}^t w_j^m$, $\eta_t = \mathbb{E}_q[\prod_{i=1}^t w_i(X_{0:t})]$, where \mathbb{E}_q denotes expectation under which $X_t | X_{0:t-1}$ has

the conditional density function $q_t(\cdot|X_{0:t-1})$ for $1 \leq t \leq T$. Letting $\Psi_0 = \psi_T$ and $\Psi_t(X_{0:t}) = \mathbb{E}_q \left\{ \psi(X_{0:T}) \prod_{i=1}^T w_i(X_{0:i}) | X_{0:t} \right\}$ for $1 \leq t \leq T$, define

$$\begin{aligned}\epsilon_{2t-1}^m &= \sum_{i:A_{t-1}^m=m} \{\psi(\tilde{X}_{0:t}^i) - \Psi_{t-1}(x_{0:t-1}^i)\} H_{t-1}^i, \\ \epsilon_{2t}^m &= \sum_{i:A_{t-1}^m=m} (\#_t^m - mW_t^i) \{\Psi_t(\tilde{X}_{0:t}^i) H_t^{B_t^i} - \Psi_0\},\end{aligned}$$

in which $W_t^i = w_t^i / \sum_{j=1}^M w_t^j$, $\#_t^i$ is the number of copies of $\tilde{X}_{0:t}^i$ generated by bootstrap resampling from $\{\tilde{X}_{0:t}^1, \dots, \tilde{X}_{0:t}^M\}$ in Algorithm 1 (where the B_t^i are also defined). Then $(\#_t^1, \dots, \#_t^M) \sim \text{Multinomial}(M; W_t^1, \dots, W_t^M)$ and

$$\hat{\psi}_T - \psi_T = \{(\bar{w}_1 \cdots \bar{w}_T)^{-1} \eta_T\} M^{-1} \sum_{m=1}^M (\epsilon_1^m + \cdots + \epsilon_{2T-1}^m); \quad (1.11)$$

see Eq.(3.3) and (3.36) of [9], which shows $\{\epsilon_t^m, 1 \leq t \leq 2T-1\}$ is a martingale difference sequence and that $(\bar{w}_1 \cdots \bar{w}_T)^{-1} \eta_T = 1 + o_p(1)$ under the integrability assumptions $\eta_T < \infty$ and $\mathbb{E}_q[\prod_{i=1}^T w_i^2(X_{0:t})] < \infty$, and which then uses these to prove Theorem 1.1.

1.3 Monte Carlo Markov Chain with known posterior likelihood $\pi(\theta|Y_{1:t})$

Section 1.2 introduces SMC to solve the case of unknown latent states but known model parameters. In this section, we switch to the other way around, assuming known latent states (thus posterior likelihood $\pi(\theta|Y_{1:t})$ can be easily calculated up to some multiplicative constant) but unknown model parameters.

1.3.1 Metropolis-Hastings MCMC

SMC requires good proposal distributions to mimic target distribution as close as possible. When efficient proposal distributions are prohibitive, Monte-Carlo Markov Chain (MCMC) is developed to circumvent this problem. The basic idea of MCMC method is to construct a Markov chain, where the Markov state is a realized random variable that we want to sample and state transition is the generation of a new sample from the old sample, such that the stationary distribution of this Markov chain is the target distribution. The mechanism that a new sample is generated from an old sample is designed to exploit the information of old samples and thus can still make sampling efficient when good proposal distributions are restrictive.

In the context of HMMs, when parameters θ are unknown but posterior density $\pi(\theta|Y_{1:t})$ is known to some multiplicative constant (i.e., $\pi(\theta|Y_{1:t}) \propto \psi(\theta)$ with known $\psi(\theta)$), we can sample θ from this target distribution with MCMC methods. MCMC achieves this by maintaining a Markov chain $\{\theta^1, \theta^2, \dots, \theta^K, \dots\}$ such that the limit distribution of θ^K is $\pi(\theta|Y_{1:t})$. There are many types of MCMC algorithms in this framework, among which Metropolis-Hastings(MH) type is the most commonly used. Algorithm 2 summarizes the state transition for MH scheme MCMC. In practice, we first iterate MH transitions for some large number κ times to make the Markov chain (asymptotically) reach stationarity. This procedure is called the burn-in period. Then we iterate MH type transition for another K times to collect $\theta^{\kappa+1}, \dots, \theta^{\kappa+K}$ as samples (asymptotically) distributed from posterior density $\pi(\theta|Y_{1:t})$. Then we can estimate values of interest by $\mathbb{E}(\phi(\theta)|Y_{1:t}) \approx \frac{1}{K} \sum_{k=\kappa}^{\kappa+K} \phi(\theta^k)$, where $\phi(\theta)$ is a pre-specified function of random variable θ .

Algorithm 2 Metropolis-Hastings transition from θ^{k-1} to θ^k

1. Sample $\tilde{\theta} \sim q(\cdot | \theta^{k-1})$
 2. Sample $\alpha \sim \text{Uniform}(0, 1)$
 3. If $\alpha < \max(1, \frac{q(\theta^{k-1} | \tilde{\theta})}{q(\tilde{\theta} | \theta^{k-1})} \frac{\psi(\tilde{\theta})}{\psi(\theta^{k-1})})$, set $\theta^k := \tilde{\theta}$; otherwise, $\theta^k := \theta^{k-1}$.
-

1.3.2 MCMC with sequential substitution

Although MCMC methods with MH iterations are widely used computational tools in Bayesian inference on a parameter $\theta \in \Theta$ that has prior density function with respect to some measure $\nu_{\theta'}$, they do not have convergence rate guarantees in terms of the number of iterations to automate termination of the iterations. On the other hand, if the posterior density $\pi(\cdot | Y_{1:T})$ were known and easy to sample from, then standard Monte Carlo could be used to generate i.i.d. $\theta_1, \dots, \theta_N$ from $\pi(\cdot | Y_{1:T})d\nu_\theta$ and estimate $\phi := \mathbb{E}[\phi(\theta) | Y_{1:T}]$ by the sample average $\hat{\phi} = N^{-1} \sum_{n=1}^N \phi(\theta_n)$, with estimated standard error $\hat{\sigma}_N / \sqrt{N}$, where $\hat{\sigma}_N^2 = (N-1)^{-1} \sum_{n=1}^N (\phi(\theta_n) - \hat{\phi})^2$, assuming that $\mathbb{E}\phi^2(\theta) < \infty$. Moreover, by the central limit theorem, $\hat{\phi} \pm N^{-1/2} \hat{\sigma}_N z_{1-\alpha/2}$ is an approximate $(1 - \alpha)$ -level confidence interval for ϕ , from which one can determine N to ensure $\hat{\phi}$ to be some prescribed tolerance limit z of ϕ via $N^{-1/2} \hat{\sigma}_N z_{1-\alpha/2} \leq \epsilon$; we use z_q to denote the q th quantile of the standard normal distribution. This has motivated Lai et al. [34] to develop the following novel MCMC procedure that is asymptotically equivalent (as $N \rightarrow \infty$) to the oracle case that assumes known posterior distribution, for which importance sampling can be used if the known posterior distribution is difficult to sample from.

Algorithm 3 Updating Procedure $SS(\Theta_{b,k}, \omega_k^b)$ for MCMC-SS

1. Sample $\tilde{\theta}$ from $f(\cdot | \gamma(\Theta_{b,k-1}))$ as candidate atom.
 2. Let $\theta_{\nu+1,k-1}^b = \tilde{\theta}$ and compute $\lambda_{i,k}^b = \frac{f(\theta_{i,k-1}^b | \gamma(\Theta_{b,k-1}))}{\psi(\theta_{i,k-1}^b)}, i = 1, \dots, \nu + 1$.
 3. Sample J from $\{1, \dots, \nu + 1\}$ with probability $\pi_{i,k}^b = \lambda_{i,k}^b / \sum_{j=1}^{\nu+1} \lambda_{j,k}^b$ for i .
 4. If $J = \nu + 1$, let $\Theta_{b,k} = \Theta_{b,k-1}$. Otherwise let $\Theta_{b,k} = (\Theta_{b,k-1} \cup \{\tilde{\theta}\}) \setminus \{\theta_{J,k-1}^b\}$.
 5. Let $\omega_{i,k}^b = 1/\pi_{i,k}^b$ for $i = 1, \dots, \nu$, and $\omega_k^b = (\omega_{1,k}^b, \dots, \omega_{\nu,k}^b)$.
-

As in the MH procedure, let $\psi(\theta)$ be a given function that is proportional to the target density $\pi(\theta | Y_{1:T})$, treating the observations Y_1, \dots, Y_T as constants. Let $\{f(\cdot | \gamma), \gamma \in \Gamma\}$ be a family of positive proposal densities with respect to ν_θ , where Γ is a convex subset of \mathbb{R}^d . MCMC-SS initializes by choosing $\gamma_0 \in \Gamma_0$ and generating νB i.i.d. $(\theta_{1,0}^1, \dots, \theta_{1,0}^\nu); \dots; (\theta_{B,0}^1, \dots, \theta_{B,0}^\nu)$ from the proposal distribution $f(\cdot | \gamma_0)dm$, thereby forming the B disjoint sets $\Theta_{b,0} = \{\theta_{b,0}^1, \dots, \theta_{b,0}^\nu\}$. At stage k , it uses a sequential substitution procedure $SS(\Theta_{b,k})$ to update the atom set in the b th block and to assign the weight $\omega_{i,k}^b$ to the i th atom in $\Theta_{b,k}, b = 1, \dots, B$, as summarized in Algorithm 3. MCMC-SS estimates ϕ by

$$\hat{\phi} = \frac{1}{B(K - \kappa)} \sum_{b=1}^B \sum_{k=\kappa+1}^K \hat{\phi}_{b,k}, \text{ with } \hat{\phi}_{b,k} = \frac{\sum_{i=1}^\nu \omega_{i,k}^b \phi(\theta_{i,k}^b)}{\sum_{i=1}^\nu \omega_{i,k}^b}, \quad (1.12)$$

where κ represents an initial burn-in period that is asymptotically negligible as $K \rightarrow \infty$ and $B \rightarrow \infty$, $\kappa \succ \log B$ but $\kappa = o(K)$. In many applications, the parameter γ of the family of proposal densities is a function $\gamma : \mathcal{P} \rightarrow \Gamma$, where \mathcal{P} is the space of

probability measures on Θ . Assuming this framework, we now describe the choice of $\gamma_{b,k-1}$ in Algorithm 3. For $k \leq \kappa$, let $\gamma_{b,k-1} = \nu^{-1} \sum_{\theta \in \Theta_{b,k-1}} \gamma(\theta)$, which is the mean of the empirical measure of the atoms in the b th block at the end of stage $k-1$. On the other hand, for $k > \kappa$, we pool across blocks by letting $\tilde{\gamma}_{k-1} = B^{-1} \sum_{b=1}^B \gamma_{b,k-1}$, which we use as the modified $\gamma_{b,k-1}$ for all blocks. Therefore, after the burn-in period, we can carry out the update $\text{SS}(\Theta_{b,k})$ in the order $b = 1, \dots, B$, so that if the candidate atom in $\text{SS}(\Theta_{b,k})$ is not used for block b , it can serve as candidate atom for block $b+1$ ($\leq B$) and thereby obviates the need to generate another random variable from $f(\cdot | \tilde{\gamma}_{k-1})$, an obvious advantage for high-dimensional and complicated states. Lai et al. [34] have proved the following result and shown how it implies the aforementioned oracle properties of MCMC-SS.

Theorem 1.2. *Suppose $\mathbb{E}\phi^2(\theta) < \infty$ and there exist $\beta > \alpha > 0$ and $V : \Theta^\nu \rightarrow [1, \infty)$ such that*

$$\int_{\Theta^\nu} V(\boldsymbol{\theta}) f(\theta^1 | \gamma_0) \cdots f(\theta^\nu | \gamma_0) d\nu_\theta(\theta^1) \cdots d\nu_\theta(\theta^\nu) < \infty, \quad (1.13)$$

$$\alpha V(\boldsymbol{\theta}) \leq \lambda(\tilde{\theta} | \gamma(\boldsymbol{\theta})) \leq \beta V(\boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta} \in \Theta^\nu \text{ and } \tilde{\theta} \in \Theta, \quad (1.14)$$

where $\lambda(\tilde{\theta} | \gamma) = f(\tilde{\theta} | \gamma) / \pi(\tilde{\theta} | Y_{1:T})$.

(a) Let $G_{b,k}$ be the joint distribution of $(\theta_{1,k}^b, \dots, \theta_{\nu,k}^b)$ and let F^ν be the probability measure on Θ^ν that has the density of ν independent components each of which has density $f(\cdot | \gamma_p)$ with respect to ν_θ , where $\gamma_p = \text{argmin}_{\gamma \in \Gamma} I(f_\gamma || \pi)$ and $I(f || \pi) = \mathbb{E}_\pi \{ \log(\pi(\theta) / f(\theta)) \}$ is the Kullback-Leibler divergence of f from π , in which we have used f_γ and π to denote $f(\cdot | \gamma)$ and $\pi(\cdot | Y_{1:T})$, respectively. Then there exist positive constants a and c such that $\|G_{b,k} - F^\nu\|_V \leq ce^{-\alpha k}$ for $1 \leq k \leq K$, where

$\|\cdot\|_V$ denotes the weighted total variation norm associated with the weight function V . Hence after $k \succ \log B$ iterations, $\sum_{b \leq B} \|G_{b,k} - F^\nu\|_V \rightarrow 0$.

(b) Let $N = \nu B(K - \kappa)$ be the total number of atoms used in (1.12). Then as $K \rightarrow \infty$ and $B \rightarrow \infty$ such that $B = O(K)$, $\sqrt{N}(\hat{\phi} - \phi) \Rightarrow N(0, v)$, in which v can be consistently estimated by

$$\hat{v} = \frac{1}{B(K - \kappa)} \sum_{b=1}^B \sum_{k=\kappa+1}^K \frac{1}{\nu - 1} \sum_{i=1}^{\nu} (\phi(\theta_{i,k}^b) - \hat{\phi}_{b,k})^2. \quad (1.15)$$

1.4 Joint estimation of hidden states and unknown parameters

For general case where both model parameters θ and latent states $X_{0:t}$ are unknown, there are many attempts to learn the two components jointly. In this section, we give a brief introduction to different approaches, including augmented particles filters, traditional MCMC methods, particle based MCMC methods, and numerical methods.

1.4.1 Augmenting the state with the parameter

The first attempt is to augment state of particle filter with model parameters so that estimating posterior density of $\mathbb{P}(\theta, X_{0:T} | Y_{1:T})$ can be easily achieved by standard particle filter algorithm. Let augmented state be

$$Z_t := (X_t, \theta_t),$$

with initial density

$$\mathbb{P}(Z_0) = \mathbb{P}(\theta_0, X_0) = \pi(\theta_0)p_0(X_0|\theta_0),$$

and transition density

$$\mathbb{P}(Z_t|Z_{t-1}) = \mathbb{P}(\theta_t, X_t|\theta_{t-1}, X_{t-1}) = p_t(X_t|X_{t-1}, \theta_t)\delta_{\theta_{t-1}}(\theta_t),$$

where $\delta_{\theta_{t-1}}(\theta_t)$ is a delta density concentrated on θ_{t-1} . However, this augmented state does not have the forgetting property of many real world Markov models, which prevents particle filter algorithm from degenerating. Therefore, the augmented state particle filter algorithm will fail eventually. In fact, the exploration of parameter $\theta \in \Theta$ will only happen at the first random initialization for the initial state (X_0, θ_0) . Later, each propagation step of particle filter algorithm will reduce the diversity of the particles because there are no dynamics for parameter θ_t . When more observations Y_t arrives, the particles will degenerate into a single particle (X_t^*, θ_t^*) .

To solve this degeneration issue, Kitagawa [31] proposes a practical variant via an artificial dynamics of the model parameter:

$$\theta_{t+1} = \theta_t + \epsilon_t, \tag{1.16}$$

where ϵ_t is a noise term to bring diversity for θ . However, the introduction of this artificial noise leads to bias that is difficult to quantify. In addition, these methods requires tedious tuning for hyper-parameters such as variance bandwidth of the noise.

1.4.2 Gibbs sampling

Researchers have used MCMC methods to generate samples from posterior distribution for Bayesian models. State space models are also Bayesian models that require sampling from the posterior distribution of $X_{0:t}, \theta|Y_{1:t}$. Since state space models usually contain high-dimensional latent states, Gibbs sampling, a special kind of MCMC algorithm, is particularly suitable for this setting. Several attempts [20],[27] have been made to use Gibbs sampling on HMMs like continuous-time financial models. In particular, Eraker et al. [20] publish a paper in which they use Gibbs sampling to get posterior mean of model parameters as well as hidden volatilities for stochastic volatility model (SV), stochastic volatility model with jumps in returns (SVJ), stochastic volatility model with independent jump in returns and volatility (SVIJ) and stochastic volatility model with contemporaneous jumps in returns volatility (SVCJ). An overview of stochastic volatilities model can be found in Section 3.3 and A.3. Eraker et al. [20] compare estimations from these different models and find strong evidence for jumps in volatilities and jumps in returns. Letting Θ_i be the i th component of model parameters, $J_{t\Delta}$ be a boolean variable to indicate a jump happened at time t , $\xi_{t\Delta}^y, \xi_{t\Delta}^v$ be jump sizes for returns and volatilities respectively and $V_{t\Delta}$ be the hidden volatility at time t . They conduct the following Gibbs sampling sequence:

parameters : $\mathbb{P}(\Theta_i|\Theta_{-i}, J, \xi^y, \xi^v, V, Y), i = 1, \dots, k$

jump times : $\mathbb{P}(J_{t\Delta} = 1|\Theta, \xi^y, \xi^v, V, Y), t = 1, \dots, T$

jump sizes : $\mathbb{P}(\xi_{t\Delta}^y|\Theta, J_{t\Delta} = 1, \xi^v, V, Y), t = 1, \dots, T$

: $\mathbb{P}(\xi_{t\Delta}^v|\Theta, J_{t\Delta} = 1, V, Y), t = 1, \dots, T$

volatility : $\mathbb{P}(V_{t\Delta}|V_{(t+1)\Delta}, V_{(t+1)\Delta}, \Theta, J, \xi^y, \xi^v, Y), t = 1, \dots, T$

However, due to the high correlations of sampled variables, Gibbs sampling tends to get stuck in local optima and fails to discover the true high likelihood area of posterior density. In Appendix A.3, we also demonstrate this phenomena by a simulation experiment on SVCJ model. Moreover, Belaygorod et al [5] attempt to solve this problem by delicately designed block Gibbs sampling, but their algorithm (1) only partly mitigates the issue instead of avoiding it, and (2) is too specific to generalize. As pointed out by Johanners et al [27, Section 3.5.2 and 6], blocking is an important issue for Gibbs sampling method and has to be dealt with carefully. They also emphasize that MCMC can only be used as off-line algorithm and it is important to develop sequential (online) algorithms for practical applications.

1.4.3 Particle marginal Metropolis-Hastings and SMC²

Andrieu et al. [2] and Chopin et al. [11] consider time-homogeneous Markov chains $X_t \sim p_\theta(\cdot|X_{t-1})$ for $t \geq 1$ and $X_0 \sim p_\theta(\cdot)$, with latent states and observations $Y_t \sim g_\theta(\cdot|X_t)$, in which θ is an unknown parameter with a prior density function $\pi(\cdot)$ with respect to some measure ν_θ on the parameter space Θ . The posterior density of $(\theta, X_{0:T})$ given $Y_{1:T}$ is proportional to

$$p_T(\theta, x_{0:T}) = \pi(\theta)p_\theta(x_0) \prod_{t=1}^T \{p_\theta(x_t|x_{t-1})g_\theta(Y_t|x_t)\}. \quad (1.17)$$

PMCMC uses SMC, involving M particles (each of which consists of a sampled parameter and state trajectory) at every iteration k to construct an approximation \hat{p}_T to p_T in a Metropolis-Hastings (MH) MCMC scheme that uses a proposal density

Algorithm 4 PMCMC at the k th iteration, initialized with $\theta^0 \sim f(\cdot)$

1. $\theta^* \sim f(\cdot | \theta_{k-1})$
 2. Run SMC (Algorithm 1) to generate M particles $X_{0:t}^{m,k}$ with corresponding weights $w_T^{m,k}$. Let $\hat{p}_T(\theta^*) = \sum_{m=1}^M w_T^{m,k}$.
 3. Accept θ^* with probability $1 \wedge \{\hat{p}_T(\theta^*)f(\theta_{k-1}|\theta^*)\}/\{\hat{p}_T(\theta_{k-1})f(\theta^*|\theta_{k-1})\}$.
 4. If θ^* is accepted, let $\theta^k = \theta^*$ and $(X_{0:t}^{m,k}, w_T^{m,k})$ be the corresponding weighted particles.
-

$f(\cdot | \theta_{k-1})$ with respect to the measure ν_θ to sample θ_k at the k th iteration, as summarized in Algorithm 4. Chopin et al. [11, Sect. 1.2] point out the difficulties in the asymptotic analysis of PMCMC as k becomes infinite. In particular, although [2] shows that under some strong assumptions, the MCMC scheme converges to a measure in total variation norm as $k \rightarrow \infty$, for fixed value of M , the limiting measure is not the target posterior distribution of $(\theta, X_{0:t})$. On the other hand, allowing M to approach ∞ with k would lead to an analytically intractable scheme involving state spaces whose dimensions change with k . Chopin et al. [11] propose the SMC² scheme to target heuristically the posterior distribution of $(\theta, X_{0:t})$ given $Y_{1:t}$ ($1 \leq t \leq T$) as follows. It involves N θ -particles, which we will call ‘‘atoms’’, and attaches to each atom θ a particle filter that propagates and resamples M particles (state trajectories $X_{0:t}^m$) generated by SMC (as in Algorithm 1 with the given θ). It carries out the MH iterations to determine if a candidate atom is accepted (as in Step 3 of Algorithm 4). For the N atoms $\theta_t^1, \dots, \theta_t^N$ and their corresponding importance weights at time t generated in this way, if the degeneracy criterion in Chopin [10] is satisfied, carry out bootstrap resampling of the weighted parameter-particle set to replace it by an unweighted set. The procedures are summarized in Algorithm 5. Chopin et al [11]

give a “formal justification” of SMC² in their Proposition 1 and numerical studies in their Section 4, to show its better performance over several other methods in the existing literature, although no convergence theory as $k \rightarrow \infty$ is proved in [11].

Algorithm 5 SMC²

- *Initialize for $t = T_0$:*
 1. run N independent PMCMC(Algorithm 4) given initial observation $y_{1:T_0}$ to get $S = \{\theta^1, \dots, \theta^N\}$ together with their associated particles $\{\tilde{X}_{1:T_0}^{n,m}, w_{T_0}^{n,m}, \hat{p}_{\theta^n}(y_{1:T_0})\}$.
 2. initialize atom weight $w(\theta^n) := 1$ for all n
 - *For $t = T_0 + 1, \dots, T$:*
 1. run SMC(Algorithm 1) starting from $\{\tilde{X}_{0:t-1}^{n,m}, w_{t-1}^{n,m}, \hat{p}_{\theta^n}(y_{1:t-1})\}$ to get $\{\tilde{X}_{0:t}^{n,m}, w_t^{n,m}, \hat{p}_{\theta^n}(y_{1:t})\}$ for all n
 2. update atom weight $w(\theta^n) := w(\theta^n) \frac{\hat{p}_{\theta^n}(y_{1:t})}{\hat{p}_{\theta^n}(y_{1:t-1})}$ for all n
 3. if coefficient of variation of $w(\theta^n)$ exceeds certain threshold:
 - bootstrap from $S = \{\theta^1, \dots, \theta^N\}$ according to weight $w(\theta^n)$ with replacement, to get \tilde{S} and reset atom weight $w(\theta^n) := 1$ for all n
 - run N independent PMCMC(Algorithm 4) given observation $y_{1:t}$ starting from \tilde{S} to bring diversity and independence.
-

1.4.4 Particle Gibbs sampling

Other than a particle version of Metropolis-Hastings MCMC, Andrieu et al [2] also propose a particle version of Gibbs sampling (PG sampler). They propose to sample alternatively from $\mathbb{P}(\theta|Y_{1:T}, X_{0:T})$ and $\mathbb{P}_\theta(X_{0:T}|Y_{1:T})$ as in Algorithm 7. Although sampling from $\mathbb{P}(\theta|Y_{1:T}, X_{0:T})$ is normally easy (though potentially tedious), sampling from $\mathbb{P}_\theta(X_{0:T}|Y_{1:T})$ is typically impossible. Hence, they apply a particle approximation to $\mathbb{P}_\theta(X_{0:T}|Y_{1:T})$. They propose a conditional SMC update to achieve this as in Algorithm 6. The conditional update is based on an input of pre-specified latent path $X_{0:T}$. Other remaining $N - 1$ particles remain the same as in a normal SMC algorithm.

Algorithm 6 Conditional SMC update

Step 1 let $X_{0:T} = \left(X_0^{B_0}, X_1^{B_1}, \dots, X_{T-1}^{B_{T-1}}, X_T^{B_T} \right)$ be a path that is associated with the ancestral lineage $B_{0:T}$.

Step 2 for $t = 0$,

- (a) for $k \neq B_0$, sample $X_0^k \sim q_\theta(\cdot|Y_1)$ and
- (b) compute $w_0^k := p_0(X_0^k)/q_0(X_0^k)$ and normalize weights $W_0^k \propto w_0^k$

Step 3 for $t = 1, \dots, T$

- (a) for $k \neq B_t$, sample index A_{t-1}^k from weights $W_{t-1}^1, \dots, W_{t-1}^n$
 - (b) for $k \neq B_t$, sample $X_t^k \sim q(\cdot|Y_{1:t}, X_{t-1}^{A_{t-1}^k})$ and
 - (c) compute $w_t^k := p_t(X_t^k|X_{t-1}^{A_{t-1}^k})g_t(Y_t|X_t^k)/q_t(X_t^k|Y_{1:t}, X_{t-1}^{A_{t-1}^k})$ and normalize weights $W_t^k \propto w_t^k$
-

When number of particles $N \rightarrow \infty$, PG sampler converges to Gibbs sampling. However, in practice, PG sampler requires the same magnitude of number of particles

Algorithm 7 Particle Gibbs sampler

Step 1 initialization, $i = 0$ – set $\theta(0)$, $X_{0:T}(0)$ and $B_{1:T}(0)$ arbitrarily.

Step 2 for iterations $i \geq 1$,

- (a) sample $\theta(i) \sim \mathbb{P}(\cdot | Y_{1:T}, X_{0:T}(i-1))$
 - (b) run a conditional SMC algorithm targeting $\mathbb{P}_{\theta(i)}(X_{0:T}|Y_{1:T})$ conditional on $X_{0:T}(i-1)$ and $B_{0:T}(i-1)$, and
 - (c) sample $X_{0:T}(i), B_{0:T}(i) \sim \hat{\mathbb{P}}_{\theta(i)}(\cdot | Y_{1:T})$, where $\hat{\mathbb{P}}$ is the empirical distribution from particles in step (b)
-

as in PMMH in order to mix well for sampled path $(X_{0:T}, B_{0:T})$. Compared to Particle marginal Metropolis-Hastings, which converges to Metropolis-Hastings MCMC, PG sampler is not very efficient thus not practical. Fortunately, in recent years, Lindsten et al. publish a series of papers [35, 36, 37] that propose an improved PG sampler called Particle Gibbs with ancestor sampling (PGAS), which successfully reduce the requirements of number of particles via a variant conditional SMC update in Algorithm 8. This algorithm also keep the pre-specified path $X_{0:T}$, but additionally sample ancestor $A_t^{B_t}$ as well. Through this, mixing of latent states $X_{0:T}$ can be greatly improved. In their simulation studies, they demonstrate $N = 1000$ particles of PG sampler is comparatively efficient as $N = 10$ particles of PGAS. Chopin et al. [12] further provide a formal theory for PGAS.

PGAS algorithm is very useful when model parameters space $\theta \in \Theta$ is high-dimensional, where Metropolis-Hastings type algorithms fail. But reader needs to notice that PGAS can only be used as an off-line algorithm due to the nature of Gibbs sampling.

Algorithm 8 Conditional SMC update with ancestor sampling

Step 1 let $X_{1:T} = \left(X_1^{B_1}, X_2^{B_2}, \dots, X_{T-1}^{B_{T-1}}, X_T^{B_T} \right)$ be a path that is associated with the ancestral lineage $B_{1:T}$.

Step 2 for $t = 0$,

- (a) for $k \neq B_0$, sample $X_0^k \sim q_0(\cdot)$ and
- (b) compute $w_0^k := p_0(X_0^k)/q_0(X_0^k)$ by equation xx and normalize weights $W_0^k \propto w_0^k$

Step 3 for $t = 1, \dots, T$

- (a) for $k \neq B_t$, sample index A_{t-1}^k from weights $W_{t-1}^1, \dots, W_{t-1}^n$
 - (b) for $k \neq B_t$, sample $X_t^k \sim q(\cdot | Y_{1:t}, X_{t-1}^{A_{t-1}^k})$
 - (c) compute $\tilde{w}_{t-1|T}^k := w_{t-1}^k \frac{\mathbb{P}_\theta(X_{0:t-1}^k, X_{t:T}^{B_t^k} | Y_{1:T})}{\mathbb{P}_\theta(X_{0:t-1}^k | Y_{1:t-1})}$ and normalize weights $\tilde{W}_{t-1|T}^k \propto \tilde{w}_{t-1|T}^k$
 - (d) sample $A_{t-1}^{B_{t-1}}$ from weights $\tilde{W}_{t-1|T}^1, \dots, \tilde{W}_{t-1|T}^n$
 - (e) compute $w_t^k := p_t(X_t^k | X_{t-1}^{A_{t-1}^k}) g_t(Y_t | X_t^k) / q(X_t^k | Y_{1:t}, X_{t-1}^{A_{t-1}^k})$ and normalize weights $W_t^k \propto w_t^k$
-

1.4.5 Markov chain approximation method

All the aforementioned works have been focusing on the inference of HMMs via statistical sampling. However, Zeng first investigates partially observed framework of Markov processes by a numerical scheme in his thesis [47]. Zeng develops corresponding Bayesian inference via filtering equations to deal with unknown model parameters. He characterizes the evolution of the posterior distribution by a normalized filtering equation. He presents a weak convergence theorem and provides a consistent recursive algorithm for computing Bayesian estimates for streaming financial transaction data. A more recent paper by Hu et al. [25] illustrates this approximation algorithm to four specific models built for the U.S. Treasury notes transactions data. Further

review and discussion of their model are described in Section 3.3. Here we give a brief introduction of their recursive algorithm. Considering model parameters θ and hidden states X_t with a grid gratuity of ϵ , this algorithm discretizes the parameter space and hidden state space into $\theta_{\vec{v}} \in \Theta_v$ and $X_w \in \mathcal{X}_w$ respectively. Given previous discretized posterior estimation $p_\epsilon(\theta_{\vec{v}}, x_w; t_i)$, they propagate the time integral to the next time step with an Euler scheme by

$$\begin{aligned} p_\epsilon(\theta_{\vec{v}}, x_w; t_{i+1}-) \approx & p_\epsilon(\theta_{\vec{v}}, x_w; t_i) + [\beta_v(\theta_{\vec{v}}, x_{w-1}) p_\epsilon(\theta_{\vec{v}}, x_{w-1}; t_i) \\ & - (\beta_v(\theta_{\vec{v}}, x_w) + \delta_v(\theta_{\vec{v}}, x_w)) p_\epsilon(\theta_{\vec{v}}, x_w; t_i) \\ & + \delta_v(\theta_{\vec{v}}, x_{w+1}) p_\epsilon(\theta_{\vec{v}}, x_{w+1}; t_i)] (t_{i+1} - t_i), \end{aligned} \quad (1.18)$$

where β, δ_v are functions derived from normalized filtering equations. Then given this propagated unnormalized posterior density, they normalize it by

$$p_\epsilon(\theta_{\vec{v}}, x_w; t_{i+1}) = \frac{p_\epsilon(\theta_{\vec{v}}, x_w; t_{i+1}-) p(y_j|x_w, \rho_m)}{\sum_{\vec{v}', w'} p_\epsilon(\theta_{\vec{v}'}, x_{w'}; t_{i+1}-) p(y_j|x_{w'}, \rho_{m'})} \quad (1.19)$$

Hence, posterior density $p_\epsilon(\theta_{\vec{v}}, x_w; t_{i+1})$ is recursively calculated. Zeng's method is sequential and recursive, making it suitable for online filtering. However, this method is purely numerical with no statistical sampling procedures. It suffers the curse of dimensionality. In fact, when parameter θ or hidden states X_t are not bounded, there is no way to approximate the space well with a finite grid.

Chapter 2

A new adaptive particle filter

Following PMCMC and SMC² in Section 1.4.3, consider a latent time-homogeneous Markov chain $X_t \sim p_\theta(\cdot|X_{t-1})$ and observations $Y_t \sim g_\theta(\cdot|X_t)$, with unknown parameter θ that has prior density $\pi(\cdot)$ with respect to some measure ν_θ . The posterior density of $(\theta, X_{0:t})$ given $Y_{1:T}$ is proportional to $p_T(\theta, x_{0:T})$ given by (1.17). In Sections 2.1 and 2.2, we introduce an off-line adaptive procedure that combines MCMC-SS with SMC for joint parameter and state estimation and establish its asymptotic optimality by considering the oracle case in which the posterior density $\pi(\theta|Y_{1:T})$ is known explicitly. Section 2.3 considers oracle scenarios for group sequential or recursive procedures, and a recursive MCMC-SS/SMC filter is developed in Section 2.4 that attains the oracle bound. Computational and implementation details are given in Section 2.5.

2.1 Oracle case in which $\pi(\theta|Y_{1:T})$ is known and easy to sample from

One can sample N i.i.d. $\theta^n \sim \pi(\cdot|Y_{1:T})$ and $\tilde{\mathbb{X}}_T^n|\theta^n = (\tilde{X}_{0:t}^{1,n}, \dots, \tilde{X}_{0:t}^{M,n})$ consisting of M particles generated by SMC with θ^n as the parameter in p_θ and g_θ . Estimate $\psi := \mathbb{E}[\psi(X_{0:T})|Y_{1:T}]$ by $\hat{\psi} = \frac{1}{N} \sum_{n=1}^N \hat{\psi}^n$, where $\hat{\psi}^n = (\sum_{m=1}^M w_t^m \psi(\tilde{X}_{0:T}^{m,n})) / \sum_{m=1}^M w_T^m$. The classical central limit theorem with consistent standard error estimate can be applied to $\hat{\psi}^n$, which can be combined with the CLT of Chan and Lai [9] for particle filters with known parameter θ (giving limiting variance $\sigma^2(\theta)$) to yield the following.

Theorem 2.1. *Under certain integrability conditions, $\sqrt{MN}(\hat{\psi} - \psi) \Rightarrow N(0, \sigma^2)$, in which $\sigma^2 = \int \sigma^2(\theta) d\pi(\theta|Y_{1:T})$ can be consistently estimated by*

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{n=1}^N \hat{\sigma}_n^2, \quad \hat{\sigma}_n^2 = \frac{1}{M} \sum_{m=1}^M \left\{ \sum_{i:A_{T-1}^i=m} \frac{w_T^{i,n}}{\bar{w}_T^n} [\psi(\tilde{X}_{0:t}^{i,n}) - \hat{\psi}^n] \right\}^2. \quad (2.1)$$

2.2 Combining MCMC-SS with SMC

Without assuming the posterior density $\pi(\cdot|Y_{1:T})$ of θ to be known explicitly so that i.i.d. samples can be drawn from it, recall that the posterior density of $(\theta, X_{0:T})$ given $Y_{1:T}$ is proportional to p_T given by (1.17), hence

$$\pi(\theta|Y_{1:T}) \propto \psi(\theta) := \mathbb{E}[p_T(\theta, X_{0:T})|\theta]. \quad (2.2)$$

The conditional expectation in (2.2) can be evaluated by (1.10) that involves M particles $\tilde{X}_{0:T}^1, \dots, \tilde{X}_{0:T}^M$ generated by SMC, as described in Algorithm 1, for given θ . Using (2.2) to evaluate the target density modulo a normalizing constant, we can

Algorithm 9 MCMC-SS/SMC at the k th iteration for $k > \kappa$

1. For $b = 1, \dots, B$, update $SS(\Theta_{b,k}, \omega_k^b)$ as in Algorithm 3 for MCMC-SS, using target density $\psi(\theta)$ in (2.2).
2. For each $\theta_{i,k}^b \in \Theta_{b,k}$, use Algorithm 1 to generate M particles $\tilde{X}_{0:T}^{b,i,k,m}$ ($m = 1, \dots, M$) by SMC.
3. Parameter estimation: Estimate $\phi := \mathbb{E}[\phi(\theta)|Y_{1:T}]$ by (1.12).
4. State estimation: Estimate $\psi := \mathbb{E}[\psi(X_{0:T})|Y_{1:T}]$ by

$$\hat{\psi} = \frac{1}{N} \sum_{k=\kappa+1}^K \sum_{b=1}^B \frac{\sum_{i=1}^\nu \omega_{i,k}^b \hat{\psi}_{i,k}^b}{\sum_{i=1}^\nu \omega_{i,k}^b}, \hat{\psi}_{i,k}^b = \frac{\sum_{m=1}^M w_T^{b,i,k,m} \psi(\tilde{X}_{0:T}^{b,i,k,m})}{\sum_{m=1}^M w_T^{b,i,k,m}}. \quad (2.3)$$

apply MCMC-SS as in Algorithm 3 to determine the posterior density $\pi(\cdot|Y_{1:T})$ of θ . In particular, by Theorem 1.2(a), after the burn-in period $\kappa \succ \log B$ with $\kappa = o(K)$, the joint distribution of the atoms $\theta_{i,k}^b$ in $\Theta_{b,k}$ behaves like the ν -fold product measure F^ν on Θ^ν , which in turn implies that for $k > \kappa$, the atom set $\{\theta_{i,k}^b : 1 \leq i \leq \nu, 1 \leq b < B\}$ at iteration k behaves like an i.i.d. sample of atoms drawn from $\pi(\cdot|Y_{1:T})d\nu_\theta$ as in the oracle case. This suggests combining MCMC-SS for parameter estimation with SMC for state estimation at iteration $k > \kappa$ of the MCMC-SS scheme into MCMC-SS/SMC for joint parameter and state estimation described in Algorithm 9, for which we have the following.

Theorem 2.2. *The conclusions of Theorem 2.1 also hold for $\hat{\psi}$ defined by (2.3), but with (2.1) replaced by*

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{k=\kappa+1}^K \sum_{b=1}^B \hat{\sigma}_{b,k}^2, \hat{\sigma}_{b,k}^2 = \frac{1}{M\nu} \sum_{m=1}^M \sum_{i=1}^\nu \left\{ \sum_{j:A_{T-1}^i=m} \frac{w_T^{b,i,k,j}}{\bar{w}_T^{b,i,k}} [\psi(\tilde{X}_{1:T}^{b,i,k,j}) - \hat{\psi}_{i,k}^b] \right\}^2, \quad (2.4)$$

where $N = \nu B(K - \kappa)$ is the total number of atoms, hence MN is the total number of trajectories. Moreover, $\sqrt{N}(\hat{\phi} - \psi) \Rightarrow N(0, v)$, in which v can be consistently estimated by (1.15).

2.3 Group sequential oracle procedure

Note that SMC described in Algorithm 1 (assuming known θ) is a recursive procedure. When θ is unknown and replaced by the posterior distribution $\theta|Y_{1:T}$ in the oracle case of Section 2.1, the recursive feature of the particle filter is lost because $X_{0:t}^n$ requires θ^n to be generated from $\pi(\cdot|Y_{1:t})$, which is different from $\pi(\cdot|Y_{1:t-1})$. To restrict to recursive procedures, we apply a group sequential approach that provides the posterior distribution $\theta|Y_{1:t_j}$ for the state update of $X_{t_j+1:t}$ for $t_j < t \leq t_{j+1}$, $j = 1, 2, \dots$, but with importance weights adjusted to reflect posterior distribution $\theta|Y_{1:t}$ change. This also gives additional flexibility by allowing θ to be piecewise constant in specifying the prior distribution that includes jumps (regime switches, or change-points), as will be illustrated in Section 3.3 where details on the oracle procedure are also given.

We next consider the case without parameter jumps. Douc et al.[16] have introduced a novel information technique to analyze the likelihood function $L_t(\theta)$ of a parametric family of HMMs, which they use to prove the strong consistency of the maximum likelihood estimator based on $Y_{1:t}$ under “a rather minimal set of assumptions.” Their result also shows that the posterior distribution $\pi(\cdot|Y_{1:t})d\nu_\theta$ is also strongly consistent in the sense of converging to the probability measure degenerate at the true parameter. Hence in this case there is no loss in asymptotic efficiency (as $t \rightarrow \infty$) in using the posterior distribution $\theta|Y_{1:t_j}$ for the state update of $X_{t_j+1:t}$ for $t_j < t \leq t_{j+1}$, $j = 1, 2, \dots$

Theorem 2.3. Suppose with probability 1, $\pi(\cdot|Y_{1:t})d\nu_\theta$ converges weakly to the probability measure degenerate at the true parameter generating the HMM in Theorem 2.1. Then $\sigma_t^2 := \int \sigma^2(\theta)d\pi(\theta|Y_{1:t}) \rightarrow \sigma^2(\theta^*)$ a.s., where θ^* is the true parameter as in Section 1.1. Hence the group sequential oracle scheme is asymptotically as efficient as the off-line oracle scheme as $t \rightarrow \infty$.

2.4 Recursive MCMC-SS/SMC filter

This recursive adaptive particle filter is based on recursive representations of $\pi(\theta|Y_{1:t_j})d\nu_\theta$, $j = 1, 2, \dots$, that can be implemented by MCMC-SS (instead of assuming known $\pi(\theta|Y_{1:t_j})d\nu_\theta$ as in Section 2.3) to update $\Theta_{b,k}(t_{j-1})$ and $\omega_k^b(t_{j-1})$ at iteration $k > \kappa$ as in Algorithm 3 so that for each $\theta_{i,k}^b \in \Theta_{b,k}(t_{j-1})$, Algorithm 1 can be used to generate M particles $\tilde{X}_{t_{j-1}+1:t_j}^{b,i,k,m}$ ($m = 1, \dots, M$) by SMC. Beginning with time t_1 , perform $K > \kappa$ iterations as in Algorithm 9 that yields an atom set $\cup_{1 \leq b \leq B, \kappa+1 \leq k \leq K} \Theta_{b,k}(t_1)$ consisting of $N_1 = \nu B(K - \kappa)$ atoms. The atom set with N_1 atoms provides the aforementioned representation of $\pi(\theta|Y_{1:t_1})d\nu_\theta$ to update $\Theta_{b,K+1}(t_2)$, which we describe more generally for $\Theta_{b,K+j}(t_j)$ as follows. Note that by (1.17),

$$p_{t_j}(\theta, X_{0:t_j}) = p_{t_{j-1}}(\theta, X_{0:t_{j-1}}) \left\{ \prod_{i=t_{j-1}+1}^{t_j} p_\theta(X_t|X_{t-1}) g_\theta(Y_t|X_t) \right\}. \quad (2.5)$$

Instead of taking conditional expectation given θ as in (2.2) for the off-line MCMC-SS/SMC procedure, we combine the recursive representations of $\pi(\theta|Y_{t_1+1:t_2}), \dots, \pi(\theta|Y_{t_{j-1}+1:t_j})$ with the aforementioned representation of $\pi(\theta|Y_{1:t_1})$ to derive from (2.5)

the recursive adaptive particle filter

$$\psi_{t_j}(\theta) = \psi_{t_1}(\theta)\psi_{t_1+1:t_2}(\theta)\cdots\psi_{t_{j-1}+1:t_j}(\theta) \quad (2.6)$$

for $j \geq 2$, where $\psi_{t_1}(\theta) = \mathbb{E}[p_{t_1}(\theta, X_{0:t_1})|\theta]$,

$$\psi_{t_{j-1}+1:t_j}(\theta) = \mathbb{E}_{\mathcal{X}(t_{j-1})} \left\{ \prod_{t=t_{j-1}+1}^{t_j} p_\theta(X_t|X_{t-1})g_\theta(Y_t|X_t) \middle| \theta \right\}, \quad (2.7)$$

and $\mathbb{E}_{\mathcal{X}(t_{j-1})}$ denotes expectation under the initial distribution of $X_{t_{j-1}}$ that can be represented by MN_j particles as described below. First note that after $n_1 := K(> \kappa)$ iterations of the MCMC-SS algorithm performed at time t_1 , n_2 iterations of Algorithm 3 are carried out to update $\Theta_{b,n_1+n_2}(t_2)$ from $\Theta_{b,n_1}(t_1)$, and then n_3 iterations to update $\Theta_{b,n_1+n_2+n_3}(t_3)$ from $\Theta_{b,n_1+n_2}(t_2)$, etc. The ψ in Step 2 of Algorithm 3 now corresponds to ψ_{t_1} for $\Theta_{b,n_1}(t_1)$, $\psi_{t_1+1:t_2}$ given by (2.7) for the update $\Theta_{b,n_1+n_2}(t_2)$, $\psi_{t_2+1:t_3}$ for $\Theta_{b,n_1+n_2+n_3}(t_3)$, etc. This recursive adaptive particle filter will be denoted by *recMCMC-SS/SMC*. The choice of n_2, n_3, \dots will be discussed at the end of this section.

To evaluate (2.7) for a given value of θ , we first show that the distribution of $X_{t_{j-1}}$ can be simulated by using MN_j particles, where $N_j = \nu B(n_1 + \dots + n_{j-1} - \kappa)$. Suppose $\Theta_{b,k}(t_h)$, $1 \leq b \leq B$, $\kappa + 1 \leq k \leq n_1 + \dots + n_h$, has been generated; in particular, for $h = 1$, $\Theta_{b,k}(t_1)$ is generated by Algorithm 9 for $\kappa + 1 \leq k \leq n_1$. Then for each $\theta_{i,k}^b \in \Theta_{b,k}(t_h)$, we can use SMC to generate M particles $\tilde{X}_{t_h+1:t_{h+1}}^{b,i,k,m}$, with weight $w_{t_{h+1}}^{b,i,k,m}$, for $m = 1, \dots, M$. Using bootstrap resampling converts them to M particles $X_{t_h+1:t_{h+1}}^{b,i,k,m}$ ($m = 1, \dots, M$) with equal weights. In particular, for $h+1 = j-1$,

the empirical measure of

$$\Omega(t_{j-1}) := \{X_{t_{j-1}}^{b,i,k,m} : 1 \leq b \leq B, 1 \leq i \leq \nu, \quad (2.8)$$

$$k = n_1 + \dots + n_{j-1}, 1 \leq m \leq M\},$$

can be used to represent $\mathcal{X}(t_{j-1})$ in (2.7) so that given θ we can use SMC (Algorithm 1) to generate M particles $\tilde{X}_{t_{j-1}:t_j}^1(\theta), \dots, \tilde{X}_{t_{j-1}:t_j}^M(\theta)$ after initializing each by a random draw of $\tilde{X}_{t_{j-1}}^m$ from $\Omega(t_{j-1})$, i.e. $\tilde{X}_{t_{j-1}}^1, \dots, \tilde{X}_{t_{j-1}}^M$ are i.i.d. random variables such that $P(\tilde{X}_{t_{j-1}}^m = x) = (MN_j)^{-1}$ for $x \in \Omega(t_{j-1})$. Using (1.10) to estimate $\psi_{t_{j-1}+1:t_j}(\theta)$ then provides an estimator that differs from $\psi_{t_{j-1}+1:t_j}(\theta)$ by $O_p(1/\sqrt{M})$ by Theorem 1.1. The recursive adaptive particle filter is initialized at time t_1 with output $\Theta_{b,n_1}(t_1), 1 \leq b \leq B$, and $\Omega(t_1)$. Algorithm 10 summarizes how it is updated at time $t_j (j > 1)$ from the output $\Omega(t_{j-1})$ and $\Theta_{b,n_1+\dots+n_{j-1}}(t_{j-1}), 1 \leq b \leq B$, at time t_{j-1} .

Algorithm 10 Update of recMCMC-SS/SMC at time t_j , giving $\Omega(t_j)$ and $\Theta_{b,n_1+\dots+n_j}(t_j)$ as output

Input: $\Omega(t_{j-1})$ and $\Theta_{b,n_1+\dots+n_{j-1}}(t_{j-1})$, $1 \leq b \leq B$

1. For $k = n_1 + \dots + n_{j-1}$ and $\theta_{b,k} \in \Theta_{b,k}(t_{j-1})$, $1 \leq b \leq B$, sample i.i.d. $\tilde{X}_{t_{j-1}}^{b,i,k,1}, \dots, \tilde{X}_{t_{j-1}}^{b,i,k,M}$ from $\Omega(t_{j-1})$.
 2. Generate $\tilde{X}_{t_{j-1}+1:t_j}^{b,i,k,m}$ with weights $w_{t_j}^{b,i,k,m}$ for $1 \leq m \leq M$ by SMC.
 3. Resample from $\tilde{X}_{t_j}^{b,i,k,m}$ with probability $\propto w_{t_j}^{b,i,k,m}$ ($1 \leq m \leq M$), to convert them to $X_{t_j}^{b,i,k,m}$ with equal weights and thereby define $\Omega(t_j)$ by (2.8) with $j-1$ replaced by j .
 4. For $b = 1, \dots, B$, update $SS(\Theta_{b,n_1+\dots+n_{j-1}}(t_{j-1}), \omega_{n_1+\dots+n_{j-1}}^b)$ by Algorithm 3 in which the target density $\psi(\theta)$ is $\psi_{t_{j-1}:t_j}(\theta)$ given by (2.7), which is evaluated by (1.10) involving M particles $\tilde{X}_{t_{j-1}:t_j}^m$ ($1 \leq m \leq M$) for given θ and $\tilde{X}_{t_{j-1}}^m$ generated in Step 1.
-

Recall that the basic idea underlying the asymptotic efficiency of the off-line MCMC-SS/SMC procedure is that MCMC-SS provides a good atom-set representation of the posterior distribution $\pi(\theta|Y_{1:T})d\nu_\theta$, as explained in the paragraph preceding Theorem 2.2. Although recMCMC-SS/SMC does not seem to be related to it, this posterior distribution of θ is involved implicitly in (2.8) in which the states $X_{t_{j-1}}^{b,i,k,m}$ ($m = 1, \dots, M$) are associated with $\theta_{i,k}^b \in \Theta_{b,k}(t_{j-2})$ that we use in the MCMC-SS approximation of the posterior distribution $\pi(\theta|Y_{1:T})d\nu_\theta$. The key idea behind recMCMC-SS/SMC is (2.6) that allows recursive updating, whereas the off-line MCMC-SS/SMC procedure takes conditional expectation of the right-hand side

of (2.5) given θ as in (2.2) that results in a non-recursive procedure. Under the assumptions of Theorem 2.3, $\pi(\cdot|Y_{1:t})d\nu_\theta$ converges weakly as $t \rightarrow \infty$ to the probability measure degenerate at θ^* , with probability 1. Hence the conclusion of Theorem 2.3 also holds for recMCMC-SS/SMC.

Theorem 2.4. *Under the assumptions of Theorem 2.3, recMCMC-SS/SMC has the same asymptotic properties as the off-line MCMC-SS/SMC scheme at $T = t_j$ for $j = 1, 2, \dots$, as $t_j \rightarrow \infty, M \rightarrow \infty, K \rightarrow \infty, B \rightarrow \infty$ such that $B = O(K)$, and $\kappa \succ \log B$ but $\kappa = o(K)$.*

Theorem 2.4 suggests the following choice of n_2, n_3, \dots in Algorithm 10:

$$n_2 = \dots = n_J = K (= n_1), n_{(i-1)J} = \dots = n_{iJ} = \lceil K/2^i \rceil \quad (2.9)$$

for $i \geq 2$, where $\lceil x \rceil$ is the smallest positive integer $\geq x$ and J is an integer $\geq \nu$. Thus the total number of iterations in Step 4 of Algorithm 10 is bounded by $2JK$.

2.5 Computational methods and implementation

In Algorithm 3 to implement MCMC-SS, by caching the evaluations of the likelihood $\psi(\cdot)$ for all of the atoms, we only need to evaluate $\psi(\cdot)$ once per iteration for the proposal $\tilde{\theta}$. Note that this has the same cost as Metropolis-Hastings. In addition, we need to evaluate $f(\cdot|\gamma(\Theta_{b,k-1}))$ once for each $\theta_{i,k-1}^b$ for a total of νB evaluations per iteration. This calculation can be vectorized and efficiently computed in parallel. For problems where the likelihood evaluation is much more expensive than the evaluation of the proposal density (as is the case for SMC), the additional running time due to the νB proposal evaluations is small, and convergence can be significantly faster

for MCMC-SS compared to Metropolis-Hastings due to the adaptive proposals in MCMC-SS. In addition, since MCMC-SS can substitute any one of the ν atoms in the block with the proposal depending on their likelihood ratios, MCMC-SS has a much higher acceptance rate, which is advantageous when the likelihood calculation is expensive, than Metropolis-Hastings. In Algorithm 9 to implement MCMC-SS/SMC, the combination of MCMC-SS in the outer loop and SMC in the inner loop allows joint learning of posterior parameters θ (which is efficient with MCMC-SS since the proposal distribution adapts to the posterior) and nonlinear filtering of nonlinear state-space models with adaptive model parameters θ through SMC.

Chapter 3

Simulation studies and applications

In this chapter, we illustrate how the methods and results in Section 2.3 and 2.4 can be applied to adaptive particle filters in automatic navigation, high-frequency econometrics of transactions in electronic trading and stochastic volatility models incorporating jumps in returns and volatility. Further discussion of these topics and related literature will be given in Chapter 4.

3.1 A linear Gaussian model with unknown model parameters

For linear state space models $X_{t+1} = A_t X_t + w_{t+1}$, $Y_t = B_t X_t + \varepsilon_t$, in which $X_t \in \mathbb{R}^d$, A_t and B_t are given coefficient matrices, $w_t \sim N(0, \Sigma)$ and $\varepsilon \sim N(0, V)$ are independent random disturbances, the Kalman filter has been widely used in automatic navigation since its introduction in 1960. In this section, we consider the simplest univariate linear state space model. We derive analytical expressions for posterior

densities $\mathbb{P}(X_{0:T}|Y_{1:T})$ and $\pi(\theta|Y_{1:T})$. Then we can compare the empirical results from recMCMC-SS/SMC with the ground truth for demonstration purposes. We visualize the results from recMCMC-SS/SMC and empirically verify the asymptotic theory.

Now, consider the following state space model,

$$X_t = \varphi_1 X_{t-1} + \sigma_1 \epsilon_t + c \quad (3.1)$$

$$Y_t = \varphi_2 X_t + \sigma_2 \xi_t \quad (3.2)$$

where $X_0 = 0$, ϵ_t and ξ_t are two normal Gaussian noises. $(c, \sigma_1, \sigma_2, \varphi_1, \varphi_2)$ are model parameters.

In order to get the ground truth of posterior information, we first derive the true posterior densities of $X_{0:T}|\theta, Y_{1:T}$ and $\theta|Y_{1:T}$.

Theorem 3.1. *The posterior distribution $X_t|Y_{1:T}, \theta$ is a Gaussian distribution $N(\mu_t, \lambda_t)$, with the following recurrence relation of μ_t and λ_t .*

$$\begin{aligned} \mu_{t+1} &= \frac{\sigma_2^2 (\varphi_1 \mu_t + c) + \varphi_2 (\varphi_1^2 \lambda_t + \sigma_1^2) Y_{t+1}}{\varphi_2^2 (\varphi_1^2 \lambda_t + \sigma_1^2) + \sigma_2^2} \\ \lambda_{t+1} &= \frac{\sigma_2^2 (\varphi_1^2 \lambda_t + \sigma_1^2)}{\varphi_2^2 (\varphi_1^2 \lambda_t + \sigma_1^2) + \sigma_2^2} \end{aligned} \quad (3.3)$$

Theorem 3.2. *The posterior distribution $\theta|Y_{1:T}$ is*

$$\pi(\theta|Y_T) \propto \sigma_1^{-T} \sigma_2^{-T} \exp \left(-\frac{Tc^2}{2\sigma_1^2} - \frac{1}{2\sigma_2^2} \sum_{t=1}^T y_t^2 \right) |\mathbf{A}|^{-1/2} \exp(\mathbf{b}' \mathbf{A}^{-1} \mathbf{b}) \quad (*) \quad (3.4)$$

where

$$\mathbf{A} = \frac{1}{2\sigma_1^2} \begin{pmatrix} 1 + \varphi_1^2 & -\varphi_1 & 0 & \cdots & 0 \\ -\varphi_1 & 1 + \varphi_1^2 & -\varphi_1 & \ddots & \vdots \\ 0 & -\varphi_1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 + \varphi_1^2 & -\varphi_1 \\ 0 & \cdots & 0 & -\varphi_1 & 1 \end{pmatrix} + \frac{\varphi_2^2}{2\sigma_2^2} \mathbf{I}$$

and

$$\mathbf{b} = \frac{c}{2\sigma_1^2} \begin{pmatrix} 1 - \varphi_1 \\ 1 - \varphi_1 \\ \vdots \\ 1 - \varphi_1 \\ 1 \end{pmatrix} + \frac{\varphi_2}{2\sigma_2^2} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T-1} \\ y_T \end{pmatrix}$$

To make the demonstration simple, we treat c as an unknown parameter and all other parameters as known. Then we only need to sample from $\mathbb{P}(c|Y_{1:T})$ and $\mathbb{P}(X_{0:T}|Y_{1:T})$. We design the following numerical experiment:

1. Generate 1000 realizations of $X_{0:T}$ and $Y_{1:T}$, where $T = 50$
2. Use offline MCMC-SS/SMC to estimate $\mathbb{E}(\theta, X_{10}|Y_{1:10})$ and its standard error
3. Use online recMCMC-SS/SMC to estimate $\mathbb{E}(\theta, X_t|Y_{1:t})$ and its standard error recursively, where $t = 11, \dots, 50$
4. For both offline and online estimate, we compute the fraction of 1 s.e. and 2 s.e. confidence intervals that contains the true mean of parameter c and hidden state X_T .

We setup the hyper-parameters of the experiment as follows:

1. For SMC algorithm, we choose $M = 100$ particles and naive proposal $q(X_{t+1}|X_t, Y_{1:t}) = f(X_{t+1}|X_t)$. This choice of proposal contains no information on observations $Y_{1:t}$ but is suitable to test the performance.
2. For MCMC-SS algorithm, we choose $N = 100$ atoms and a naive Gaussian kernel proposal.
3. We set the true value of model parameter $c = 0.1$.

Figure 3.1 visualizes the posterior distribution of $\pi(c|Y_{1:t})$. From Theorem 3.2, we are able to calculate the true posterior distribution and compare it with the empirical distribution from the 100 atoms. We also plot the posterior mean path of c and mark the true posterior mean value derived from the posterior analytical expression. There are several observations from Figure 3.1. First, the empirical densities match the true densities very well. Secondly, the dispersion of posterior density is squeezed when more observations $Y_{1:t}$ arrive. This is in line with the intuition that more samples of observations $Y_{1:t}$ make the estimation of c more accurate. Third, the historical posterior mean path of c converges to a steady value around the true value $c = 0.1$, indicating that MCMC-SS/SMC catches the true value well.

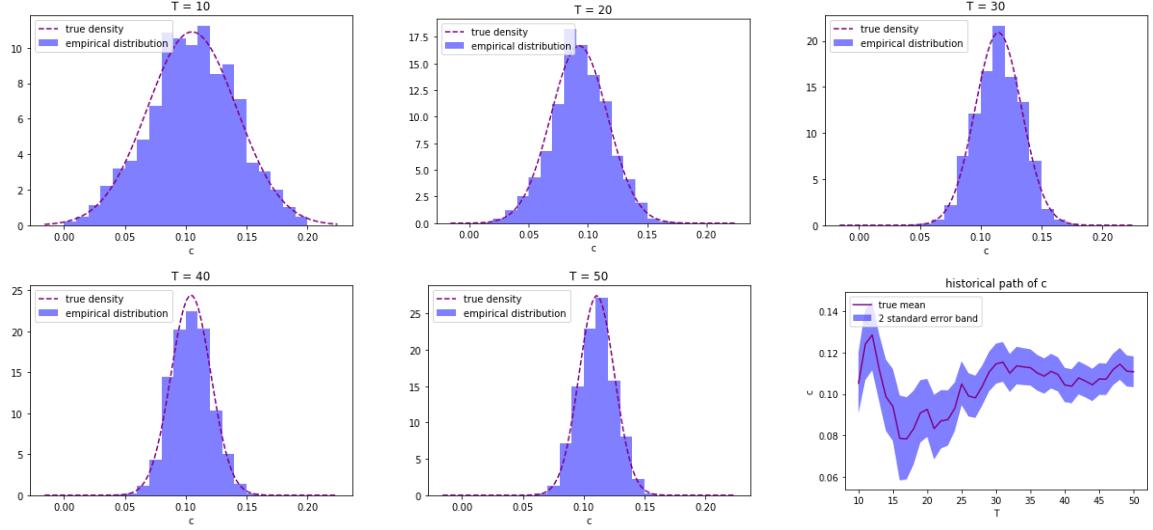


Figure 3.1: Marginal posterior distribution of model parameter c from APF estimation conditional on $Y_{1:t}$

Figure 3.2 visualizes the posterior distribution of $\mathbb{P}(X_t|Y_{1:t})$. From Theorem 3.1, we are able to calculate the true posterior distribution and compare it with the empirical distribution from the 10,000 particles. Notice that there are 100 atoms and each atom has 100 particles leading to $MN = 10,000$ particles in total. Therefore, the empirical distribution is smoother and matches the true density much better than that of the model parameter c . We also observe that the dispersion of posterior $\mathbb{P}(X_t|Y_{1:t})$ is squeezing as well, but less than $\mathbb{P}(c|Y_{1:t})$. The squeezing is no longer significant after $T = 30$. This is due to the intrinsic uncertainty of estimating hidden states X_t .

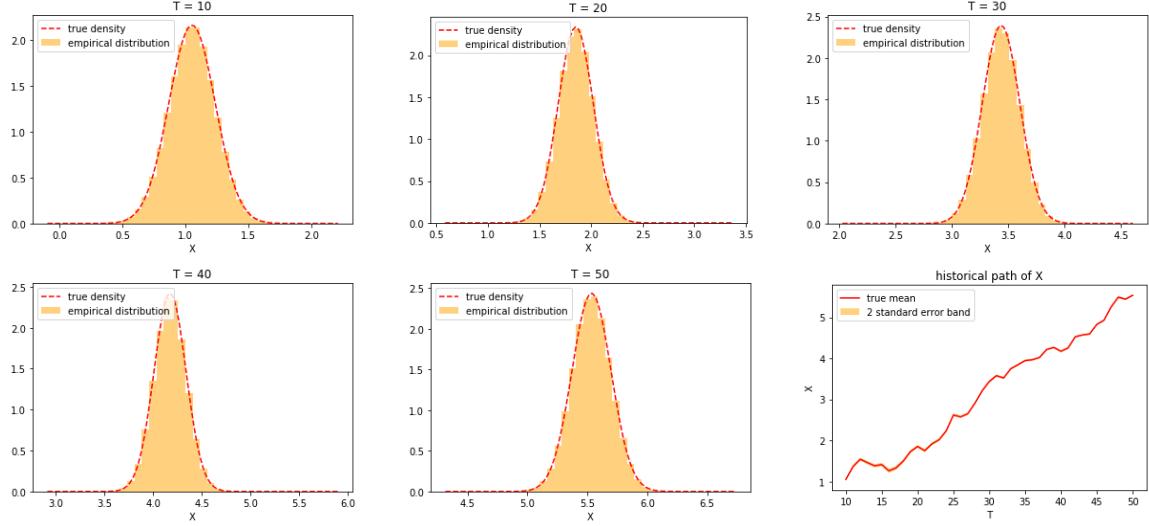


Figure 3.2: Empirical marginal posterior distributions of latent state x from APF estimation conditional on $Y_{1:t}$

Next, we conduct asymptotic normality verification for posterior mean estimation. Letting $\hat{c}|Y_{1:t}$ be the estimate of $c|Y_{1:t} := \mathbb{E}[c|Y_{1:t}]$ and $\hat{X}_t|Y_{1:t}$ be the estimate of $X_t|Y_{1:t} := \mathbb{E}[X_t|Y_{1:t}]$ from recMCMC-SS/SMC. From Theorem 2.2, MCMC-SS/SMC is able to provide an estimated standard error $\hat{\sigma}_c$ for $\hat{c}|Y_{1:t}$ and an estimated standard error $\hat{\sigma}_{X_t}$ for $\hat{X}_t|Y_{1:t}$. We define t-value

$$z_1 := (\hat{c}|Y_{1:t} - c|Y_{1:t})/\hat{\sigma}_c$$

$$z_2 := (\hat{X}_t|Y_{1:t} - X_t|Y_{1:t})/\hat{\sigma}_{X_t}$$

If the estimated standard error is correct, we should expect that $|z_1| < 1, |z_2| < 1$ have a frequency around 0.683, and $|z_1| < 2, |z_2| < 2$ have a frequency around 0.954. Table 3.1 reveals that this is indeed the case.

		within 1 s.e.	within 2 s.e.	
t	$c Y_{1:t}$	$X_t Y_{1:t}$	$c Y_{1:t}$	$X_t Y_{1:t}$
10	0.661	0.661	0.942	0.942
20	0.654	0.659	0.942	0.942
30	0.655	0.659	0.939	0.938
40	0.652	0.653	0.945	0.944
50	0.663	0.660	0.945	0.946

Table 3.1: Fraction of APF estimation within one and two standard deviations compared to true posterior mean

3.2 A nonlinear state space model with unknown parameters

The last section illustrates the application of MCMC-SS/SMC to linear state space models. Extensions to nonlinear state space models (in which $X_{t+1} - w_{t+1}$ is a nonlinear function of X_t , or $Y_t - \varepsilon_t$ is a nonlinear function of X_t , or both) include the extended Kalman filter based a linear approximation [41], and its refinement called the “unscented Kalman filter” [28, 45], Gaussian sum approximations [1], and spline approximations to the posterior densities and the density functions of w_t and ε_t [8, 14, 43]. In particular, the nonlinear state space model.

$$X_t = \frac{X_{t-1}}{2} + \frac{25X_{t-1}}{1 + X_{t-1}^2} + 8 \cos(1.2t) + w_t, \quad Y_t = \frac{X_t^2}{20} + \varepsilon_t, \quad (3.5)$$

with $w_t \sim N(0, \sigma_1^2)$, $\varepsilon_t \sim N(0, \sigma_2^2)$ and $X_0 \sim N(0, 5)$, was used in [43] to illustrate the application of the spline approximations. It was analyzed further by Kitagawa in [30, pp.1060-1062] using linear spline approximations, from which he also derived maximum likelihood estimates of σ_1^2 and σ_2^2 . Assuming known σ_1 and σ_2 , Gordon, Salmond and Smith [22] applied SMC (Algorithm 1 with $q_t(\cdot|X_{0:t-1}) = p_t(\cdot|X_{t-1})$) to evaluate the nonlinear filter for (3.5); this proposal density does not use $Y_{1:t-1}$, as pointed out by [2, p.274]. Section 3.1 of [2] considers the case of unknown $\theta := (\sigma_1, \sigma_2)$ and assumes independent prior distributions $IG(0.01, 0.01)$ for σ_1^2 and σ_2^2 , where IG stands for the inverse gamma distribution. It uses the particle marginal Metropolis-Hastings (PMMH) sampler, which is the same as PMCMC in Algorithm 4, and the particle Gibbs sampler [2, Section 2.4.3] to construct off-line adaptive particle filters for $T = 500$ observations.

We can clearly apply recMCMC/SMC to handle both parameter and state estimate online for (3.5), for which the term $p_\theta(X_t|X_{t-1})g_\theta(Y_t|X_t)$ in (2.5) is given by

$$\begin{aligned} p_\theta(X_t|X_{t-1}) &= \frac{1}{\sqrt{2\pi}\sigma_1} \exp \left\{ -\frac{1}{2\sigma_1^2} \left[X_t - \frac{X_{t-1}}{2} - \frac{25X_{t-1}}{1+X_{t-1}^2} + 8\cos(1.2t) \right]^2 \right\}, \\ g_\theta(Y_t|X_t) &= \frac{1}{\sqrt{2\pi}\sigma_2} \exp \left\{ -\frac{1}{2\sigma_2^2} \left(Y_t - \frac{X_t^2}{20} \right)^2 \right\}. \end{aligned}$$

The results of [16] apply to the likelihood function $L_t(\theta)$ of this model, hence Theorem 2.4 holds and we can use (2.9) to choose the number of iterations (n_1, n_2, \dots) in Algorithm 10. The following simulation study of the performance of recMCMC/SMC is similar to that in Section 3.1 of [2] on PMCMC (called the PMMH sampler there) and compares these two methods for estimating parameters σ_1 and σ_2 at the time

Table 3.2: Means and standard deviations (in parentheses) of Monte Carlo estimates of $\mathbb{E}(X_t|Y_{1:t})$.

T	oracle SMC	recMCMC-SS/SMC
100	-3.005(0.481)	-2.282(0.098)
200	-2.430(0.095)	-2.346(0.030)
300	17.165(0.041)	17.172(0.004)
400	13.349(0.037)	13.357(0.003)
500	-2.535(0.303)	-2.739(0.022)

horizons $t_i = 100i$ (respectively, $T = 100i$) for the recursive adaptive filter (respectively, for PMCMC), $i = 1, \dots, 5$. Instead of the particle Gibbs sampler (described in Section 1.4.4) to perform similarly to PMCMC, we consider SMC² (described in Section 1.4.3) also for comparison. To compare the performance of filtering latent states X_t , we also design a baseline where model parameters are known and an oracle SMC is conducted. Following [2], we set $\sigma_1 = \sqrt{10}$ and $\sigma_2 = 1$. We choose $M = 1000$ particles per atom for all methods. For recMCMC-SS/SMC, we choose $N = 500$ atoms. For the offline PMCMC and SMC², 40000 MCMC iterations are conducted for these methods, where the first 8000 iterations are discarded as burn-in periods. We repeat the experiment for 100 times so that empirical estimation errors can be observed and compared between different methods. Table 3.2 shows the filtered latent state X_t for recMCMC-SS/SMC and the oracle SMC. Notice they have similar estimation values but recMCMC-SS/SMC has a much smaller estimation error. This is expected because oracle SMC only has $M = 1000$ particles but recMCMC-SS/SMC has $NM = 500 \times 1000$ particles in total. From Table 3.3, it can be seen that although recMCMC-SS/SMC is an online algorithm, it has a comparable performance of estimating parameters with state-of-the-art offline procedure PMCMC and SMC².

Table 3.3: Means and standard deviations (in parentheses) of Monte Carlo estimates of $\sigma_1 (= \sqrt{10})$ and $\sigma_2 (= 1)$. The values for σ_2 are below those for σ_1 .

T	recMCMC-SS/SMC	PMCMC	SMC2
100	2.8599(0.0195)	2.8353(0.0274)	2.8304(0.0277)
	1.0835(0.0128)	1.0943(0.0157)	1.0981(0.0198)
200	3.0790(0.0160)	3.0773(0.0153)	3.0794(0.0161)
	0.9096(0.0074)	0.9099(0.0081)	0.9094(0.0095)
300	3.1720(0.0121)	3.1748(0.0123)	3.1725(0.0137)
	0.9694(0.0077)	0.9676(0.0071)	0.9705(0.0082)
400	3.0836(0.0134)	3.0861(0.0118)	3.0855(0.0114)
	1.0363(0.0062)	1.0368(0.0057)	1.0391(0.0064)
500	3.0171(0.0135)	3.0169(0.0134)	3.0172(0.0106)
	1.0617(0.0065)	1.0618(0.0070)	1.0621(0.0057)

3.3 Adaptive particle filters for latent efficient price process with marked point process observations and exogenous parameter jumps

Chapter 4 of [24] gives an overview of high-frequency econometrics, which “has emerged as an active area of research in financial econometrics” concerned with the econometric analysis of high-resolution transactions data in electronic exchanges. The bulk of high-frequency econometrics is estimation of the quadratic variation, also called “realized variance”, of an asset’s underlying efficient price process, but not much has been done on the adaptive filtering problem of evaluating the posterior distribution of the efficient price and unknown parameters given the observed transactions. Important advances in this adaptive filtering problem were made by Zeng [47] and recently by Hu, Kuipers, and Zeng [25, 26]. In particular, [25] assumes an exponential autoregressive conditional duration (EACD) model [24, Section 4.6.1]

for the point process $0 \leq T_1 < T_2 < \dots$ of transaction times, models the “marks” $Y_i = Y_i(T_i)$ that specify various market microstructure features (the price or price change, traded volume, at T_i) as in [19], and also includes a “vector process of observable economic or market factors” V_t besides (T_i, Y_i) to be the observations in the HMM, in which the latent efficient price process is diffusion $dX_t = \mu_t dt + \sigma_t dB_t$ as in [24, p.107]. The observations are related to the latent price process X_t through (a) the intensity $\lambda(\cdot | X_t, V_{t-}, F_{t-})$ of the point process of transaction times, in which $F_{t-} = \{(T_i, Y_i) : T_i < t\}$, and (b) the transaction price Y_i^P (with P denoting the price component of the mark Y_i):

$$Y_i^P = R(X_{T_i} + \varepsilon_i, 1/M), \quad (3.6)$$

where $R(\cdot, 1/M)$ is the rounding function to the nearest integer multiple of $1/M$ because of discrete price quotes in ticks $1/M, 2/M, \dots$ and ε_i is market microstructure noise [24, pp. 104,106] that is assumed to be i.i.d. with a two-sided geometric distribution [25, p.49]:

$$\mathbb{P}\{\varepsilon_i = y\} = (1 - \rho)\mathbb{I}_{\{y=0\}} + \frac{1}{2}(1 - \rho)\rho^{M|y|}\mathbb{I}_{\{y \neq 0\}}, y = 0, \pm 1/M, \pm 2/M, \dots, \quad (3.7)$$

and independent of $\{X_t, t \geq 0\}$.

In (3.7), $0 < \rho < 1$ is a parameter of the HMM. Moreover, the intensity $\lambda(\cdot | X_t, V_{t-}, F_{t-})$ given by the EACD (1,1) model for the inter-transaction times $D_i = T_i - T_{i-1}$ involve nonnegative parameters ω, γ, β in

$$D_i^* = h_i e_i, \quad h_i = \omega + \gamma D_{i-1}^* + \beta h_{i-1}, \quad (3.8)$$

where the e_i are i.i.d. exponential random variables with mean 1, $D_i^* = D / \exp(\hat{g}(T_i))$ and \hat{g} is a kernel or other nonparametric estimate of intra-day periodicity [24, p.123]. The efficient price process has parameters μ, σ_0, σ_1 and σ_2 in

$$dX_t/X_t = \mu + (\sigma_0 + \sigma_1 V_{t-}^N + \sigma_2 V_{t-}^\pm) dB_t, \quad (3.9)$$

where V_s^N is the binary variable that is equal to 1 if there is related news announcement at time s and to 0 otherwise, and V_s^\pm is the variable indicating whether the transaction at time s is bid (+) or ask (-). The main results of [25] are (i) Kushner's nonlinear filtering equation, which is a nonlinear stochastic partial differential equation (SPDE), for functionals $\psi_t = \int \psi(x) d\pi_t(x)$, where π_t is the posterior distribution of X_t given the observations up to time t , and (ii) a small-time Markov chain approximation to the SPDE that leads to a recursive filter, which can be shown to converge under certain regularity conditions when the time-mesh and the mesh used to discretize the mark space \mathbb{Y} approach 0. One such condition is that the intensity λ is $\mathcal{F}_t^{F,V}$ -predictable [25, p.43]. Although Section 3 of [25] assumes θ_t to be a latent state undergoing Markovian dynamics so that π_t is the posterior distribution of (θ_t, X_t) , subsequent sections of [25], e.g. p.53] assume θ_t to be time-invariant and can be consistently estimated from the observed transactions. We next show how MCMC-SS/SMC can provide a much more flexible approach to adaptive filters in the Hu-Kuipers-Zeng model, allowing exogenous parameter jumps, and establish its asymptotic efficiency.

Although most of the news announcements V_s^N , such as companies' earnings reports or the US fed funds rate announced by the Federal Open Market Committee, do not induce structural economic changes, some announcements result in structural

changes that can be represented by changes in the parameters of (3.7)-(3.9). We use N^* to denote this subset of news announcements. Examples include bankruptcies of companies in the 2001 dot-com crash, US invasion of Iraq in March 2003, Freddie Mac's announcement that it would no longer purchase high risk mortgages and the bankruptcy of the mortgage lender New Century Financial Corporation in 2007 that marked the beginning of the 2007-2009 Financial Crisis and the subsequent Great Recession, and the news that the US left the recession in June 2009 with the GDP recovering to 70% of the pre-recession level. To extend the recursive MCMC-SS/SMC filter to this setting, we assume that the times t_j (at which the parameter updates are made) include the times of news announcements in N^* . Hence, if t_j is not one of these times, let t_j^* be the most recent time with news in N^* . Since a new parameter regime is initialized at t_{j^*} , $n_1 := K(> \kappa)$ iterations of the MCMC-SS algorithm are performed at time t_{j^*} , n_2 iterations of Algorithm 3 are carried out to update $\Theta_{b,n_1+n_2}(t_{j^*+1})$, and n_{j-j^*} iterations are used to update $\Theta_{b,n_1+\dots,n_{j-j^*}+1}(t_j)$. Replacing $X_{0:t_j}$ (respectively, $X_{0:t_{j-1}}$) in (2.5) by $X_{t_{j^*}:t_j}$ (respectively, $X_{t_{j^*}:t_{j-1}}$), we can modify (2.8) and Algorithm 10 similarly (with t_1 replaced by t_{j^*}) and still use (2.9) to define n_2, n_3, \dots . In this framework of piecewise constant parameters θ that have jumps at times of news announcements in N^* , recMCMC-SS/SMC basically restarts itself at those times and the conclusion of Theorem 2.4 still holds as $\#\{i : t_{j^*} \leq T_i < t_j\} \rightarrow \infty$.

We now describe some details for parameter updates in recMCMC-SS/SMC. First note that the parameters ω, γ and β in (3.8) are associated with “exogenous sampling” that can be estimated by maximum likelihood (e.g., by using `garchfit` in the MATLAB toolbox `garch`) from the inter-transaction durations from t_{j^*+1} to t_{j-1} , to be used as the EACD parameters for updating the particle set $\Omega(t_j)$ as in Algorithm 10. The

intra-day periodicity \hat{g} can be estimated similarly by kernel smoothing regression (e.g., via MATLAB). By (3.9),

$$X_{T_i} = X_{T_{i-1}} \exp\left\{\left(\mu - \frac{1}{2}v_i\right)D_i + \sqrt{v_i D_i} z_i\right\}, \quad (3.10)$$

where z_i are i.i.d. standard normal random variables and $v_i = (\sigma_0 + \sigma_1 V_{T_{i-1}}^N + \sigma_2 V_{T_{i-1}}^\pm)^2$. Thus, $p_\theta(X_{T_i}|X_{T_{i-1}})$ in (2.5) is a normal density function that involves the parameters μ, σ_0, σ_1 and σ_2 of the efficient price process. The term $g_\theta(Y_{T_i}|X_{T_i})$ in (2.5) involves the parameter ρ and is given by the two-sided geometric distribution (3.7) for $\varepsilon_i = Y_{T_i} - R(X_{T_i})$, where $R(x)$ is the integer multiple of $1/M$ closest to x as in (3.6).

For the simulation study, we set true model parameters as $\mu = 50.08\%, \sigma_0 = 5.28\%, \sigma_1 = 4.99\%, \sigma_2 = -0.8\%$ and $\rho = 10\%$ for $0 \leq T \leq 4000$. We then set a parameter jump at $T = 4001$ with $\mu = 50.08\%, \sigma_0 = 10\%, \sigma_1 = 10\%, \sigma_2 = -2\%$ and $\rho = 10\%$ for $4000 < T \leq 8000$. These sets of parameters are consistent with observed transactions data as estimated in [25, table 3, model 4]. Notice that trade intensity $\lambda(t)$ is not critical in the estimation of model parameters $\mu, \sigma_0, \sigma_1, \sigma_2, \rho$ due to exogenous sampling. We follow [26] to assume a constant trade intensity of 0.0058 trades/seconds. We apply recMCMC-SS/SMC to the simulation data with $N = 800$ atoms and $M = 4000$ particles. It can be seen from Table 3.5 that the posterior means of model parameters converge to the true value.

We also design an oracle case where true model parameters are known and an oracle SMC is conducted. We compare the estimated latent efficient prices X_t from oracle SMC with the estimation from recMCMC-SS/SMC. As shown in Table 3.4, we find that even though recMCMC-SS/SMC has no information of true model parameters, it can jointly learn the latent efficient prices and model parameters very

well. There is little difference between the estimations from oracle SMC and those from recMCMC-SS/SMC. This may be due to the strong information contained in observed prices Y_t .

Table 3.4: Filtered latent prices by Oracle SMC and recMCMC-SS/SMC from the simulation data of the latent efficient price process. Estimated posterior mean errors are in parentheses

T	oracle SMC	recMCMC-SS/SMC
1000	103.5938011(0.0000241)	103.5938298(0.0000244)
2000	107.5459017(0.000182)	107.5462218(0.000194)
3000	112.265371(0.0000104)	112.2653358(0.0000114)
4000	112.2187243(2.3e-13)	112.2186766(0.0000387)
5000	105.374844(0.000098)	105.375063(0.000163)
6000	111.264562(0.000086)	111.264356(0.000311)
7000	118.656128(0.000118)	118.656018(0.00019)
8000	116.530849(0.000112)	116.530977(0.000176)

Table 3.5: RecMCMC-SS/SMC estimation of posterior means and standard deviations for model parameters of the simulated latent efficient price process

T	μ	σ_0	σ_1	σ_2	ρ
True	50.07	5.28	4.99	-0.8	10
1000	108.682(24.40)	5.376(0.201)	4.538(0.347)	-0.916(0.249)	7.198(2.167)
2000	98.022(16.274)	5.363(0.153)	4.622(0.284)	-0.883(0.184)	9.064(1.579)
3000	99.041(13.692)	5.232(0.118)	4.897(0.225)	-0.786(0.138)	9.139(1.344)
4000	80.594(12.854)	5.165(0.103)	4.923(0.188)	-0.705(0.128)	9.425(1.153)
True	50.07	10.0	10.0	-2.0	10
5000	163.7496(43.695)	10.158(0.376)	9.042(0.652)	-2.36(0.437)	11.723(3.506)
6000	133.5852(29.82)	10.158(0.264)	9.277(0.531)	-2.249(0.311)	13.837(2.68)
7000	141.34176(26.545)	9.981(0.208)	9.776(0.396)	-2.155(0.251)	13.072(2.385)
8000	104.28264(23.097)	9.864(0.183)	9.776(0.317)	-1.914(0.213)	12.435(2.132)

3.4 Adaptive particle filters in stochastic volatility models with contemporaneous jumps

Noting that models with both diffusive stochastic volatility (SV) and jumps in returns are “incapable of fully capturing the empirical feature of equity index returns or option prices”, Eraker et al [20] propose to remedy this by incorporating jumps in stochastic volatility, as in the SVIJ and SVCJ models developed by Duffie et al. [18], in which CJ refers to contemporaneous jumps in returns and volatilities and IJ refers to independently arriving jumps in both. They use an off-line MCMC method, developed by Johannes and Polson [27], to estimate the parameters and states of SVCJ, which they focus on “as it has the most complicated distributional structure” and which is defined by the SDE

$$\begin{aligned} d \log(S_t) &= \mu dt + \sqrt{v_{t-}} dZ_t^1 + \xi_t^Y dJ_t, \\ dv_t &= \kappa(\theta - v_{t-}) dt + \sqrt{v_{t-}} \sigma dZ_t^2 + \xi_t^v dJ_t, \end{aligned} \quad (3.11)$$

where S_t is the observed asset price and v_t is its latent volatility process.

We now show how recMCMC-SS/SMC can be applied to give an online particle filter that updates the parameter estimates at time t_{j-1} to generate M particles by SMC as in Steps 1-3 of Algorithm 10. Following [20, p.1276 and Appendix A], we use an Euler scheme for time-discretization of (3.11) into

$$\begin{aligned} Y_t &= \mu + \sqrt{v_{t-1}} (\rho Z_t^1 + \sqrt{1 - \rho^2} Z_t^2) + J_t \xi_t^Y, \\ v_t &= v_{t-1} + \kappa(\theta - v_{t-1}) + \sqrt{v_{t-1}} \sigma Z_t^1 + J_t \xi_t^v, \end{aligned} \quad (3.12)$$

with state transitions on a daily basis, where Z_t^1 and Z_t^2 are i.i.d. standard normal, $Y_t = \log(S_t/S_{t-1})$ is the observed daily return and v_t is the latent volatility after day t , $J_t \sim \text{Bernoulli}(\lambda)$ assumes with value 1 (respectively, 0) if a contemporaneous jump in return and volatility occurs (respectively, does not occur) on day t , with jump sizes ξ_t^v and ξ_t^Y given by

$$\xi_t^v / \mu_v \sim \text{Exponential}(1), \quad \xi_t^Y | \xi_t^v \sim N(\mu_Y + \rho_Y \xi_t^V, \sigma_Y^2). \quad (3.13)$$

The parameters of the model are $\mu, \rho, \kappa, \theta, \sigma, \lambda, \mu_v, \mu_Y, \rho_Y$ and σ_Y .

Simulation study For the simulation study, we abandon PMCMC and SMC² in this case because they cannot perform well due to the high dimension of model parameters. In order to test the online estimation of parameters for recMCMC-SS/SMC, we compare it with offline MCMC-SS/SMC. We use $N = 800$ atoms and $M = 800$ particles. To test the performance of filtering hidden volatilities, we conduct an oracle SMC similar to those in Section 3.2 and 3.3. As seen in Table 3.6, although recMCMC-SS/SMC has no information of model parameters, it can estimate the latent volatilities and jump rates well. From Table 3.7, we can see the posterior mean with two standard deviations can cover the true value. In addition, we notice that as more observations arrive, the posterior standard deviations of model parameters get smaller, which indicates a more accurate estimation.

Table 3.6: Filtered volatilities and jump rates from Oracle SMC and recMCMC-SS/SMC for the simulation data of the SVCJ process. Estimated posterior mean error are in parentheses.

T	oracle SMC		recMCMC-SS/SMC	
	$\mathbb{E}(v_t Y_{1:t})$	$\mathbb{E}(J_t Y_{1:t})$	$\mathbb{E}(v_t Y_{1:t})$	$\mathbb{E}(J_t Y_{1:t})$
1000	5.466(0.055)	0.019(0.005)	5.821(0.013)	0.023(0.001)
2000	9.005(0.062)	0.031(0.007)	9.817(0.032)	0.046(0.001)
4000	5.738(0.05)	0.412(0.026)	6.045(0.012)	0.409(0.004)

Table 3.7: RecMCMC-SS/SMC estimation of posterior mean and standard deviations for model parameters of the simulated SVCJ process

	$T=1000$		$T=2000$		$T=4000$	
True	offline	online	offline	online	offline	online
μ	0.05	0.125(0.064)	0.078(0.056)	0.085(0.061)	0.105(0.046)	0.093(0.046)
κ	0.02	0.022(0.005)	0.019(0.004)	0.019(0.004)	0.021(0.003)	0.021(0.003)
θ	0.7	0.648(0.218)	0.771(0.245)	0.721(0.239)	0.708(0.241)	0.715(0.257)
σ	0.1	0.07(0.037)	0.106(0.031)	0.1(0.029)	0.099(0.029)	0.098(0.028)
ρ	-0.5	-0.329(0.387)	-0.664(0.167)	-0.644(0.189)	-0.56(0.15)	-0.572(0.151)
λ	0.1	0.082(0.025)	0.097(0.022)	0.098(0.023)	0.12(0.021)	0.117(0.023)
μ_Y	-1.4	-1.095(1.148)	-1.219(0.908)	-1.212(0.915)	-0.698(0.649)	-0.692(0.713)
σ_Y	1.7	1.605(0.83)	1.411(0.713)	1.434(0.739)	1.348(0.65)	1.401(0.65)
μ_v	1	1.309(0.358)	1.15(0.241)	1.158(0.276)	0.94(0.15)	0.979(0.185)
ρ_Y	-3.6	-3.2(0.851)	-3.443(0.75)	-3.471(0.809)	-3.743(0.623)	-3.673(0.726)

Filtering on return data of S&P 500 We also apply recMCMC-SS/SMC on the return data of S&P 500. We obtain daily log return of S&P 500 from 1980 to 2018. Data from year 1980 to 1995 is used to get initial posterior information. Then we run recMCMC-SS/SMC to obtain filtering results on data from year 1996 to 2018. Figure 3.4 shows the results. We compare the filtered volatilities with the VIX index, a metric that reflects market opinions on spot volatilities. We find the two curves matches each other very well, which indicates that recMCMC-SS/SMC can filter useful information from pure return data. More results on the estimation path of model parameters are supplemented in Appendix A.3.4.

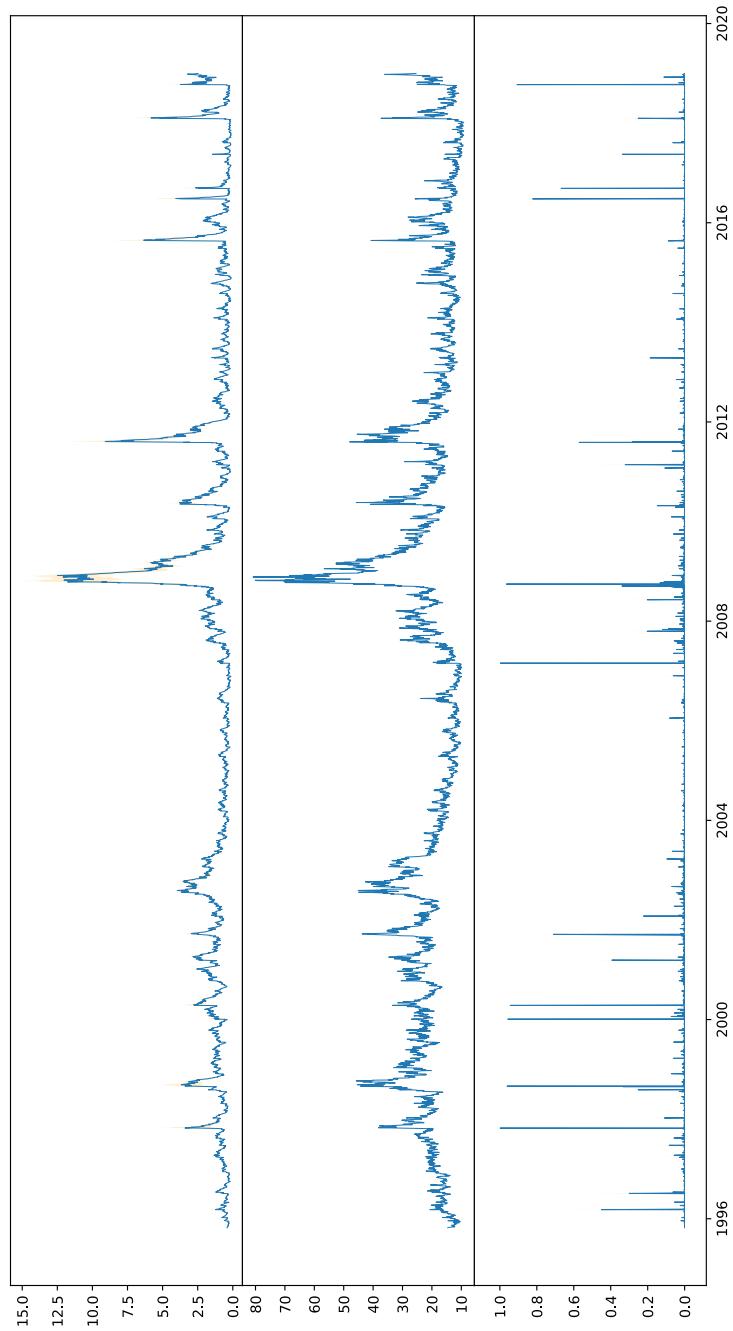


Figure 3.3: Filtered volatility $\mathbb{E}[v_t|Y_{1:t}]$, VIX and jump probability $\mathbb{E}[J_t|Y_{1:t}]$

Chapter 4

Conclusion

In real world applications, many problems can be modeled as hidden Markov models, also known as state space models, where two unknown components, latent states paths $X_{0:t}$ and model parameters θ , are inferred from observed data $Y_{1:t}$. In addition, because of the nature of Markov models, the observed sequence is growing and we need to develop online and recursive algorithms so as to satisfy the need of online filtering (instead of offline batch estimation).

Extant literature has been focusing on simpler problems because simultaneously handling unknown parameters and latent states for online filtering is too difficult. There are three simplified cases. (1) Assuming model parameters to be known, sequential Monte Carlo, also known as particle filters, is introduced. This algorithm is recursive and effective in filtering latent states given observed data and known parameters. (2) On the other hand, assuming known latent states thus tractable posterior likelihood $\pi(\theta|Y_{1:t})$, Monte Carlo Markov Chain (MCMC) algorithms are introduced to infer posterior distributions of model parameters given observed data. In particular, MCMC with sequential substitutions are recently developed, which allows

adaptive proposal and high acceptance rate of newly proposed parameter atoms with improved computation efficiency. (3) Assuming both model parameters and latent states to be unknown but focusing on the offline batch estimation, Particle Marginal Metropolis-Hastings (PMMH), Particle Gibbs sampler (PG sampler) and SMC² are developed. These algorithms use particle filters to infer latent states and another MCMC layer on top of the particles to infer model parameters.

In this thesis, we have introduced a new approach to adaptive particle filters, called MCMC-SS/SMC that accommodates all complicated assumptions – (1) unknown model parameters, (2) unknown latent states, and (3) online filtering settings. We accomplish this by combining the theory of particle filters for state estimation when the parameters are known with that of a novel MCMC scheme using sequential state substitutions for parameter estimation (MCMC-SS). The MCMC-SS provides a representative parameter atom population which are approximately distributed to posterior distributions $\pi(\theta|Y_{1:t})$ with importance weights. Then for each atom, another layer of particle filters are applied to filter latent states. This hierarchy structure (N atoms for parameters and MN particles for latent states) maintains a weighted empirical distribution, where a complete asymptotic theory can be developed.

Furthermore, MCMC-SS/SMC is also very practical as an off-the-shelf algorithm. In real world applications like stochastic volatility models in Section 3.4, where model parameters have complicated distributions with involved correlations between different dimensions, traditional methods like PMMH or PG sampler need delicate designs of proposals. However, MCMC-SS/SMC can exploit the structure in posterior densities from the representative atom sets automatically via proposal family $\gamma(\Theta)$ as in Algorithm 3. Users need no customized tunings or calibrations for MCMC-SS/SMC.

For future research, because MCMC-SS/SMC uses a two-layer structure that maintains N atoms and NM particles, it is very computationally expensive. How to reduce the number of particles needed is an open question. One promising direction is to design more efficient proposal family $\gamma(\Theta)$.

Appendix A

Numerical experiments supplements

In this chapter, we supplement more numerical experiments as well as technical details for the problems described in Chapter 3. In some of the figures or tables, due to the long abbreviation of MCMC-SS/SMC or recMCMC-SS/SMC, we will use APF (adaptive particle filter) to denote our new algorithm when there is no ambiguity.

A.1 Nonlinear state space model

In Section 3.2, we have shown a simulation to compare recMCMC-SS/SMC, PMCMC and SMC². In this section, we design simulations for online and offline scenarios separately and study various metrics to explain the superior performance of MCMC-SS/SMC.

The first experiment is for offline MCMC-SS/SMC. We test the performances of MCMC-SS/SMC, PMCMC and SMC² by estimating $\mathbb{E}(\sigma_1|Y_{1:T})$ and $\mathbb{E}(\sigma_2|Y_{1:T})$. We

simulate the state space model of (3.5) with a sample size of $T = 500$. For the number of particles, we try different configurations with $M = 500, 1000, 2000, 4000$. For the number of atoms, we set $N = 25$ atoms for SMC² and MCMC-SS/SMC. We run $K = 100,000$ iterations of PMCMC, SMC² and MCMC-SS/SMC. Samples from the first 50,000 iterations are discarded as burn-in periods. This experiment is repeated for 100 times, so that we can use 5 and 95 percentile values among the 100 simulations as error bands for different metrics.

For analysis of the algorithm performance, we examine the following metrics:

1. Empirical error of estimated posterior mean: we take the MCMC sample mean as posterior mean estimation for each run (100 runs in total) and calculate the sample standard deviation of the 100 simulations. This is the most direct metric to describe the performance of an algorithm. As can be seen in Table A.1, MCMC-SS/SMC has a substantial lower error compared to PMCMC and SMC². SMC² outperforms PMCMC by a small margin.
2. Acceptance ratio: we count the average frequency that a newly proposed atom is accepted. This metric can provide insights on why MCMC-SS/SMC converges faster than SMC². Given a higher acceptance rate, the atom sets can move more quickly and thus bring more diversity and mixing. MCMC-SS/SMC has a substantial higher acceptance rate than PMCMC and SMC². Plots are shown in Figure A.1.
3. Auto-correlation: we calculate the empirical auto-correlation between the average estimation of every 25 MCMC iterations. This metric is popular in the MCMC literature, because it shows the level of independence among iterations. As seen from Figure A.2, MCMC-SS/SMC has a substantially lower

auto-correlation than SMC². Notice that we do not include PMCMC because it is not comparable to the above definition of empirical auto-correlations.

4. Convergence speed: we compare the convergence of posterior likelihood $\pi(Y_{1:T}|\theta)$.

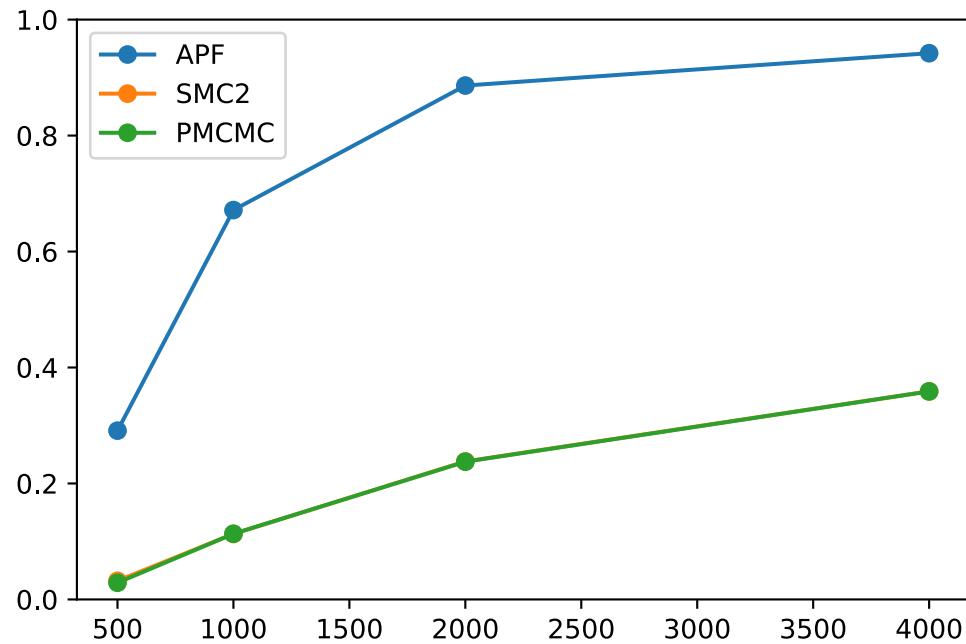
Plots are shown in Figure A.3. To refine the metric, we define the average estimated likelihood of $50,000 \sim 100,000$ MCMC iterations as stationary, and we count how many MCMC iterations are needed to reach that critical value, as shown in Table A.2. Since PMCMC has only one atom but MCMC-SS/SMC and SMC² have multiple atoms, PMCMC has the highest convergence rate. Between MCMC-SS/SMC and SMC², MCMC-SS/SMC has a substantially higher convergence rate.

#particles	$\hat{\mathbb{E}}[\sigma_1 Y_{1:t}]$ relative error			$\hat{\mathbb{E}}[\sigma_2 Y_{1:t}]$ relative error		
	PMCMC	SMC ²	APF	PMCMC	SMC ²	APF
500	197.1%	0.0132	97.0%	191.1%	0.0071	80.2%
1000	93.8%	0.0073	85.4%	109.9%	0.0035	78.8%
2000	98.7%	0.0039	59.2%	113.9%	0.0017	80.6%
4000	119.2%	0.0024	65.5%	115.9%	0.0014	57.7%

Table A.1: Sample errors for the 100 simulations. Take SMC² as baseline and calculate the relative error of PMCMC and MCMC-SS/SMC to SMC².

#particles	500	1000	2000	4000
PMCMC	2132(6238)	582(112)	498(78)	493(77)
SMC ²	34151(10736)	17685(3765)	14635(923)	14518(996)
APF	15659(8068)	5148(1960)	3012(692)	2717(554)

Table A.2: Number of MCMC iteration needed to reach stationary

Figure A.1: Plots for acceptance rates of MCMC-SS/SMC, SMC² and PMCMC, where x axis is number of particles; y axis is the average acceptance rate.

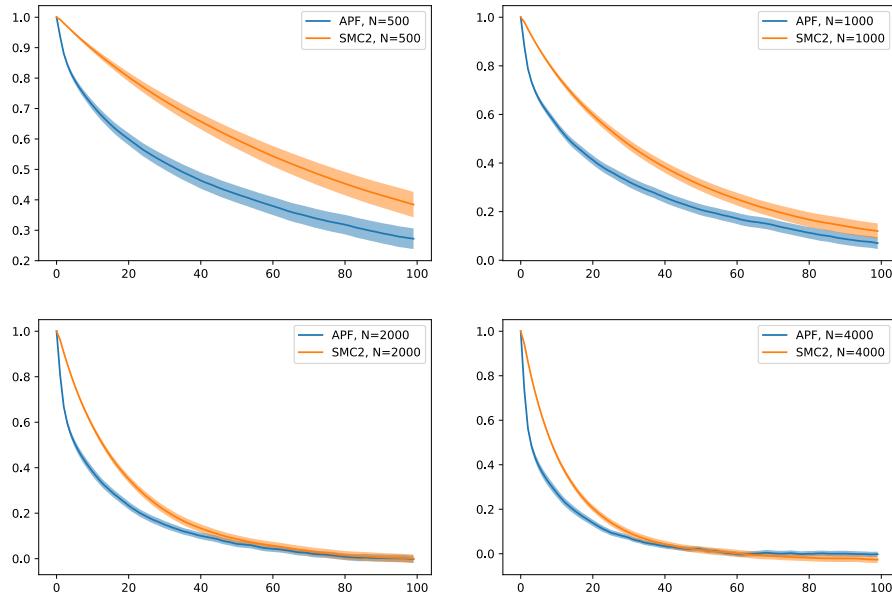


Figure A.2: Plots for auto-correlation of MCMC-SS/SMC and SMC² under different number of particles N , where x axis is number of MCMC iterations; y axis is the average auto-correlation.

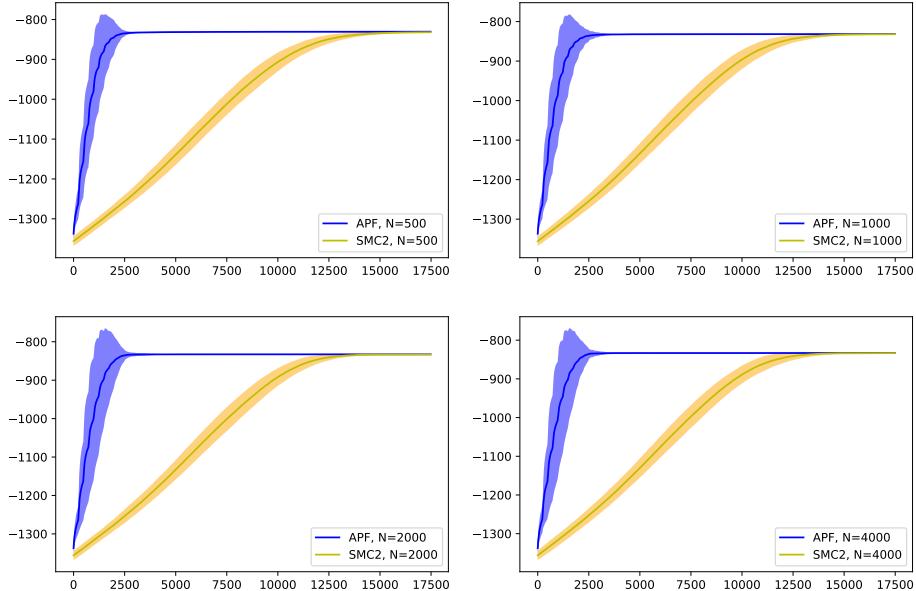


Figure A.3: Plots for convergence speed of MCMC-SS/SMC and SMC² under different number of particles N , where x axis is number of MCMC iterations; y axis is log likelihood estimated by particle filter.

The second is an online experiment for recMCMC-SS/SMC and SMC² to estimate $\mathbb{E}(\sigma_1|Y_{1:t})$ and $\mathbb{E}(\sigma_2|Y_{1:t})$, where $t = 100, \dots, 500$. Notice that PMCMC is not suitable for online estimation and is not included in this experiment. We choose $N = 2000$ atoms and $M = 2000$ particles. We conduct $K = 10,000$ MCMC iterations for each group sequential MCMC update. This experiment is also repeated for 100 times to get error bands of different metrics. Figure A.4 and A.5 show the recursive posterior mean estimations and its standard errors. Figure A.6 puts estimation errors from recMCMC-SS/SMC and SMC² together to compare the diffidence. From these figures and tables, we can see online estimation errors for recMCMC-SS/SMC are significantly lower than SMC². Essentially this is due to the faster convergence and better mixing of recMCMC-SS/SMC compared to SMC². RecMCMC-SS/SMC utilizes the information of all atoms jointly and thus can bring more diversities to the atom sets given the same number of MCMC iterations; on the contrary, SMC² treats each atom independently, leading to slower convergence and worse mixing.

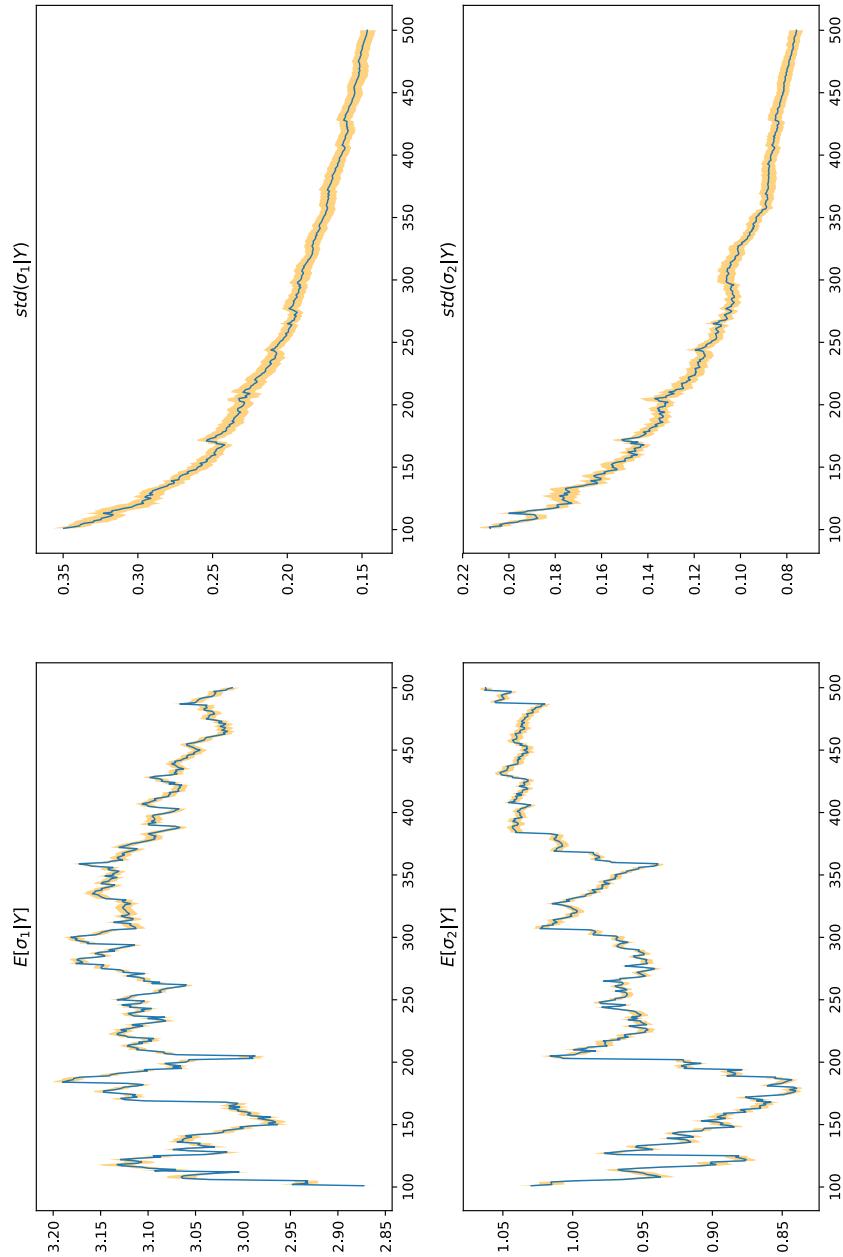
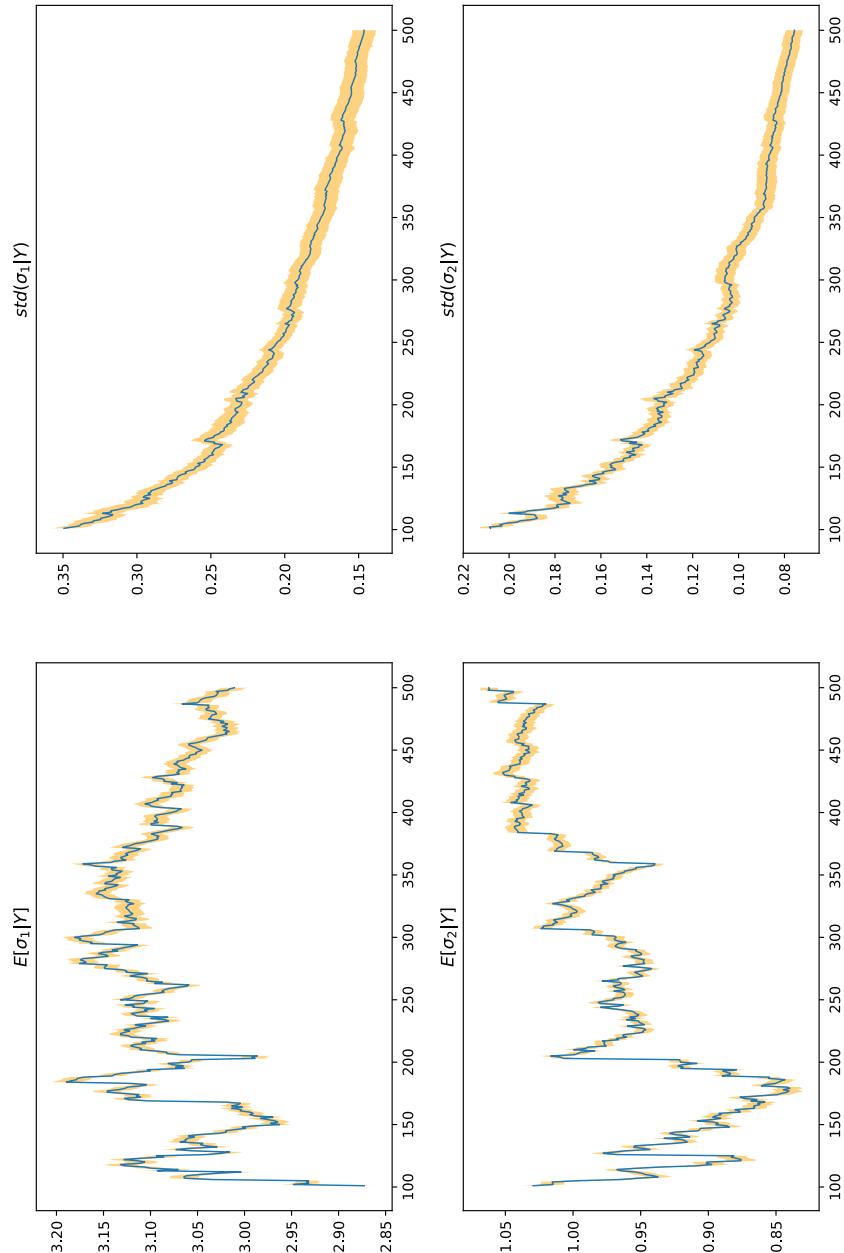


Figure A.4: Posterior mean and standard error estimated from recMCMC-SS/SMC

Figure A.5: Posterior mean and standard error estimated from SMC²

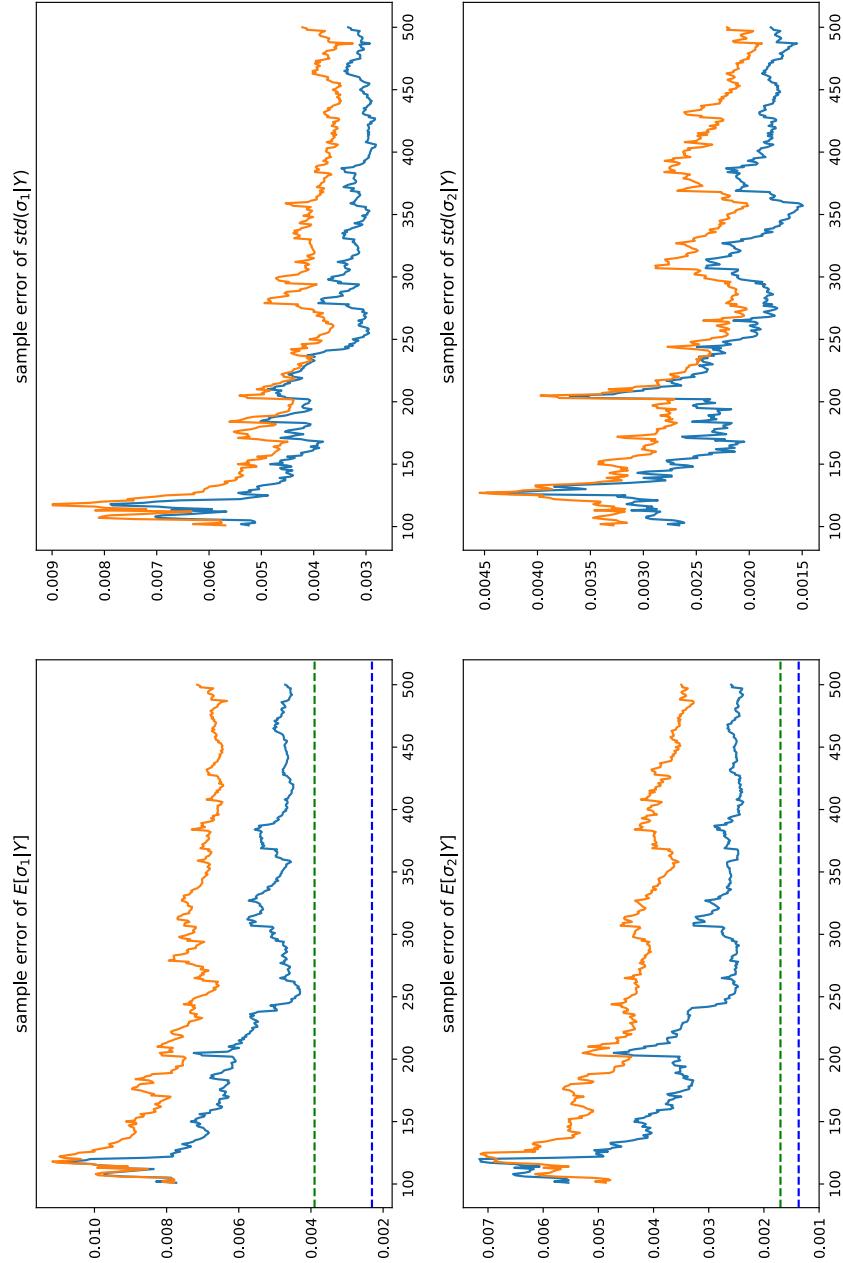


Figure A.6: Online sample error for different setups, where x axis is time t ; y axis is the error. Blue dashed: MCMC-SS/SMC error, green dashed: offline SMC² error, blue: recMCMC-SS/SMC error, orange: SMC² error

A.2 Latent efficient price process

In Section 3.3, we introduce latent efficient prices with marked point process observations and exogenous parameter jumps. In this section, we supplement the details of how we design the proposal distributions of SMC for the state space model of that problem. Next, we apply MCMC-SS/SMC to the same dataset of [25] to compare the estimated model parameters.

A.2.1 Proposal designs

The naive SMC proposal density is to use the transition density (3.10) itself as the proposal:

$$X_{T_i}^j := X_{T_{i-1}}^j \exp \left\{ (\mu - \frac{1}{2}v_i)D_i + N(0, v_i D_i) \right\}, 1 \leq j \leq M,$$

Then we adjust the importance weight as

$$w_i^j = w_{i-1}^j g_\theta(Y_{T_i} | X_{T_i}^j), 1 \leq j \leq M.$$

Unfortunately, this naive proposal density is very inefficient, because no information contained in observation $Y(T_i)$ about latent state $X(T_i)$ is exploited.

First, let us analyze how much information contained in observation prices. Notice equation (3.6) describes the relation between latent states X_{T_i} and Y_{T_i} . The rounding function has a noise with variance around $1/M^2$. The market microstructure noise

has a variance of

$$\text{Var}(\varepsilon_i) = \sum_{k=1}^{\infty} \frac{k^2}{M^2} (1 - \rho) \rho^k = \frac{1}{M^2} \frac{\rho(\rho + 1)}{(1 - \rho)^2}$$

If $\rho = 0.05$, $M = 64$, $\sigma_0 = 7.8 \times 10^{-5}$, $\mathbb{E}(D_i) = 1/0.0058$ seconds as estimated in [25], the variance of rounding noise is around 10^{-4} and $\text{Var}(\varepsilon_i) \approx 10^{-5}$. However, for the variance of latent states,

$$\text{Var}(X_{T_{i+1}}|X_{T_i}) \approx \sigma_0^2 X_{T_i}^2 D_i \approx (7.8 \times 10^{-5})^2 \times 10^4 / 0.0058 \approx 10^{-2}.$$

Therefore, the information contained in observation prices Y_{T_i} is much richer than the information contained in the latent states transitions themselves. It is very important to posteriorly sample next latent state conditioning on the observation price. To utilize observation information, the ideal proposal distribution should be

$$q(X_{T_i}|X_{T_{i-1}}, Y_{T_i}) = \frac{p(X_{T_i}|X_{T_{i-1}})g(Y_{T_i}|X_{T_i})}{\int p(X_{T_i}|X_{T_{i-1}})g(Y_{T_i}|X_{T_i})dY_{T_i}dX_{T_i}}, \quad (\text{A.1})$$

where $p(X_{T_i}|X_{T_{i-1}})$ comes from equation (3.10) and $g(Y_{T_i}|X_{T_i})$ comes from a complicated combination of equation (3.6) and (3.7). This proposal density is too complicated to sample from and we will introduce some approximation to it below. Firstly, notice that the first term $p(X_{T_i}|X_{T_{i-1}})$ has approximation $\tilde{p}(X_{T_i}|X_{T_{i-1}}) = N(X_{T_{i-1}} + X_{T_{i-1}}\mu D_i, X_{T_{i-1}}^2 v_i D_t)$ when $D_i = T_i - T_{i-1}$ is small. In our specific application, the percentage change between two transaction is indeed below 0.1%, so the approximation should be sufficient. Secondly, to approximate the observation density $g(Y_{T_i}|X_{T_i})$, we can encode the likelihood of $X_{T_i} - Y_{T_i}$ as a Gaussian density with

mean zero and variance $\frac{1}{M^2}$, i.e. $\tilde{g}(Y_{T_i}|X_{T_i}) = N(X_{T_i}, 1/M^2)$. With $\tilde{p}(X_{T_i}|X_{T_{i-1}})$ and $\tilde{g}(Y_{T_i}|X_{T_i})$ as approximated densities, we can plug these two Gaussian densities into equation (A.1). Define $\mu_A, \sigma_A, \mu_B, \sigma_B$ as

$$\begin{cases} \mu_A = X_i + \mu X_i \Delta t \\ \sigma_A^2 = X_i^2 \sigma_i^2 \Delta t \end{cases} \quad \begin{cases} \mu_B = Y_{i+1} \\ \sigma_B^2 = \frac{1}{M^2} \end{cases}$$

Then the approximate full posterior density is

$$q(X_{i+1}|X_i, Y_{i+1}) = N\left(\frac{\sigma_B^2}{\sigma_A^2 + \sigma_B^2}\mu_A + \frac{\sigma_A^2}{\sigma_A^2 + \sigma_B^2}\mu_B, \frac{\sigma_A^2 \sigma_B^2}{\sigma_A^2 + \sigma_B^2}\right).$$

We call this proposal density ‘‘approximate full posterior density’’.

Given the two proposal densities (naive proposal, approximate full posterior sampling), we conduct a simulation study to test its efficiencies. Notice that we can estimate the likelihood of posterior density $\pi(Y_{1:T}|\theta)$ from particle filters. We repeat the estimation for 1,000 simulations, and compute the sample error. From the results in Table A.3, we can observe that given the same number of particles, the approximate full posterior proposal have the lowest errors. On the other hand, given the same level of errors, the approximate full posterior proposal costs far less than the naive proposals. Hence, our approximation of equation (A.1) works well. We have used this approximate full posterior proposal throughout the numerical experiments in Sections 3.3 and A.2.2.

#particles	NP error	NP time	AFP error	AFP time
1000	12.76119	0.203651	1.287059	0.235444
2000	10.87919	0.401519	0.954077	0.464879
4000	10.56866	0.793764	0.675782	0.930985
8000	9.378089	1.587471	0.478113	1.869897
16000	8.525059	3.173492	0.352021	3.720968
32000	8.24185	6.349483	0.248371	7.440366
64000	7.553386	12.8127	0.179924	14.93494

Table A.3: Particle filter estimation error (standard error for 1,000 simulations) of likelihood for $T = 1000$, where “NP” is naive proposal, “AFP” is approximate full posterior proposal, “time” is how many seconds it takes to evaluate the likelihood of one atom

A.2.2 Applications in real world dataset

In Section 3.3, we apply recMCMC-SS/SMC to simulated data and show that recMCMC-SS/SMC can estimate model parameters and latent states very well. In this section, we will apply MCMC-SS/SMC to a real dataset originating from Hu et al. [25]. The dataset consists of a complete record of trades of the “on-the-run” (OTR) 10-year U.S. Treasury note during the period 8/15/2000 to 12/31/2000. Summary statistics of the dataset are included in Table A.4.

# trades	Min	Max	Median	Mean	Std	Skewness	Kurtosis
10006	98.6875	105.6563	100.0625	100.3787	1.3777	1.9014	6.3022

Table A.4: Summary statistics for the OTR 10-year note, 8/15/2000 - 12/31/2000

Table A.5 shows the estimation results of MCMC-SS/SMC and [25]. Reader may find some differences between our results and [25]’s results. This is because [25]’s method is numerical and only converges to the true value when the number of grids

goes to infinity. But in reality their numerical methods suffer the curse of dimensionality, leading to inaccurate estimation. One piece of evidence is that their estimation of the annual return of bond is 50.07% with a standard deviation of 7.39%, which obviously deviate from reasonable values. Nevertheless, MCMC-SS/SMC remains sound estimations for all parameters. Figure A.7 shows the empirical distributions from MCMC-SS/SMC as well as the correlations between different model parameters. Interestingly, we find significant negative correlations between σ_0 and σ_2 . This is explainable: bid-initiated trades happen approximately every two trades. A higher σ_2 will increase the overall volatility. To make the overall volatility stay unchanged, the base volatility σ_0 must be lower.

	MCMC-SS/SMC		Approx. Markov	
posterior:	mean	s.e.	mean	s.e.
μ	22.71%	7.97%	50.07%	7.39%
σ_0	5.11%	0.07%	5.28%	0.02%
σ_1	4.62%	0.42%	4.99%	0.60%
σ_2	-0.33%	0.09%	-0.80%	0.38%
ρ	3.50%	0.46%	3.21%	0.13%

Table A.5: Estimation results of MCMC-SS/SMC and [25]’s approximate Markov methods

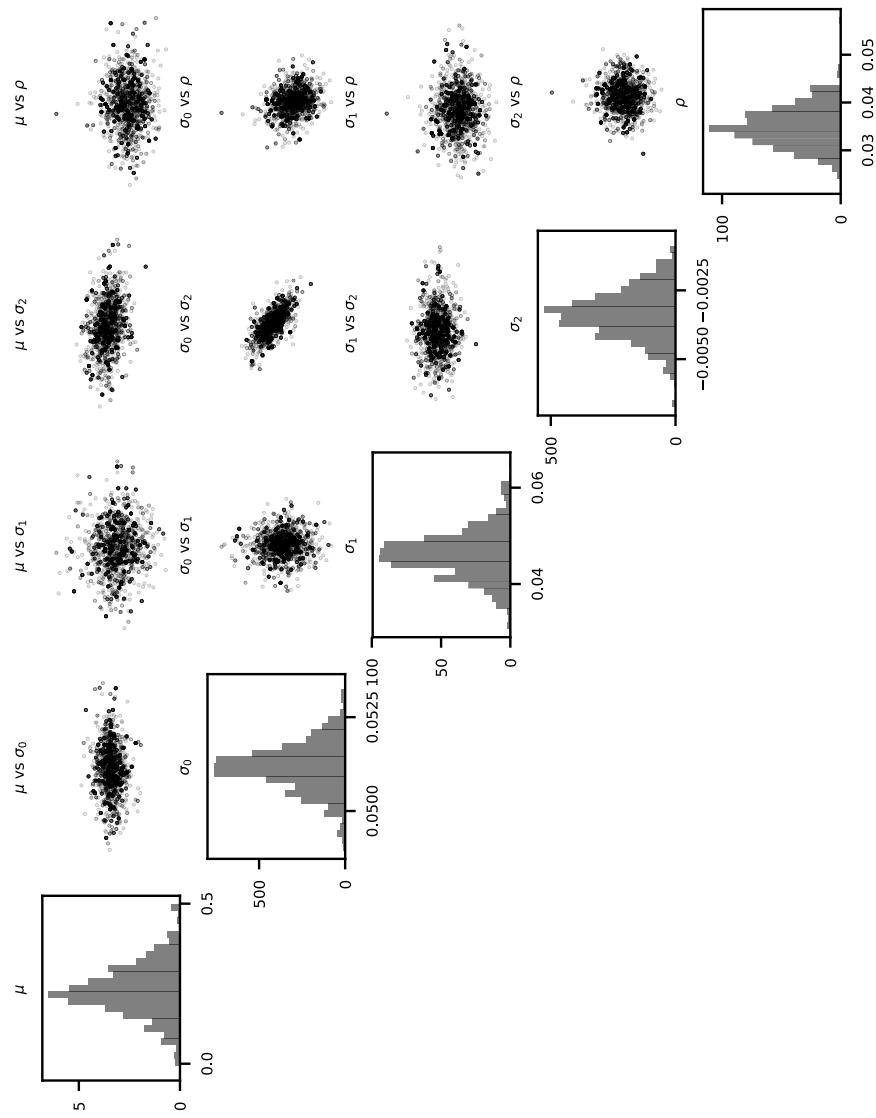


Figure A.7: MCMC-SS/SMC samples visualization for the model parameters of latent efficient prices process

A.3 Stochastic volatility models

In this section, we include more detail on the stochastic volatility models. Section A.3.1 introduces different kinds of stochastic volatility models and explains how we can model them into state space models. Section A.3.2 explains different proposal distributions we design and a simulation study is conducted to compare these proposals. Section A.3.3 conducts the same simulation study as in [20, Appendix A] to compare performance of MCMC-SS/SMC and Gibbs sampling. Section A.3.4 includes more results of filtered model parameters paths for S&P 500 data.

A.3.1 State space modeling

Eraker et al. [20] give a thorough overview of the development of stochastic volatility models that describe financial asset returns. We briefly introduce the four models they investigate. Notice that for each model, we mark the newly introduced variable in blue to emphasize the difference.

1. Stochastic volatility model (SV): This model is the most basic model for stochastic volatility models. Compared to constant volatility Brown motion, SV introduces a stochastic volatility that has a mean-reversion property.

$$d \log(S_t) = \mu dt + \sqrt{V_{t-}} dZ_t^1$$

$$dv_t = \kappa(\theta - v_{t-})dt + \sqrt{v_{t-}}\sigma dZ_t^2$$

where S_t is stock price, v_t is spot volatility, μ is mean return, κ is mean-reverting speed, θ is long-term mean volatility, σ is volatility drift size, Z_t^1 and Z_t^2 are Brownian drifts.

2. Stochastic volatility model with jumps in returns (SVJ): This model includes an additional term, jumps in returns, to explain the multiple market crashes in history.

$$d \log(S_t) = \mu dt + \sqrt{v_{t-}} dZ_t^1 + \xi_t^Y \textcolor{blue}{dJ}_t^Y$$

$$dv_t = \kappa(\theta - v_{t-})dt + \sqrt{v_{t-}} \sigma dZ_t^2$$

where dJ_t^Y is return jump term, and ξ_t^Y is return jump size.

3. Stochastic volatility model with independent jump in returns and volatility (SVIJ): since there are consecutive jumps in returns following trading days after a market crash, a jump in volatility is also introduced in SVIJ model.

$$d \ln(S_t) = \mu dt + \sqrt{v_{t-}} dZ_t^1 + \xi_t^Y dJ_t^Y$$

$$dv_t = \kappa(\theta - v_{t-})dt + \sqrt{v_{t-}} \sigma dZ_t^2 + \xi_t^V \textcolor{blue}{dJ}_t^v$$

where dJ_t^v is volatility jump term and ξ_t^v is volatility jump size.

4. Stochastic volatility model with contemporaneous jumps in returns and volatility (SVCJ): if we assume returns and volatilities jumps happen at the same time, they can be based on a contemporaneous jump event.

$$d \log(S_t) = \mu dt + \sqrt{v_{t-}} dZ_t^1 + \xi_t^Y \textcolor{blue}{dJ}_t$$

$$dv_t = \kappa(\theta - v_{t-})dt + \sqrt{v_{t-}} \sigma dZ_t^2 + \xi_t^v \textcolor{blue}{dJ}_t$$

where dJ_t is to describe jump happens at the same time.

As described in Section 3.3, we focus on modeling the most complicated SVCJ model. The SDE of SVCJ model can be approximated to the following discrete transitions:

$$\begin{aligned}\ln\left(\frac{S_t}{S_{t-1}}\right) &= \mu + \sqrt{V_{t-1}}(\rho Z_t^1 + \sqrt{1-\rho^2}Z_t^2) + J_t \xi_t^Y \\ V_t &= V_{t-1} + \kappa(\theta - V_{t-1}) + \sqrt{V_{t-1}}\sigma Z_t^1 + J_t \xi_t^V\end{aligned}$$

We treat blue variables as hidden states. The likelihoods of red variables are observable. More formally, let the latent states be

$$X_t := (V_{t-1}, Z_t^1, J_t, \xi_t^V),$$

and the observation states be

$$Y_t := \log(S_t/S_{t-1}).$$

Then the transition probability from X_{t-1} to X_t is

$$\begin{aligned}\mathbb{P}(X_t|X_{t-1}) &= \mathbb{P}(V_{t-1}, Z_t^1, J_t, \xi_t^V | V_{t-2}, Z_{t-1}^1, J_{t-1}, \xi_{t-1}^V) \\ &= p(\xi_t^V; \mu_v)p(J_t; \lambda)\phi(Z_t^1)\mathbf{1}_{V_{t-1}=V_{t-2}+\kappa(\theta-V_{t-2})+\sqrt{V_{t-2}}\sigma Z_{t-1}^1+J_{t-1}\xi_{t-1}^V}\end{aligned}$$

where $p(\xi_t^V; \mu_v)$ is the PDF of exponential distribution; $p(J_t; \lambda)$ is the PDF of Bernoulli distribution; $\phi(Z_t^1)$ is the PDF of standard normal distribution.

The observation probability of Y_t conditioning on X_t is

$$\mathbb{P}(Y_t|X_t) = \mathbb{P}(\sqrt{V_{t-1}(1-\rho^2)}Z_t^2 + J_t\xi_t^Y|V_{t-1}, Z_t^1, J_t, \xi_t^V).$$

A.3.2 Proposal designs

After modeling the problem to a HMM, we can design different proposal densities $q(X_{t+1}|X_t, Y_{1:t})$. There are many possible proposals based on how much information are exploited from observed states Y_t . Here, we consider three forms: (1) sampling without observation information; (2) posteriorly sampling jump term; (3) posteriorly sampling all terms.

No posterior This proposal density is the naive particle filter proposal. We choose the transition density as proposal, i.e. $q_t(X_{t+1}|X_t) = p(X_{t+1}|X_t)$. For each component in the hidden state, we can sample independently without using observation information.

$$\begin{aligned} q(X_t|X_{t-1}) &= p(V_{t-1}, Z_t^1, J_t, \xi_t^V|X_{t-1}) \\ &= p(\xi_t^V; \mu_v)p(J_t; \lambda)\phi(Z_t^1)\mathbf{1}_{V_{t-1}=V_{t-2}+\kappa(\theta-V_{t-2})+\sqrt{V_{t-2}}\sigma Z_{t-1}^1+J_{t-1}\xi_{t-1}^V} \end{aligned}$$

Jump-only Since jump is a rare event in the movement of asset prices (i.e. jump rate λ is very small), if we posteriorly sample the jump term, the efficiency of particle filter can be greatly improved. We design the proposal density as the following

$$q(X_t|X_{t-1}, Y_t) = p(Z_t^1, \xi_t^V, V_{t-1}|X_{t-1}) \cdot p(J_t|Z_t^1, \xi_t^V, V_{t-1}, X_{t-1}, Y_t).$$

For the first part, we sample $(Z_t^1, \xi_t^V, V_{t-1})$ without any posterior information:

$$p(Z_t^1, \xi_t^V, V_{t-1} | X_{t-1}) = \mathbf{1}_{V_{t-1}=V_{t-2}+\kappa(\theta-V_{t-2})+\sqrt{V_{t-2}}\sigma Z_{t-1}^1+J_{t-1}\xi_{t-1}^V} \cdot \phi(Z_t^1; 0, 1) \cdot p(\xi_t^V; \mu_v)$$

For the second part, we sample J_t conditioning on $(Z_t^1, \xi_t^V, V_{t-1}, Y_t)$,

$$\begin{aligned} & p(J_t | Z_t^1, \xi_t^V, V_{t-1}, X_{t-1}, Y_t) \\ & \propto p(J_t, Z_t^1, \xi_t^V, Y_t | V_{t-1}) \\ & = \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1 + (\mu_y + \rho_J \xi_t^V) J_t, (1 - \rho^2) V_{t-1} + \sigma_y^2 J_t) p(\xi_t^V; \mu_v) \\ & \quad \cdot p(J_t; \lambda) \phi(Z_t^1; 0, 1) \\ & \propto \begin{cases} (1 - \lambda) \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1, (1 - \rho^2) V_{t-1}) & J_t = 0 \\ \lambda \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1 + (\mu_y + \rho_J \xi_t^V), (1 - \rho^2) V_{t-1} + \sigma_y^2) & J_t = 1 \end{cases} \end{aligned}$$

Full posterior Finally, if we are able to posteriorly sample all terms in hidden state X_t , this proposal density will be the most efficient among all possible proposals. We design the proposal density as the following

$$\begin{aligned} q(X_t | X_{t-1}, Y_t) &= p(V_{t-1} | X_{t-1}, Y_t) \\ &\quad \cdot p(J_t | V_{t-1}, X_{t-1}, Y_t) \\ &\quad \cdot p(\xi_t^V | J_t, V_{t-1}, X_{t-1}, Y_t) \\ &\quad \cdot p(Z_t^1 | \xi_t^V, J_t, V_{t-1}, X_{t-1}, Y_t) \end{aligned}$$

For the first part, we just assign V_{t-1} deterministically

$$p(V_{t-1} | X_{t-1}, Y_t) = \mathbf{1}_{V_{t-1}=V_{t-2}+\kappa(\theta-V_{t-2})+\sqrt{V_{t-2}}\sigma Z_{t-1}^1+J_{t-1}\xi_{t-1}^V} \cdot$$

For the second part,

$$\begin{aligned}
& p(J_t | V_{t-1}, X_{t-1}, Y_t) \\
& \propto \int p(J_t, Z_t^1, \xi_t^V, Y_t | V_{t-1}) dZ_t^1 d\xi_t^V \\
& = \int \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1 + (\mu_y + \rho_J \xi_t^V) J_t, (1 - \rho^2) V_{t-1} + \sigma_y^2 J_t) \\
& \quad \cdot p(\xi_t^V; \mu_v) p(J_t; \lambda) \phi(Z_t^1; 0, 1) dZ_t^1 d\xi_t^V \\
& = \begin{cases} (1 - \lambda) \int \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1, (1 - \rho^2) V_{t-1}) \phi(Z_t^1; 0, 1) dZ_t^1 & J_t = 0 \\ \lambda \int \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1 + \mu_y + \rho_J \xi_t^V, (1 - \rho^2) V_{t-1} + \sigma_y^2) \\ \quad \cdot \phi(Z_t^1; 0, 1) p(\xi_t^V; \mu_v) d\xi_t^V dZ_t^1 & J_t = 1 \end{cases} \\
& = \begin{cases} (1 - \lambda) \phi(Y_t; \mu, V_{t-1}) & J_t = 0 \\ \lambda \int_0^\infty \phi(Y_t; \mu + \mu_y + \rho_J \xi_t^V, V_{t-1} + \sigma_y^2) p(\xi_t^V; \mu_v) d\xi_t^V & J_t = 1 \end{cases} \\
& = \begin{cases} (1 - \lambda) \phi(Y_t; \mu, V_{t-1}) & J_t = 0 \\ \lambda \frac{\sqrt{2\pi}}{\mu_v |\rho_J|} e^{\frac{1}{2} \frac{\mu'^2}{\sigma'^2} - \frac{1}{2} \frac{(\mu + \mu_y - Y_t)^2}{V_{t-1} + \sigma_y^2}} \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma'} e^{-\frac{1}{2} \frac{(\xi_t^V - \mu')^2}{\sigma'^2}} d\xi_t^V & J_t = 1 \end{cases} \\
& = \begin{cases} (1 - \lambda) \phi(Y_t; \mu, V_{t-1}) & J_t = 0 \\ \lambda \sqrt{2\pi} (1 - \Phi(-\frac{\mu'}{\sigma'})) \frac{\sigma'}{\mu_v} e^{\frac{1}{2} \frac{\mu'^2}{\sigma'^2}} \phi(Y_t; \mu + \mu_y, V_{t-1} + \sigma_y^2) & J_t = 1 \end{cases}
\end{aligned}$$

where $\mu' = -\frac{\mu + \mu_y - Y_t}{\rho_J} - \frac{\sigma'^2}{\mu_v}$, $\sigma' = \sqrt{\frac{V + \sigma_y^2}{\rho_J^2}}$.

For the third part, if $J_t = 0$, ξ_t^V is nuisance parameter. If $J_t = 1$,

$$\begin{aligned}
& p(\xi_t^V | J_t = 1, V_{t-1}, X_{t-1}, Y_t) \\
& \propto p(\xi_t^V, J_t = 1 | V_{t-1}, X_{t-1}, Y_t) \\
& = \int p(\xi_t^V, Z_t^1, J_t = 1 | V_{t-1}, X_{t-1}, Y_t) dZ_t^1 \\
& \propto \lambda \int \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1 + \mu_y + \rho_J \xi_t^V, (1 - \rho^2)V_{t-1} + \sigma_y^2) \phi(Z_t^1; 0, 1) p(\xi_t^V; \mu_v) dZ_t^1 \\
& = \lambda \phi(Y_t; \mu + \mu_y + \rho_J \xi_t^V, V_{t-1} + \sigma_y^2) p(\xi_t^V; \mu_v) \\
& \propto \phi(\xi_t^V | \mu', \sigma'^2) \mathbf{1}_{\xi_t^V \geq 0}
\end{aligned}$$

For the forth part,

$$\begin{aligned}
& p(Z_t^1 | J_t = 0, \xi_t^V, V_{t-1}, X_{t-1}, Y_t) \\
& \propto p(Z_t^1, J_t = 0 | V_{t-1}, X_{t-1}, Y_t) \\
& \propto (1 - \lambda) \int \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1, (1 - \rho^2)V_{t-1}) \phi(Z_t^1; 0, 1) dZ_t^1 \\
& \propto \phi(Z_t^1; \rho \frac{(Y_t - \mu)}{\sqrt{V_{t-1}}}, 1 - \rho^2)
\end{aligned}$$

$$\begin{aligned}
& p(Z_t^1 | J_t = 1, \xi_t^V, V_{t-1}, X_{t-1}, Y_t) \\
& \propto (1 - \lambda) \phi(Y_t; \mu + \rho \sqrt{V_{t-1}} Z_t^1, (1 - \rho^2)V_{t-1}) \phi(Z_t^1; 0, 1) \\
& \propto \phi(Z_t^1; \frac{\rho \sqrt{V_{t-1}}(Y_t - \mu - \mu_y - \rho_J \xi_t^V)}{V_{t-1} + \sigma_y^2}, \frac{(1 - \rho^2)V + \sigma_y^2}{V + \sigma_y^2})
\end{aligned}$$

Given the three designs of proposal densities (naive proposal, jump-only proposal, full posterior proposal), we conduct the same simulation study as in Table A.3 to test its efficiencies. From the results in Table A.6, we can observe that given the same

number of particles, the full posterior proposal has the smallest errors. On the other hand, given the same level of errors, the full posterior proposal costs far less than the other proposals. We will use full posterior proposal throughout the numerical experiments in Sections 3.4, A.3.3 and A.3.4.

# particles	NP error	NP time	JOP error	JOP time	FP error	FP time
100	18.88	0.37	6.10	0.32	1.06	0.21
200	12.49	0.39	4.08	0.37	0.76	0.36
400	8.52	0.48	2.88	0.62	0.53	0.68
800	5.31	0.92	2.12	1.22	0.39	1.32
1600	3.38	1.78	1.41	2.41	0.27	2.60
3200	2.38	3.54	1.00	4.84	0.19	5.20
6400	1.59	7.13	0.72	9.76	0.14	10.49

Table A.6: Particle filter estimation error (standard error for 1000 simulations) of likelihood for $T = 5000$ SVCJ model, where “NP” is naive proposal, “JOP” is jump-only proposal, “FP” is full posterior proposal. “time” is how many seconds it takes to evaluate the likelihood of one atom

A.3.3 Comparison studies with Eraker et al. [20]

Eraker et al. [20] propose a simulation study to verify the effectiveness of their Gibbs sampling algorithm. They choose model parameters of mean return $\mu = 0.05$, mean-reversion speed $\kappa = 0.03$, mean long-term volatility $\theta = 0.50$, the volatility of volatility $\sigma = 0.10$, correlation between return and volatility $\rho = -0.50$, daily jump probability $\lambda = 0.008$, return jump mean $\mu_y = -0.200$, return jump size standard deviation $\sigma_y = 3.500$, volatility jump mean $\mu_v = 1.000$, return and volatility jump

correlation $\rho_J = -0.400$. We adopt the same true model parameters and repeat the same simulation experiments as in [20] but with MCMC-SS/SMC algorithm instead of Gibbs sampling. The steps are as follows:

1. Generate 100 realizations of simulated data $(Y_{1:T}^1, Y_{1:T}^2, \dots, Y_{1:T}^{100})$ with sample size $T = 4000$
2. Run MCMC-SS/SMC for each realization $Y_{1:T}^i$ to get estimates $\hat{\theta}^i = (\hat{\theta}_1^i, \dots, \hat{\theta}_D^i)$, and calculate mean $\hat{\theta}_d = \frac{1}{100} \sum_{i=1}^{100} \hat{\theta}_d^i$ and RMSE $\hat{\sigma}(\hat{\theta}_d) = \sqrt{\frac{1}{100} \sum_{i=1}^{100} (\hat{\theta}_d^i - \hat{\theta}_d)^2}$
3. Define t-value as $\frac{\hat{\theta}_d - \theta_d}{\hat{\sigma}(\hat{\theta}_d)/10}$

We compare our estimation results and RMSE with those reported in [20]. As shown in Table A.7, several t -values from [20] are extremely abnormal. We suspect incorrect estimation of their Gibbs sampling algorithm. However, MCMC-SS/SMC maintains a reasonable t -value range for all estimated values.

θ	true	Eraker	t-value	MCMC-SS/SMC	t-value
μ	0.050	0.0507(0.00114)	0.61	0.0519(0.00116)	-1.68
κ	0.030	0.0330(0.00062)	4.84	0.0314(0.00050)	2.76
θ	0.500	0.4755(0.00601)	-4.07	0.5108(0.00618)	1.75
σ	0.100	0.0817(0.00076)	-24.08	0.1022(0.00116)	1.93
ρ	-0.500	-0.4921(0.00711)	1.11	-0.4799(0.00762)	2.64
λ	0.008	0.0093(0.00027)	4.81	0.0080(0.00024)	-0.06
μ_y	-2.000	-2.3140(0.11087)	-2.83	-2.7334(0.19037)	-3.85
σ_y	3.500	2.9116(0.04259)	13.81	3.4900(0.06799)	-0.15
μ_v	1.000	1.4272(0.03622)	11.79	1.1961(0.04513)	4.35
ρ_J	-0.400	-0.0599(0.04369)	7.78	-0.1526(0.13921)	1.21

Table A.7: Eraker et al. reported 100 simulation results and MCMC-SS/SMC 100 simulation results, RMSE is divided by 10 in parentheses

Eraker et al. [20] then conduct an empirical study on S&P 500 returns data from 01/02/1980 to 12/31/1999. We apply MCMC-SS/SMC to the same dataset. The estimated results are summarized in Table A.8. Most of the parameters have similar estimation results. However, the estimations of σ_y , μ_v , ρ_J are very different from [20]'s results. These parameters are also the ones for which [20] has an abnormal t -value in the simulation study. Figure A.8 shows the empirical distributions and correlations between model parameters. Our results show that the correlation between jump in return and volatility is significant.

variable	Our Estimation	Our Error	Eraker Estimate	Eraker Error
μ	0.0520	0.0089	0.0554	0.0112
κ	0.0223	0.0036	0.0260	0.0041
θ	0.6925	0.0564	0.5376	0.0539
σ	0.1035	0.0089	0.0790	0.0074
ρ	-0.4627	0.0468	-0.6008	0.0623
λ	0.0044	0.0015	0.0066	0.0020
μ_y	-1.3648	1.0726	-1.7533	1.5566
σ_y	1.7668	1.1565	2.8864	0.5679
μ_v	0.9989	0.3689	1.4832	0.3404
ρ_J	-3.5883	1.0908	-0.6008	0.9918

Table A.8: SVCJ parameter for S&P 500 estimated from different approach

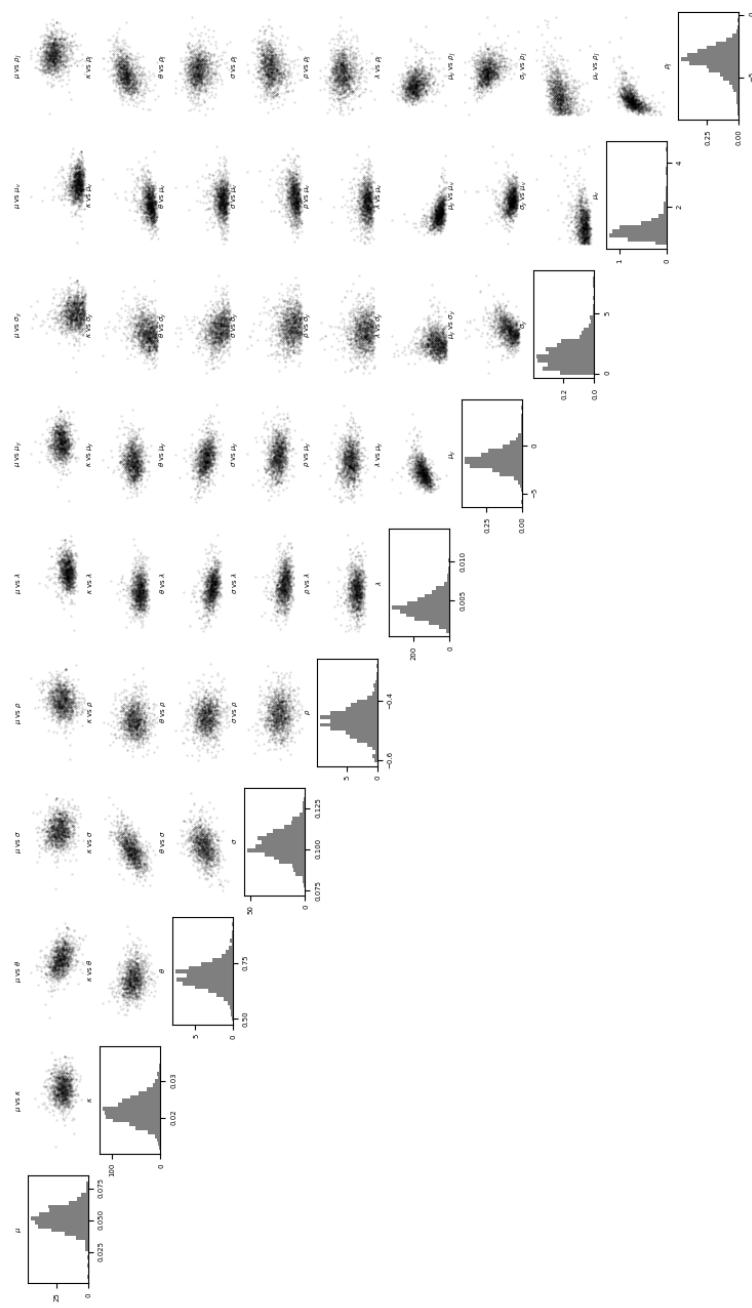


Figure A.8: MCMC-SS/SMC MCMC samples visualization

A.3.4 More filtering results for S&P 500

In Section 3.3, we present the filtered volatilities for S&P 500 from 1996 to 2018. In this section, we supplement posterior estimation paths for all model parameters.

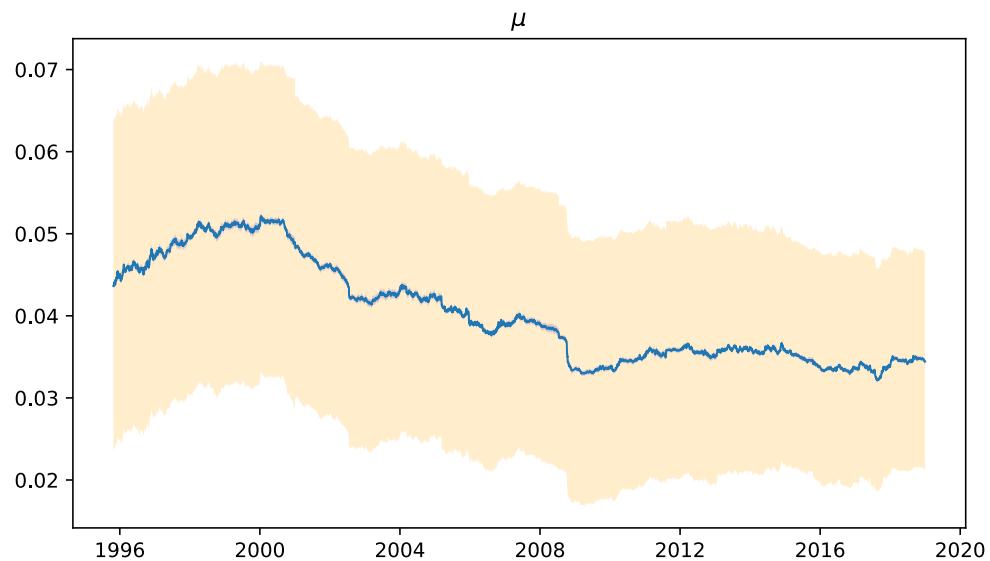
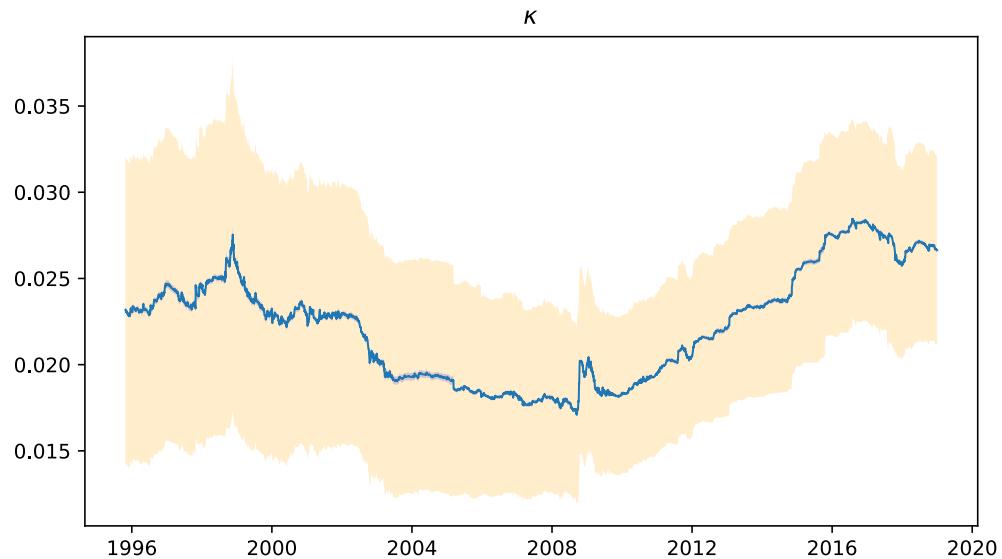
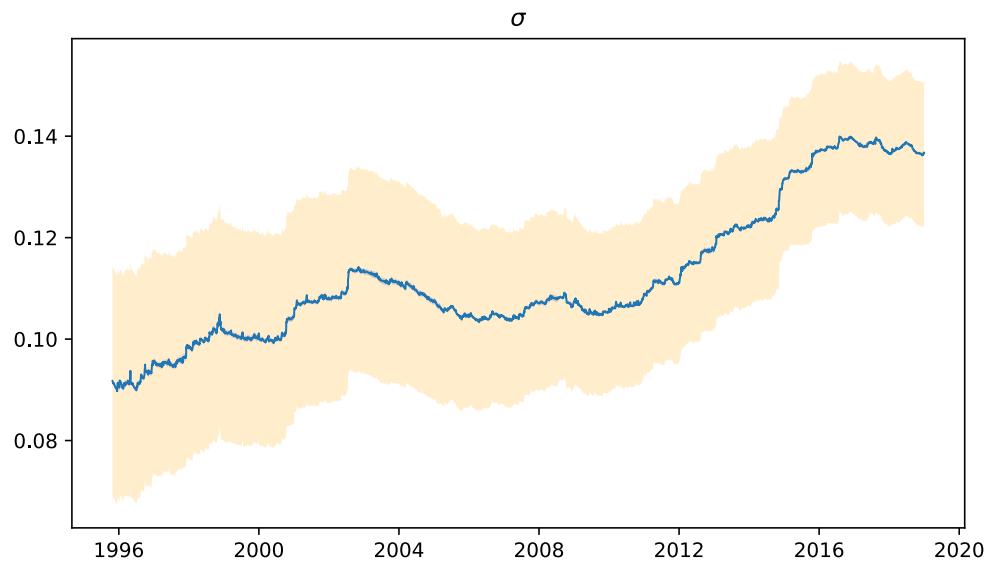
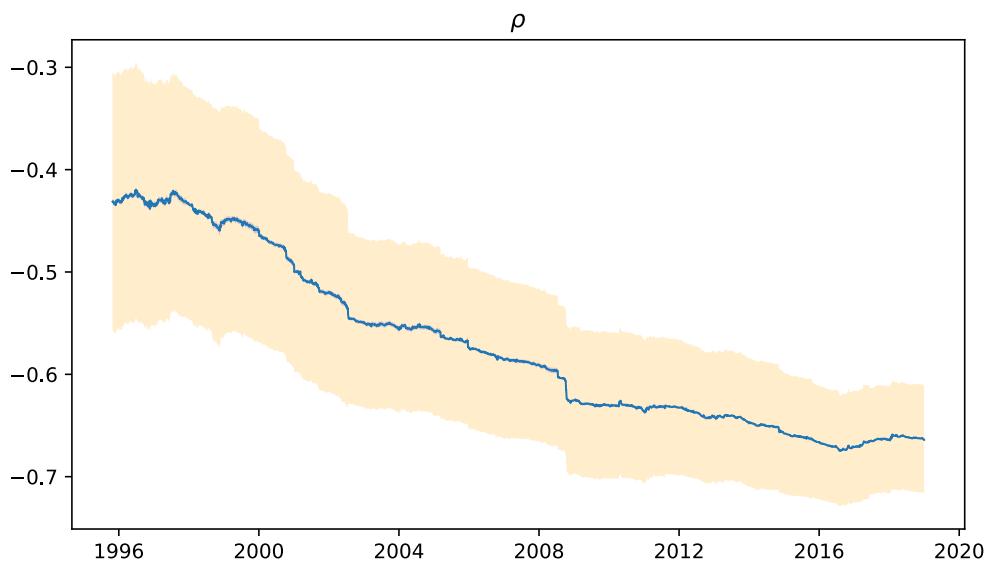
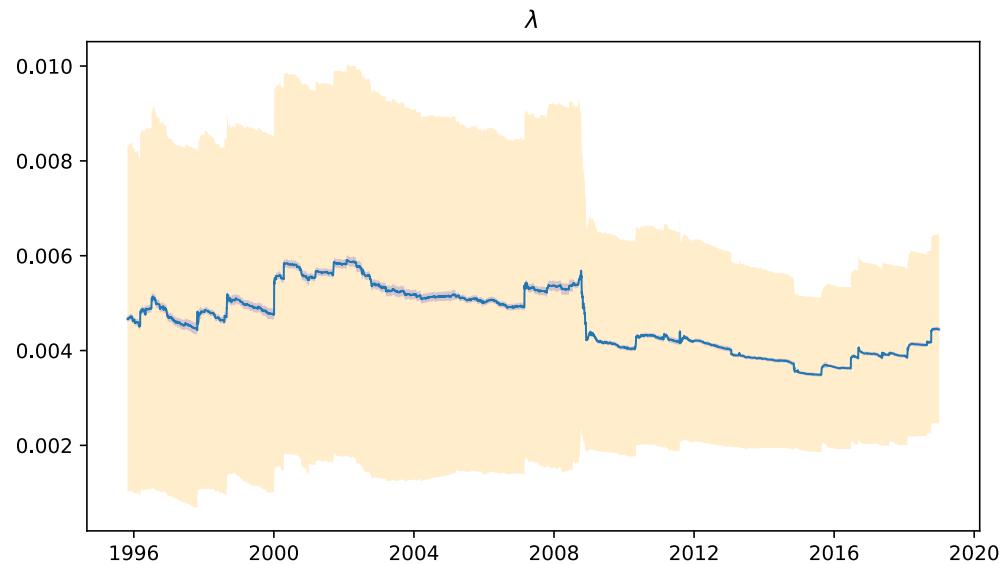
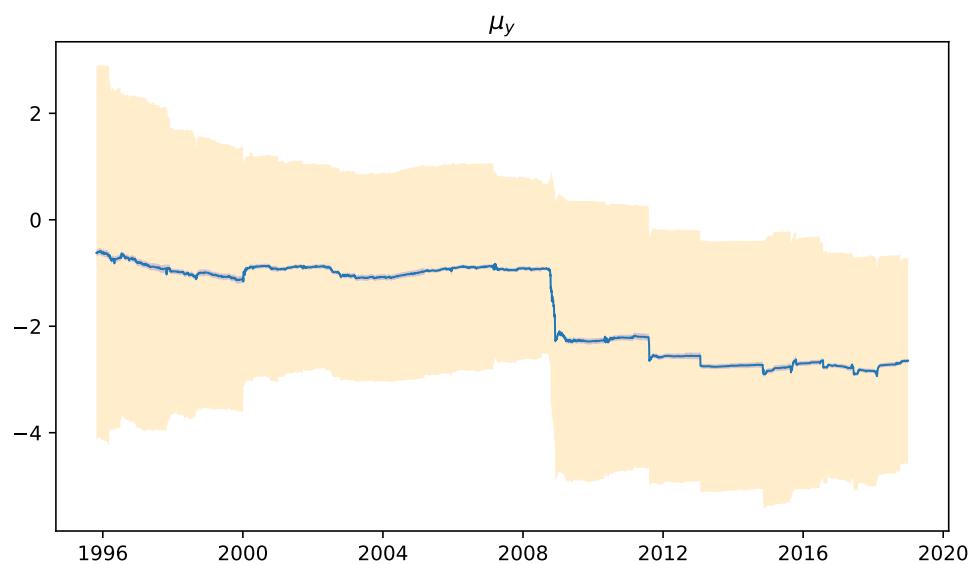


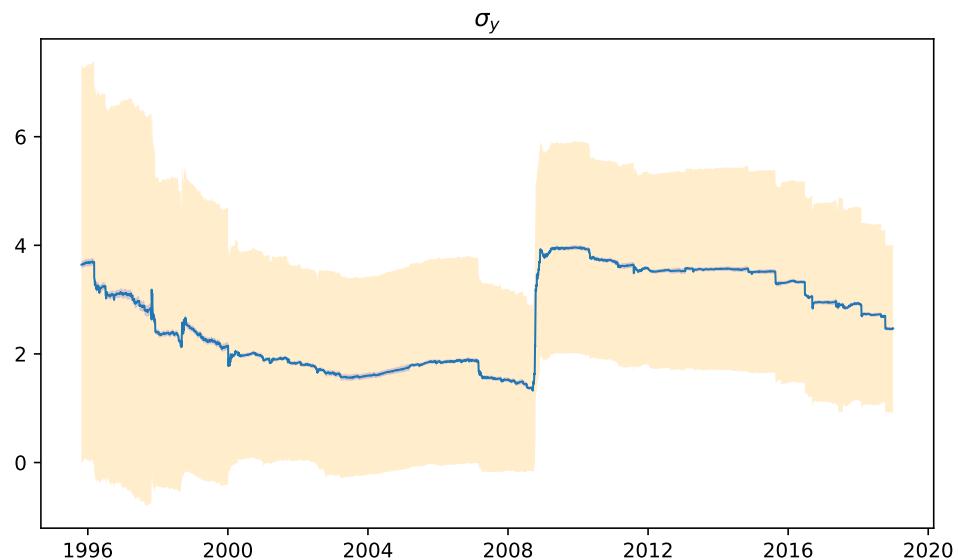
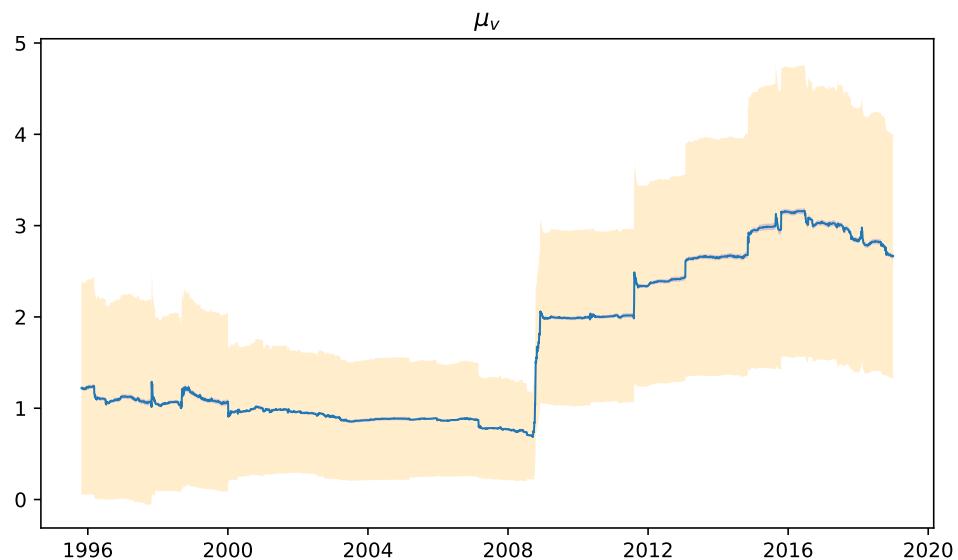
Figure A.9: Daily mean return $\mathbb{E}[\mu|Y_{1:t}]$

Figure A.10: Mean-reversion speed $\mathbb{E}[\kappa|Y_{1:t}]$ Figure A.11: Long-term volatility mean $\mathbb{E}[\theta|Y_{0:t}]$

Figure A.12: Variance for volatility $\mathbb{E}[\sigma|Y_{1:t}]$ Figure A.13: Correlation between volatility and return $\mathbb{E}[\rho|Y_{1:t}]$

Figure A.14: Jump probability $\mathbb{E}[\lambda|Y_{1:t}]$ Figure A.15: Mean jump size for return $\mathbb{E}[\mu_y|Y_{1:t}]$

Appendix 2: Filtering for parameters of SVCJ on S&P 500

Figure A.16: Mean jump size standard deviation $\mathbb{E}[\sigma_y|Y_{1:t}]$ Figure A.17: Standard deviation for jump size $\mathbb{E}[\mu_v|Y_{1:t}]$

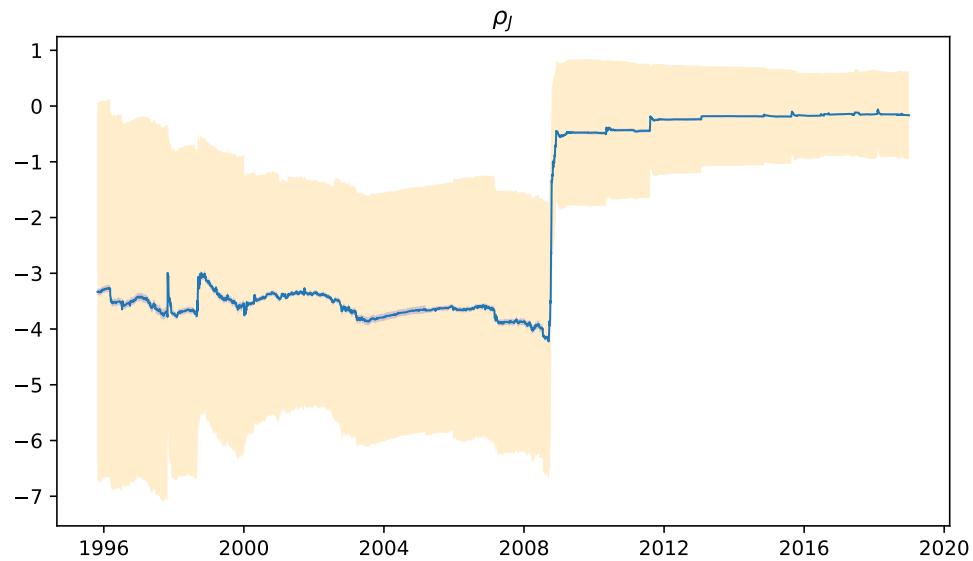


Figure A.18: Coefficient for jump return on jump volatility $\mathbb{E}[\rho_J|Y_{1:t}]$

Bibliography

- [1] Daniel Alspach and Harold Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448, 1972.
- [2] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [3] James E Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21, 1987.
- [4] James Edward Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their applications*, pages 101–111. Hillsdale, New Jersey, 1985.
- [5] Anatoliy Belaygorod and Atilio Zardetto. Practical specification of affine jump-diffusion stochastic volatility models. 2013.

- [6] Anthony Brockwell, Pierre Del Moral, Arnaud Doucet, et al. Sequentially interacting Markov chain Monte Carlo methods. *The Annals of Statistics*, 38(6):3387–3411, 2010.
- [7] Bruce M Brown et al. Martingale central limit theorems. *The Annals of Mathematical Statistics*, 42(1):59–66, 1971.
- [8] Richard S Bucy and Kenneth D Senne. Digital synthesis of non-linear filters. *Automatica*, 7(3):287–298, 1971.
- [9] Hock Peng Chan, Tze Leung Lai, et al. A general theory of particle filters in hidden Markov models and some applications. *The Annals of Statistics*, 41(6):2877–2904, 2013.
- [10] Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.
- [11] Nicolas Chopin, Pierre E Jacob, and Omiros Papaspiliopoulos. SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.
- [12] Nicolas Chopin, Sumeetpal S Singh, et al. On particle gibbs sampling. *Bernoulli*, 21(3):1855–1883, 2015.
- [13] Dan Crisan, Pierre Del Moral, and Terry Lyons. *Discrete filtering using branching and interacting particle systems*. Universite de Toulouse. Laboratoire de Statistique et Probabilites [LSP], 1998.
- [14] Rui JP de Figueiredo and YG Jan. Spline filters. In *Proc. 2nd symp. on nonlinear estimation theory and its applications*, pages 127–138, 1971.

- [15] Randal Douc and Eric Moulines. Limit theorems for weighted samples with applications to sequential Monte Carlo methods. In *ESAIM: Proceedings*, volume 19, pages 101–107. EDP Sciences, 2007.
- [16] Randal Douc, Eric Moulines, Jimmy Olsson, Ramon Van Handel, et al. Consistency of the maximum likelihood estimator for general hidden markov models. *the Annals of Statistics*, 39(1):474–513, 2011.
- [17] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.
- [18] Darrell Duffie, Jun Pan, and Kenneth Singleton. Transform analysis and asset pricing for affine jump-diffusions. *Econometrica*, 68(6):1343–1376, 2000.
- [19] Robert F Engle. The econometrics of ultra-high-frequency data. *Econometrica*, 68(1):1–22, 2000.
- [20] Bjørn Eraker, Michael Johannes, and Nicholas Polson. The impact of jumps in volatility and returns. *The Journal of Finance*, 58(3):1269–1300, 2003.
- [21] Walter R Gilks and Carlo Berzuini. Following a moving target – Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.
- [22] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.

- [23] Jim E Griffin and Stephen G Walker. On adaptive Metropolis-Hastings methods. *Statistics and Computing*, 23(1):123–134, 2013.
- [24] Xin Guo, Howard Shek, Tze Leung Lai, and Samuel Po-Shing Wong. *Quantitative trading: algorithms, analytics, data, models, optimization*. Chapman and Hall/CRC, 2017.
- [25] Grace X Hu, David R Kuipers, and Yong Zeng. Bayesian inference via filtering equations for ultrahigh frequency data (i): Model and estimation. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):34–60, 2018.
- [26] Grace X Hu, David R Kuipers, and Yong Zeng. Bayesian inference via filtering equations for ultrahigh frequency data (ii): Model selection. *SIAM/ASA Journal on Uncertainty Quantification*, 6(1):61–86, 2018.
- [27] Michael Johannes and Nicholas Polson. Mcmc methods for continuous-time financial econometrics. In *Handbook of Financial Econometrics: Applications*, pages 1–72. Elsevier, 2002.
- [28] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.
- [29] Nikolas Kantas, Arnaud Doucet, Sumeetpal S Singh, Jan Maciejowski, Nicolas Chopin, et al. On particle methods for parameter estimation in state space models. *Statistical science*, 30(3):328–351, 2015.

- [30] Genshiro Kitagawa. Non-gaussian statespace modeling of nonstationary time series. *Journal of the American statistical association*, 82(400):1032–1041, 1987.
- [31] Genshiro Kitagawa. A self-organizing state space model. *Journal of the American Statistical Association*, pages 1203–1215, 1998.
- [32] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and Bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.
- [33] Hans R Kunsch et al. Recursive Monte Carlo filters: algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983–2021, 2005.
- [34] Tze Leung Lai, Michael Zhu, and Hock Peng Chan. MCMC with sequential substitutions: theory and applications. *working paper*, 2019.
- [35] Fredrik Lindsten. An efficient stochastic approximation em algorithm using conditional particle filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6274–6278. IEEE, 2013.
- [36] Fredrik Lindsten, Michael I Jordan, and Thomas B Schön. Particle gibbs with ancestor sampling. *The Journal of Machine Learning Research*, 15(1):2145–2184, 2014.
- [37] Fredrik Lindsten and Thomas B Schön. On the use of backward simulation in the particle gibbs sampler. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3845–3848. IEEE, 2012.
- [38] Jane Liu and Mike West. Combined parameter and state estimation in

- simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer, 2001.
- [39] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.
- [40] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.
- [41] Bruce A McElhoe. An assessment of the navigation and course corrections for a manned flyby of mars or venus. *IEEE Transactions on Aerospace and Electronic Systems*, (4):613–623, 1966.
- [42] PD Moral. Feynman-kac formulae: Genealogical and interacting particle systems with applications,Probability and its applications, 2004.
- [43] ML Andrade Netto, L Gimeno, and MJ Mendes. A new spline algorithm for non-linear filtering of discrete time systems. *IFAC Proceedings Volumes*, 11(1):2123–2130, 1978.
- [44] Geir Storvik. Particle filters for state space models with the presence of unknown static parameters. *IEEE Transactions on signal Processing*, 50(2):281–289, 2002.
- [45] Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for non-linear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pages 153–158. Ieee, 2000.
- [46] Yi-Ching Yao et al. Estimation of a noisy discrete-time step function: Bayes and empirical bayes approaches. *The Annals of Statistics*, 12(4):1434–1447, 1984.

- [47] Yong Zeng. A partially observed model for micromovement of asset prices with bayes estimation via filtering. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, 13(3):411–444, 2003.