Center for
Quantitative
Economics

Semiparametric Bayesian Forecasting with an Application to Stochastic Volatility

Fabian Goessling[†] and Martina Danielova Zaharieva[†]

64/2017

First Version: 28.06.2017
This Version: 09.02.2018

[†] Department of Economics, University of Münster, Germany

wissen.leben
WWU Münster

Semiparametric Bayesian Forecasting with an Application to Stochastic Volatility

Fabian Goessling[a], Martina Danielova Zaharieva[a]

[a]*Department of Economics, Am Stadtgraben 9, University of Münster, 48143 Münster, Germany*

Abstract

We propose a new and highly flexible Bayesian sampling algorithm for nonlinear state space models under nonparametric distributions. The estimation framework combines a particle filtering and smoothing algorithm for the latent process with a Dirichlet process mixture model for the error term of the observable variables. In particular, we overcome the problem of constraining the models by transformations or the need for conjugate distributions. We use the Chinese restaurant representation of the Dirichlet process mixture, which allows for a parsimonious and generally applicable sampling algorithm. Thus, our estimation algorithm combines a pseudo marginal Metropolis Hastings scheme with a marginalized hierarchical semiparametric model. We test our approach for several nested model specifications using simulated data and provide density forecasts. Furthermore, we carry out a real data example using S&P 500 returns.

Keywords: Bayesian Nonparametrics, Particle Filtering, Stochastic Volatility, MCMC, Forecasting

1. Introduction

Time-varying volatility is a well known stylized fact of financial returns and thus not only its modeling, but especially its estimation and prediction, are of main interest for practitioners and researchers. In particular, Stochastic Volatility (SV) models are widely popular, even though direct estimation by classical Maximum-Likelihood is often infeasible. Nevertheless, Markov Chain Monte Carlo (MCMC) methods, in combination with a sampling algorithm for the latent volatility, as proposed by Jacquier et al. (2004) or Kim et al. (1998), provide a straightforward solution.[1] More recently, Jensen and Maheu (2010) added a further degree of freedom to SV models by augmenting the models with nonparametric distributions based on a Dirichlet process mixture (DPM). Thus, in addition to the stochastic latent volatility, the error term distribution

Email addresses: Fabian.Goessling@uni-muenster.de (Fabian Goessling), Martina.Zaharieva@uni-muenster.de (Martina Danielova Zaharieva)

[1] See Broto and Ruiz (2004) for a survey.

is highly flexible, which allows, in combination with a Bayesian estimation approach, to learn about the type of distribution from the data. Quite naturally, this enormous flexibility comes at the cost of high complexity as the resulting distributions are possibly non-standard.

The main literature in the field, such as Jensen and Maheu (2010), Delatola and Griffin (2011), Jensen and Maheu (2014), Delatola and Griffin (2013) or Virbickaitė et al. (2014), circumvents this challenge by restricting the model to conjugate distributions and/or transformations of the model equations, but does not offer a generalized solution. Thus, the intended flexibility of a nonparametric model with nonlinear effects of stochastic volatility is constrained by analytical feasibility. We argue that this strongly contradicts the motivation of nonparametric/nonlinear models. We suggest a new, more general estimation algorithm without artificially pruning the model's dynamics and flexibility.

The point of departure for the present paper is the state-space representation of the (semiparametric) SV model. As such, an SV model is comparable, for example, to a nonlinear dynamic stochastic general equilibrium (DSGE) model.[2] For the latter, non-conjugacy and non-standard distributions are widely accepted, and estimation is usually conducted by means of the Metropolis-Hastings (MH) algorithm and particle filter approximations of the likelihood (Fernández-Villaverde and Rubio-Ramírez, 2005). We adopt the same approach and develop our sampling algorithm on an abstract level, using generic distributions without requiring specific distributional assumptions. This allows us to present a modular sampling algorithm which nests semiparametric SV, DSGE, classical SV or even simpler models. Moreover, our presentation is straightforward and strips off the aura of mystery which sometimes surrounds Bayesian nonparametric models. In particular, we use the Chinese Restaurant Process (CRP) representation of the DPM, which enables an attractive visual representation of the sampling steps.

In our simulation exercises, we show that the new algorithm is highly flexible, reliable and straightforward to apply for several nested model specifications. We also provide a real data example using the semiparametric stochastic volatility model of Jensen and Maheu (2010) for S&P 500 data. Furthermore, we demonstrate that our algorithm provides an intuitive way of constructing density forecasts, based on the posterior distributions.

The remainder of the paper is as follows. Section 2 introduces the general setting and preliminary concepts, Section 3 presents the sampling algorithm and Section 4 provides an application to the semiparametric stochastic volatility model using simulated and real data. Section 5 concludes.

[2] See Flury and Shephard (2011) for an estimation approach to both model types.

2. General Setting

2.1. Nonlinear State-Space Model

In what follows, we consider an observable variable

$$y_t = g(s_t, \boldsymbol{\theta}, \epsilon_t), \qquad \epsilon_t \overset{iid}{\sim} \mathcal{G}, \tag{1}$$

where the latent state variable s_t follows the transition equation

$$s_t = f(s_{t-1}, \boldsymbol{\theta}, \eta_t), \qquad \eta_t \overset{iid}{\sim} \mathcal{F}. \tag{2}$$

Furthermore, $g(\cdot)$ and $f(\cdot)$ are potentially nonlinear functions, $\boldsymbol{\theta}$ is a parameter vector and $\mathcal{G}$ and $\mathcal{F}$ are continuous random distributions. For parsimony, we work on one-dimensional y_t and s_t, but the above representation applies to multivariate variables as well. Note that a parametric assumption on $\mathcal{G}$ and $\mathcal{F}$ yields the DSGE model case, and a nonparametric assumption on $\mathcal{G}$ yields the semiparametric SV model case, on which we focus.

2.2. Dirichlet Process Mixture

The DPM represents the distribution of a random variable x_t as an infinite mixture of continuous distributions, for which the mixture component parameters come from a discrete distribution G. In turn, G is constructed from the Dirichlet process prior $\mathrm{DP}(\alpha, G_0)$ (Ferguson (1973)), where α is the *concentration parameter* and G_0 the *base distribution* of the mixture component parameters $\tilde{\mu}_t$ and $\tilde{\sigma}_t$, which we parameterize as a Normal $\mathcal{N}(\cdot)$ and Gamma distribution $\Gamma(\cdot)$, respectively. Throughout the paper, we use mixtures of normals, such that the component parameters are the expected value $\tilde{\mu}_t$ and the standard deviation $\tilde{\sigma}_t$. Following the literature, the hierarchical representation is

$$x_t|(\tilde{\mu}_t, \tilde{\sigma}_t^2) \sim \mathcal{N}(\tilde{\mu}_t, \tilde{\sigma}_t^2), \tag{3}$$

$$(\tilde{\mu}_t, \tilde{\sigma}_t^2)|G \overset{iid}{\sim} G, \tag{4}$$

$$G|G_0, \alpha \sim \mathrm{DP}(G_0, \alpha), \tag{5}$$

$$G_0(\tilde{\mu}_t, \tilde{\sigma}_t^2) = \mathcal{N}(m_0, v_0^2) \times \Gamma(a_0, b_0), \tag{6}$$

where a_0, b_0, m_0 and v_0 are hyperparameters.

2.3. Chinese Restaurant Process

Our estimation algorithm is based on the CRP representation of the DPM, which represents the mixture components as tables in a restaurant, the component parameters as the location inside the restaurant and observations y_t with $t = 1, \ldots, T$ as customers entering the restaurant.[3] Before introducing the CRP in more detail, we clarify the notation in order to avoid ambiguity.

[3] Alternative representations are the stick-breaking representation (Sethuraman, 1994) or the closely related Pólya urn scheme (Blackwell and MacQueen, 1973). An overview is available in Teh (2011).

Let z_t be a label denoting which of the $k \in \{1, 2, \ldots, \infty\}$ tables (components) a customer (observation) y_t is assigned to and $z_t = k$ if customer y_t sits at table k. Furthermore, let c_k be the number of customers sitting at table k in the restaurant and define the *nonparametric* set $\boldsymbol{\phi}_k = \{\mu_k, \sigma_k\}$, which contains the parameters of component k. Thus, we have $\tilde{\mu}_t = \mu_{z_t}$ and $\tilde{\sigma}_t = \sigma_{z_t}$. Given this notation, the CRP can be summarized in two simple steps:

1. For $t = 1$:
   The first customer y_1 sits at the first table with probability 1. Thus we have $z_1 = 1$. The parameters of the first component, indexed by $k = 1$, are sampled from the base distribution, i.e. $\boldsymbol{\phi}_1 \sim G_0$.

2. For $t = 2, \ldots, T$:
   The t-th customer sits at any of the occupied tables $k \in \{1, \ldots, n\}$ with probability $\propto c_k$ or at a non-occupied table with probability $\propto \alpha$. Whenever a new table is chosen, indexed by $n + 1$, sample $\boldsymbol{\phi}_{n+1} \sim G_0$ and increment n by one. In particular, it holds that

$$\begin{aligned} P(z_{t+1} = k|\mathbf{z}_{1:t}, \alpha) &= \frac{c_k}{t + \alpha}, \\ P(z_{t+1} = n + 1|\mathbf{z}_{1:t}, \alpha) &= \frac{\alpha}{t + \alpha}, \end{aligned} \tag{7}$$

   where $\mathbf{z}_{1:t} = \{z_1, \ldots, z_t\}$.

Note that the number of possible tables is unrestricted and the corresponding discrete density is

$$p(z_{t+1}|\mathbf{z}_{1:t}, \alpha) = \frac{c_k}{t + \alpha}\delta(z_{t+1} = k) + \frac{\alpha}{t + \alpha}\delta(z_{t+1} = n + 1),$$

where $\delta(\cdot)$ is the Dirac delta function [4]. Therefore, the model capacity in terms of the parameter space is infinite. Nevertheless, the number of occupied tables is constrained by $n \leq T$. We refer to n as the number of *active* tables or *non-neglectable* components. Note that the process outlined above exhibits the typical *rich-gets-richer* property, i.e. clustering of the customers. Furthermore, as the probability of creating a new table is proportional to α, a small (large) value of α leads to fewer (more) non-empty components. Thus, the value of the concentration parameter α is of major importance. For that reason, our estimation approach additionally imposes a hyperprior on α, in order to achieve higher flexibility. The likelihood of the indicators $p(\mathbf{z}_{1:T}|\alpha)$ can be

[4] Here, the Dirac delta function is used as an indicator function, i.e. given the statement A, we define $\delta(A) = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{else.} \end{cases}$

decomposed as

$$p(\mathbf{z}_{1:T}|\alpha) = p(z_T|\mathbf{z}_{1:T-1},\alpha)p(z_{T-1}|\mathbf{z}_{1:T-2},\alpha)\dots p(z_2|z_1,\alpha)$$
$$= \prod_{i=1}^{T-1} p(z_{T+1-i}|\mathbf{z}_{1:T-i},\alpha). \tag{8}$$

Essentially, using the CRP, we study the marginalized hierarchical semiparametric model

$$x_t|(\tilde{\mu}_t,\tilde{\sigma}_t^2) \sim \mathcal{N}(\tilde{\mu}_t,\tilde{\sigma}_t^2),$$
$$(\{\tilde{\mu}_1,\tilde{\sigma}_1^2\},\dots,\{\tilde{\mu}_T,\tilde{\sigma}_T^2\})|(G_0,\alpha) \sim \mathcal{P}(\{\tilde{\mu}_1,\tilde{\sigma}_1^2\},\dots,\{\tilde{\mu}_T,\tilde{\sigma}_T^2\}|G_0,\alpha),$$
$$G_0(\tilde{\mu}_t,\tilde{\sigma}_t^2) = \mathcal{N}(m_0,v_0^2)\times\Gamma(a_0,b_0),$$

where $\mathcal{P}(\{\tilde{\mu}_1,\tilde{\sigma}_1^2\},\dots,\{\tilde{\mu}_T,\tilde{\sigma}_T^2\}|G_0,\alpha)$ is the joint distribution of the component parameters constructed by the CRP. Therefore, recall that the CRP with indicators $\mathbf{z}_{1:T}$ and component parameters $\boldsymbol{\phi}_{1:n}$ straightforwardly implies the conditional distribution $\mathcal{P}(\{\tilde{\mu}_T,\tilde{\sigma}_T^2\}|\{\tilde{\mu}_1,\tilde{\sigma}_1^2\},\dots,\{\tilde{\mu}_{T-1},\tilde{\sigma}_{T-1}^2\},G_0,\alpha)$ in closed form. Due to the exchangeability of $\{\tilde{\mu}_t,\tilde{\sigma}_t^2\}$ [5], this enables us to calculate conditional distributions for all other t as well, and is thus the impetus for the Gibbs sampling approach, which we apply in Section 3.1.1.

3. Bayesian Inference

Let $\mathbf{z}_{-t}$ denote the set of table assignments $\mathbf{z}_{1:T} = \{z_1,z_2,\dots,z_T\}$ without assignment z_t, i.e. $\mathbf{z}_{-t} = \{z_1,\dots,z_{t-1},z_{t+1},\dots,z_T\}$, analogously $\boldsymbol{\phi}_{1:n,-k}$ the set of component parameters $\boldsymbol{\phi}_{1:n}$ except for ϕ_k, i.e. $\boldsymbol{\phi}_{1:n,-k} = \{\phi_1,\dots,\phi_{k-1},\phi_{k+1},\dots,\phi_n\}$. Then, $\{\phi_k,\boldsymbol{\phi}_{1:n,-k},\boldsymbol{\phi}_{n+1:\infty}\}$ equals the full (infinite) parameter set $\boldsymbol{\phi}_{1:\infty}$. The objective is to sample from the joint posterior density

$$p(\mathbf{z}_{1:T},\boldsymbol{\phi}_{1:\infty},\boldsymbol{\theta},\alpha|\mathbf{y}_{1:T}),$$

where $\mathbf{y}_{1:T} = \{\mathbf{y}_1,\mathbf{y}_2,\dots,\mathbf{y}_T\}$ is the full data set. Our sampling approach extends Algorithm 5 of Neal (2000) to latent variables. In contrast to Jensen and Maheu (2010) or Delatola and Griffin (2011), for example, this imposes no restrictions on the distributions with regard to conjugacy. In particular, we break the sampling algorithm down into four major steps:

(A) DPM,

(B) latent variables,

(C) parameters,

(D) hyperparameter,

where each step deals with several conditional posteriors in the tradition of Gibbs blocking. We discuss each step in detail in the following sections.

[5] For the main properties of the DPM, see e.g. Teh (2011).

3.1. Sampling Algorithm

We initialize the algorithm by drawing from the priors of $\boldsymbol{\theta}$ and α, simulating the CRP, conditional on α, and subsequently running the particle smoothing algorithm to obtain initial values for the latent variables.

3.1.1. Step (A): DPM

In order to obtain a posterior sample from the DPM, we require draws from the posteriors of the table indicators $\mathbf{z}_{1:T}$ and the infinite parameter set $\boldsymbol{\phi}_{1:\infty}$. In particular, we use two Gibbs blocks, i.e. sampling from

A.1. $p(\mathbf{z}_{1:T}|\mathbf{y}_{1:T}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)$,

A.2. $p(\boldsymbol{\phi}_{1:\infty}|\mathbf{y}_{1:T}, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\theta}, \alpha)$.

To sample the table indicators (Block A.1.) we use a version of Algorithm 5 from Neal (2000). Given the states, we iteratively draw from

$$\begin{aligned} &p(z_t|\mathbf{z}_{-t}, \mathbf{y}_{1:T}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha) \\ &\qquad \propto p(\mathbf{y}_{1:T}|z_t, \mathbf{z}_{-t}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha) p(z_t|\mathbf{z}_{-t}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha) \end{aligned}$$

for all z_t with $t = 1, \dots, T$ using an MH algorithm with a proposal equal to the prior. Thus, the acceptance probability reduces to

$$\min\left\{1, \frac{p(\mathbf{y}_{1:T}|\tilde{z}_t, \mathbf{z}_{-t}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)}{p(\mathbf{y}_{1:T}|z_t, \mathbf{z}_{-t}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)}\right\}, \tag{9}$$

where $\tilde{z}_t$ denotes a candidate table indicator. Conditioned on $\boldsymbol{\phi}_{1:\infty}$, states $\mathbf{s}_{1:T}$ and table assignments $\mathbf{z}_{1:T}$, the required likelihood $p(\mathbf{y}_{1:T}|z_t, \mathbf{z}_{-t}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)$ is straightforward to calculate from Eq. (3).

The acceptance probability in (9) is valid, if the proposal $\tilde{z}_t$ is drawn from the conditional $p(z_t|\mathbf{z}_{-t}, \mathbf{s}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)$. Noting that the latter distribution is by construction independent of $\boldsymbol{\phi}_{1:\infty}$, $\mathbf{s}_{1:T}$ and $\boldsymbol{\theta}$, this is equivalent to sampling from $p(z_t|\mathbf{z}_{-t}, \alpha)$. Via the CRP definition, we know that the distribution of the cluster pattern is exchangeable, i.e. the current z_t can be regarded as the last customer entering the restaurant. Thus, a candidate table can be drawn from a multinomial distribution, constructed from the probabilities given in Eq. (7). Hence, the current customer re-enters the restaurant filled with the remaining $T-1$ customers and gets assigned either to a new or to an existing table. Denoting table counts, excluding the current customer by $c_{k,-t}$, the probabilities for sitting at one of the occupied tables $k = 1, \dots, n$ and opening a new table are given by

$$\begin{aligned} P(\tilde{z}_t = k|\mathbf{z}_{-t}, \alpha) &= \frac{c_{k,-t}}{T-1+\alpha}, \\ P(\tilde{z}_t = n+1|\mathbf{z}_{-t}, \alpha) &= \frac{\alpha}{T-1+\alpha}, \end{aligned}$$

respectively.

Block A.2. is designed to sample the infinite parameter set $\boldsymbol{\phi}_{1:\infty}$ from

$$p(\boldsymbol{\phi}_{1:\infty}|\mathbf{y}_{1:T}, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\theta}, \alpha).$$

Any empty (non-active) tables can be neglected, since

$$p(\mathbf{y}_{1:T}|\boldsymbol{\phi}_{-k,1:n}, \tilde{\phi}_k, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \alpha) = p(\mathbf{y}_{1:T}|\boldsymbol{\phi}_{-k,1:n}, \boldsymbol{\phi}_{n+1:\infty}, \tilde{\phi}_k, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \alpha),$$

i.e. a sample from the posterior of an empty table is obtained by simply drawing from the base distribution G_0 with density g_0. Thus, Block A.2. iterates through all active tables $k = 1, \ldots, n$ and samples from

$$p(\phi_k|\mathbf{y}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\phi}_{-k}, \boldsymbol{\theta}, \alpha) \propto p(\mathbf{y}_{1:T}|\boldsymbol{\phi}_{1:n,-k}, \phi_k, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\theta}, \alpha)g_0(\phi_k)$$

using a random walk MH step with acceptance probability

$$\min\left\{1, \frac{p(\mathbf{y}_{1:T}|\boldsymbol{\phi}_{1:n,-k}, \tilde{\phi}_k, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\theta}, \alpha)g_0(\tilde{\phi}_k)}{p(\mathbf{y}_{1:T}|\boldsymbol{\phi}_{1:n,-k}, \phi_k, \mathbf{s}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\theta}, \alpha)g_0(\phi_k)}\right\}.$$

In the Chinese Restaurant interpretation, this step can be regarded as moving around the occupied tables within the restaurant.

3.1.2. Step (B): Latent Variables

Step (B) samples from

$$p(\mathbf{s}_{1:T}|\mathbf{y}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha).$$

In particular, we use a particle smoother approximation and draw from a multinomial distribution using the weights and particles from the particle smoother.

The particle filter proceeds in the spirit of Flury and Shephard (2011), see Herbst and Schorfheide (2015) for details. The idea is to approximate all required densities by a particle swarm defined as the set $\{\mathbf{s}_t, \mathbf{w}_t\}$, in which $\mathbf{s}_t \in \mathbb{R}^{N_p}$, $\mathbf{w}_t \in \mathbb{R}^{N_p}$ and N_p is the number of particles. Iterating on forecasting and updating steps, the weights $\mathbf{w}_t$ enable us to track the evolution of the swarm over time. That is, we start with a randomly drawn swarm with weights equal to unity. Subsequently, according to Bayes' Theorem, the weights are updated conditional on the observation y_t. Thus, the particle filter approximates the integral

$$\begin{aligned} p(y_t|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:T}, &\boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha) \\ &= \int p(y_t|s_t, \mathbf{y}_{1:t-1}, \mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)p(s_t|\mathbf{y}_{1:t-1}, \mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)\mathrm{d}s_t, \end{aligned}$$

by taking the mean over the appropriate set of particle weights. Given these *incremental likelihoods*, we are able to calculate an unbiased particle filter approximation of the log-likelihood

$$\log p(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha) = \log p(y_1|\mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha) + \sum_{t=2}^{T} \log p(y_t|\mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \mathbf{y}_{1:t-1}, \alpha).$$

Appendix C provides details on the particle filter.

We use the reweighting particle smoother (Doucet et al. (2000)) to obtain draws $s_t|\mathbf{y}_{1:T}$ for all t. The idea behind the smoothing algorithm is to reweight the particles by Bayes' rule, in order to obtain an approximation of the smoothed distribution of s_t, which is given by

$$p(s_t|\mathbf{y}_{1:T}) = p(s_{t+1}|\mathbf{y}_{1:t}) \int \frac{p(s_{t+1}|s_t)p(s_{t+1}|\mathbf{y}_{1:T})}{p(s_{t+1}|\mathbf{y}_{1:t})} \, \mathrm{d}s_{t+1}.$$

We refer to Särkkä (2013) for a textbook treatment.

3.1.3. Step (C): Parameters

The third block is a canonical random walk MH algorithm, which samples from

$$p(\boldsymbol{\theta}|\mathbf{y}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \alpha) \propto p(\mathbf{y}_{1:T}|\mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \alpha)p(\boldsymbol{\theta}).$$

For brevity, we refer to Greenberg (2008) for details. Note that we integrate out the latent states and use the incremental likelihoods generated by the particle filter in order to obtain an unbiased approximation to the likelihood. That is, Step (C) is equivalent to the pseudo-marginal method discussed for example, by Pitt et al. (2012) or Doucet et al. (2015).

3.1.4. Step (D): Hyperparameter

The last step samples the concentration parameter α. Conditional on the indicators $\mathbf{z}_{1:T}$, the posterior of α is independent of $\mathbf{y}_{1:T}$, $\boldsymbol{\phi}_{1:\infty}$ and $\boldsymbol{\theta}$, i.e.

$$p(\alpha|\mathbf{y}_{1:T}, \mathbf{z}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}) = p(\alpha|\mathbf{z}_{1:T}) \propto p(\mathbf{z}_{1:T}|\alpha)p(\alpha),$$

implying that the concentration parameter depends *exclusively* on the clusters. Furthermore, the CRP yields a straightforward rule for calculating the conditional density of the cluster pattern (likelihood of $\mathbf{z}_{1:T}$ given α) from Eq. (7), which we use to calculate the acceptance probability

$$\min\left\{1, \frac{p(\mathbf{z}_{1:T}|\tilde{\alpha})p(\tilde{\alpha})}{p(\mathbf{z}_{1:T}|\alpha)p(\alpha)}\right\}$$

for a random walk MH algorithm. Note that the indicators $\mathbf{z}_{1:T}$ are labels, and are exchangeable (label switching), as only the cluster pattern matters for the probability $p(\mathbf{z}_{1:T}|\alpha)$. The issue of label switching in the context of Dirichlet process mixtures is addressed in more detail by Jensen and Maheu (2010).

3.2. Savage-Dickey Density Ratio

Even though our approach deviates from the conjugate priors used in the literature, we are able to calculate the Bayes factors in favor of nested models using the Savage-Dickey density ratio (Dickey (1971)), as in Jensen and Maheu (2010). Nevertheless, a slightly more general definition is required to preserve interpretability.

Consider the nested model specification, $M_2 : \alpha = \alpha_0$, for which the limiting cases $\alpha_0 = \{0, \infty\}$ correspond to a normally distributed error term and a t-distributed error term, respectively. Denoting the unrestricted model by M_1, the Bayes factor is

$$\begin{aligned}\mathrm{BF}(\alpha = \alpha_0) &= \frac{p(\mathbf{y}_{1:T}|M_2)}{p(\mathbf{y}_{1:T}|M_1)} \\ &= \frac{p(\alpha = \alpha_0|\mathbf{y}_{1:T}, M_1)}{p(\alpha = \alpha_0)},\end{aligned}$$

i.e. the ratio of the posterior density of α to its prior, both evaluated at α_0. As the hypothesis of $\alpha_0 \to \infty$ is not operational, we follow Jensen and Maheu (2010) and define the transformed variable

$$u = \frac{\alpha}{\alpha + 1},$$

and thus $u \to 1$ ($u \to 0$) as $\alpha \to \infty$ ($\alpha \to 0$). Note that u is the probability of a second component. Using the transformation, it holds that

$$\mathrm{BF}(u = u_0) = \frac{p(u = u_0|\mathbf{y}_{1:T}, M_1)}{p(u = u_0)}.$$

In contrast to Jensen and Maheu (2010), we do not impose the restriction $p(u) = \mathcal{U}(0, 1)$, such that the approximation of the Savage-Dickey ratio by the posterior draws of u has to be corrected using the prior density of u, which is calculated from the prior of α using the transformation rule

$$p(u) = \frac{p(\alpha = \frac{u}{1-u})}{(1 - u)^2}.$$

Thereupon, plots of the rescaled posterior of u carry the same information as in Jensen and Maheu (2010) and can be interpreted equivalently. In particular, the value of the Savage-Dickey ratio can be interpreted as the Bayes factor in favor of the nested models defined by the value on the abscissa.

3.3. Density Forecast

In line with Jensen and Maheu (2010), we construct the posterior density forecast

$$p(y_{T+1}|\mathbf{y}_{1:T}) = \int p(y_{T+1}|\mathbf{y}_{1:T}, \boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \mathbf{z}_{1:T}, s_{T+1}, \alpha) p(\boldsymbol{\phi}_{1:\infty}, \boldsymbol{\theta}, \mathbf{z}_{1:T}, \alpha|\mathbf{y}_{1:T})\, \mathrm{d}\boldsymbol{\phi}_{1:\infty}\, \mathrm{d}\boldsymbol{\theta}\, \mathrm{d}s_{T+1}\, \mathrm{d}\mathbf{z}_{1:T},$$

by means of the MCMC output

$$\hat{p}(y_{T+1}|\mathbf{y}_{1:T}) = \frac{1}{N} \sum_{i=1}^{N} p_{\mathcal{N}}(y_{T+1}|\mathbf{y}_{1:T}, \boldsymbol{\phi}_{1:\infty}^{(i)}, \boldsymbol{\theta}^{(i)}, s_{T+1}^{(i)}, \mathbf{z}_{1:T+1}^{(i)}, \alpha^{(i)}),$$

where $p_{\mathcal{N}}(\cdot)$ is the density of the normal distribution.

Given draw $i = 1, \dots, N$ from the posterior of $\boldsymbol{\phi}_{1:\infty}$, $\boldsymbol{\theta}$ and $\mathbf{z}_{1:T}$, we run the particle smoother and draw a latent state s_T. Given s_T, we can generate a draw s_{T+1} with the transition Eq. (2). Subsequently, we iterate the CRP forward conditional on $\mathbf{z}_{1:T}$, which either yields $z_{T+1} \in \mathbf{z}_{1:T}$ or a new component with probability $\propto \alpha$. In the latter case, we sample $\tilde{\mu}_{T+1}$ and $\tilde{\sigma}^2_{T+1}$ from the base distribution G_0. In either case, given the drawn component parameters $\tilde{\mu}_{T+1}$ and $\tilde{\sigma}^2_{T+1}$, it is straightforward to draw y_{T+1} using the observation Eq. (1).

3.4. Nested Models

Besides the full semiparametric model, which we study in detail in Section 4, our sampling algorithm nests several model specifications and is easily adapted.

If $\mathcal{G}$ is nonparametric, and s_t observable, we can use Steps (A), (C) and (D) without the filtering step. This case corresponds to a standard DPM model, where, for example, a density estimate is required (Walker (2007)). Appendix A provides an example.

If we assume a parametric distribution $\mathcal{G}$ and latent s_t, we only require the particle filter in combination with Step (C). This is, for example, the case in DSGE models, see e.g. Fernández-Villaverde et al. (2016) and Herbst and Schorfheide (2015), or standard SV models as shown in Appendix B.

4. Semiparametric Stochastic Volatility Model

The semiparametric stochastic volatility model of Jensen and Maheu (2010) is a suitable application of the sampling algorithm, as it incorporates the flexibility of the nonparametric error term into a nonlinear state-space representation of the time-varying volatility model. In particular, the model is defined by

$$y_t = \exp(s_t/2)\epsilon_t, \qquad \epsilon_t \sim \mathcal{G}, \tag{10}$$

$$s_t = \rho s_{t-1} + \eta_t, \qquad \eta_t \sim \mathcal{N}(0, \sigma^2_\eta), \tag{11}$$

where the log volatility s_t is the latent state variable, $\mathcal{G}$ an unknown distribution and $\boldsymbol{\theta} = \{\rho, \sigma_\eta\}$. Note that we set the unconditional expectation of the latent volatility equal to zero, such that the level of the volatility is captured by the nonparametric $\mathcal{G}$, ensuring identification of the SV model.

4.1. Simulated Data

Prior to adopting our sampling algorithm to real-world data, we implement the approach using simulated data. We simulate 1500 data points of the stochastic volatility model according to Eqs. (10) and (11) with parameter values $\rho = 0.95$ and $\sigma^2_\eta = 0.04$ and a mixture distribution for the simulated error term given by

$$\epsilon_t \overset{iid}{\sim} \begin{cases} \mathcal{N}(0.2825, 0.3) & \text{with prob. } 0.8, \\ \mathcal{N}(-1.3000, 1.3) & \text{with prob. } 0.2, \end{cases}$$

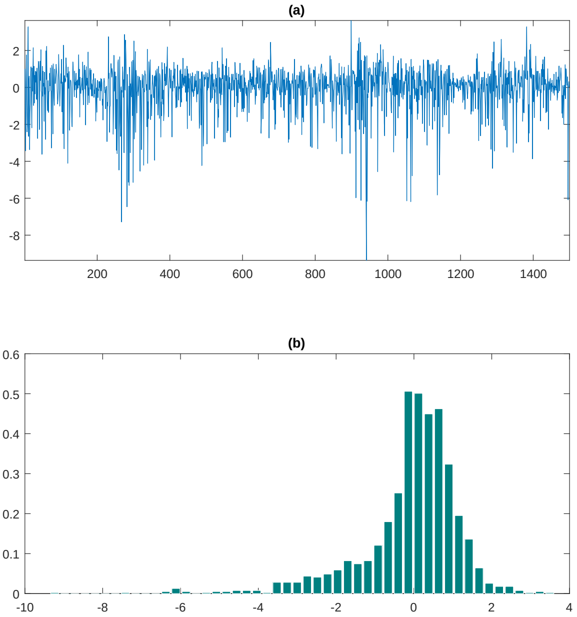

Figure 1: Simulated return data (a) and histogram of returns (b).

which scales the distribution of the observation to have zero mean, unit variance, negative skewness (≈ -1.3) and high kurtosis (≈ 8). Figure 1 plots the simulated data set.

We run the algorithm for 15000 iterations after a burn-in phase of 5000 iterations, using flat priors on the $\boldsymbol{\theta}$ parameters and parameterizing G_0 as $\mathcal{N}(0,3) \times \Gamma(1,1)$ and $p(\alpha) = \Gamma(1,1)$. Table 1 gives the posterior means and 90% Bayesian intervals. The posterior mean of the persistence parameter ρ is quite close to the true value, while the volatility σ_η of the log-volatility is slightly underestimated.

	True	Post. Mean	Interval (0.05,0.95)
ρ	0.95	0.9546	(*0.9000, 0.9821*)
σ_η^2	0.04	0.0333	(*0.0128, 0.0837*)
α	-	1.2751	(*0.4495, 2.4261*)
n	-	9.9943	(*5, 17*)

Table 1: Simulated data: Posterior means and a 90% Bayesian interval.

Figure 2 presents the graphical posterior summary. The trace plots in panels (a) and (b) and the marginal posteriors in (c) and (d) indicate that the sampling algorithm has converged to the posterior distribution. The Bayes factor (panel (e)) has the highest support at $u = 0.8$, which is in line with the underlying mixture model. The log-predictive density (blue line) in panel (i) exhibits the desired properties, i.e.

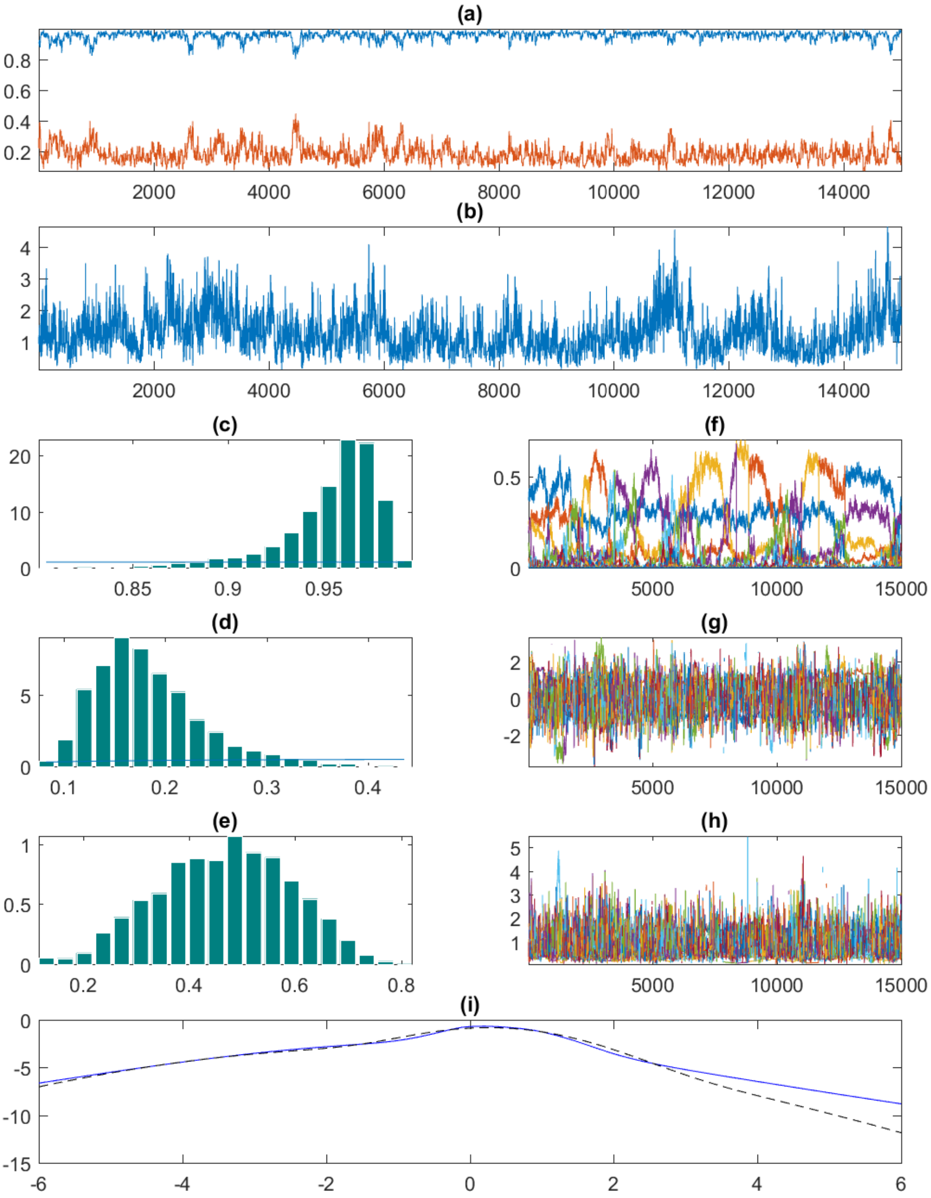

Figure 2: Simulated data application of the semiparametric stochastic volatility model): (a) trace plots of ρ (in blue) and σ_η (in red), (b) trace plots of α, (c), (d) and (e) priors (in blue) and marginal posteriors of ρ, σ_η and Bayes factors, (f) mixture weights, (g) and (h) trajectories of the mixture parameters μ_k and σ_k, (i) posterior log-predictive density (blue line) and true log-predictive density (dashed black line).

asymmetry and fat tails. Furthermore, the log-predictive density resembles the true predictive density (dashed black line) with a slightly more pronounced right tail, which we attribute to the smaller information set. Note that the true number of mixture components is two, while the average number of components n is around ten. However, most of these components are negligibly small, as can be seen in panel (f).

4.2. Real Data Application

Given the encouraging results from the simulation exercise, we turn to a real data application. We use daily S&P 500 percentage returns from 03.08.2009 to 01.05.2015 (depicted in Figure 3). The objective is to obtain a posterior sample of the parametric part of the model and to construct a one-step-ahead density forecast.

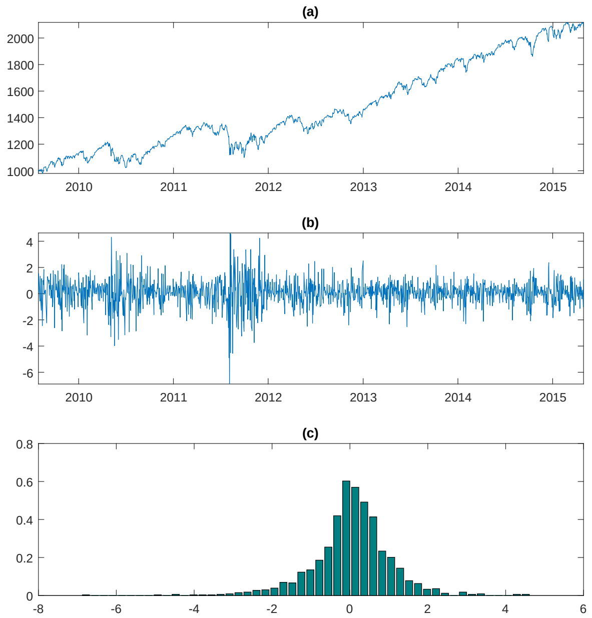

Figure 3: S&P 500 price (a), return data (b) and histogram of returns (c).

Note that the returns exhibit the typical patterns, such as heteroscedasticity and volatility clustering. Additionally, the descriptive statistics displayed in Table 2 provide further evidence of non-Gaussian behavior, in particular the negative skewness and high kurtosis. Therefore, applying the highly flexible semiparametric SV model is a natural choice.

Mean	Median	St. Dev.	Skewness	Kurtosis
0.0524	0.0736	0.9987	-0.4630	7.2468

Table 2: Descriptive statistics of the S&P 500 percentage returns.

We run the algorithm for 15000 iterations with a burn-in phase of 5000 and adopt the same priors as in Section 4.1.

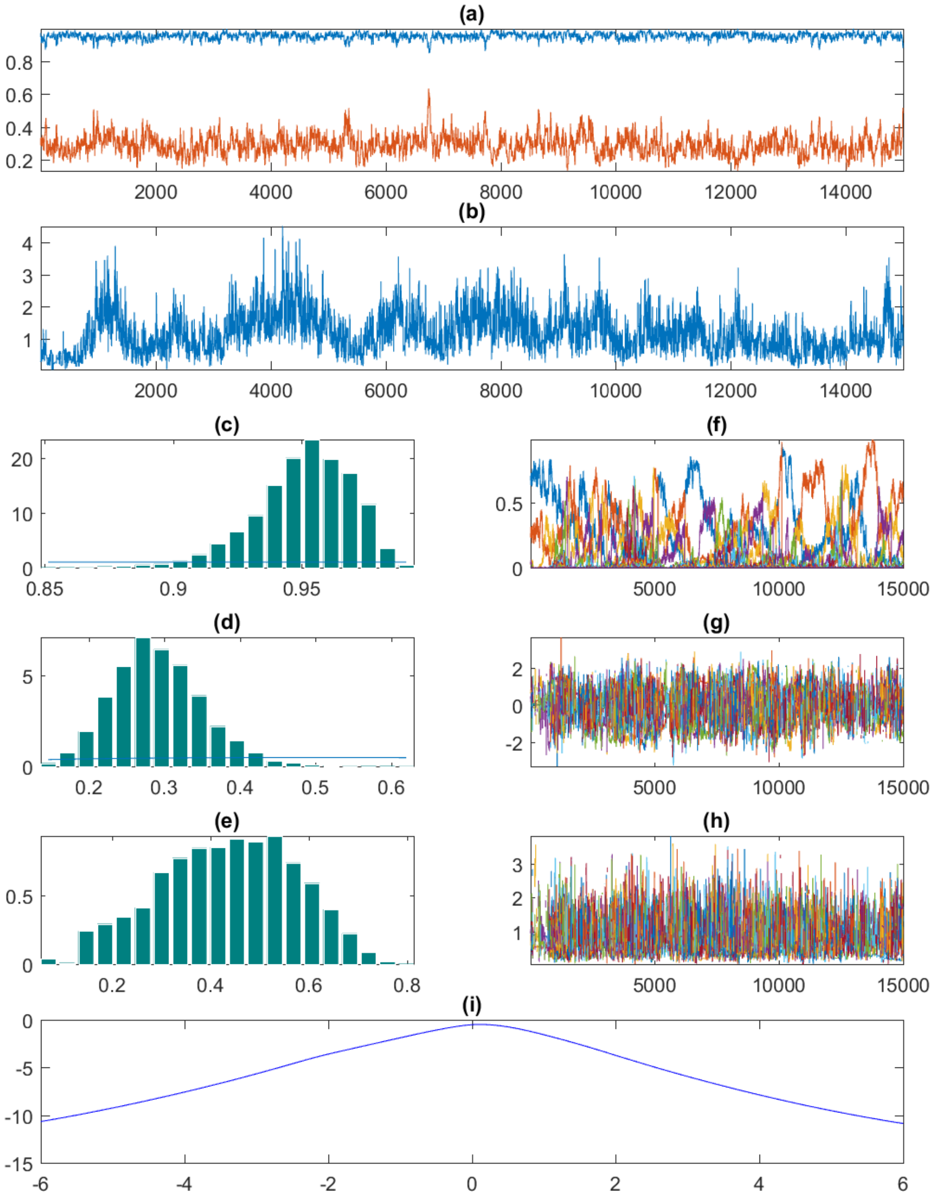

Figure 4: S&P 500 data application of the semiparametric stochastic volatility model: (a) trace plots of ρ (in blue) and σ_η (in red), (b) trace plots of α, (c), (d) and (e) priors (in blue) and marginal posteriors of ρ, σ_η and Bayes factors, (f) mixture weights, (g) and (h) trajectories of the mixture parameters μ_k and σ_k, (i) posterior log-predictive density.

The posterior means and 90% Bayesian intervals are reported in Table 3 and the complete posterior summary is shown in Figure 4. Panel (a) shows the trace plots of the $\boldsymbol{\theta}$ parameters, ρ and σ_η, indicating the convergence of the chain. Subplots (c) and (d) give the corresponding marginal posteriors, where the horizontal blue line indicates the prior. The trajectory of the concentration parameter α can be seen in panel (b), and the Bayes factor is depicted in panel (e). Note that the Bayes factors for $u_0 > 0.8$ are zero, which supports the hypothesis of a mixture. Panel (f) plots the mixture weights, and (g) and (h), the trajectories of the mixture parameters μ_k and σ_k. Those plots do not have a direct interpretation related to the model and enable us only to observe the characteristics of the DPM, such as the mixing pattern. Lastly, panel (i) shows the posterior log-predictive density, which captures the high kurtosis and the slight asymmetry observed from the raw data. In a sensitivity analysis (not reported here) we ran the algorithm in eight parallel chains with random starting values drawn from the priors, and confirmed that each chain produced comparable results.

	Post. Mean	Interval (0.05,0.95)
ρ	0.9505	(*0.9168, 0.9770*)
σ_η^2	0.0849	(*0.0404, 0.1564*)
α	1.1781	(*0.3575, 2.3076*)
n	9.1611	(*4, 16*)

Table 3: S&P Data: Posterior means and a 90% Bayesian interval.

5. Conclusion

We presented a new, flexible and general sampling algorithm for nonlinear, semiparametric state-space models. In particular, our framework integrates complex methods, such as the DPM, into a simple and intuitive estimation algorithm. As we do not rely on specific distributional assumptions or conjugacy of the priors, our approach is the first to allow for a comparison of the influence of prior distributions on semiparametric SV models. Furthermore, possible extensions include the estimation of multidimensional distributions and/or leverage effects. We leave both extensions for future research.

Appendices

A. Observable State & Nonparametric Distribution

Consider the model

$$y_t = \epsilon_t, \qquad \epsilon_t \sim \mathcal{G},$$

where $\mathcal{G}$ is a unknown distribution. In this case, we have no latent states and a nonparametric model, and use the present model to illustrate the nonparametric part of the sampler. We simulate a sample of $T = 50$ observations from the following mixture of normals

$$y_t \overset{iid}{\sim} \begin{cases} \mathcal{N}(-20, 1) & \text{with prob. } 0.2 \\ \mathcal{N}(0, 5) & \text{with prob. } 0.5 \\ \mathcal{N}(5, 1) & \text{with prob. } 0.3 \end{cases} .$$

We use Steps (A) and (D), as no filter/smoother is required. Furthermore, the likelihood (conditional on the table assignments) is given in closed form. We scale the random walk proposals to achieve an acceptance ratio of roughly 0.33. We choose non-informative priors, i.e. the base distribution $G_0(\cdot)$ is $\mathcal{N}(0, 3) \times \Gamma(1, 1)$, while the concentration parameter for the CRP α has the Gamma prior $\Gamma(1, 1)$. We run the algorithm for 20000 iterations and drop the first 5000 from the calculations. Figure A.1 shows the posterior of the concentration parameter α and the resulting predictive density. The first two panels show the trace plot of α (a), and the Bayes factors (panel (b)). Lastly, panel (c) compares the data histogram to the posterior predictive density obtained from the DPM. It is evident that the infinite mixture succeeds in identifying the distinct components and provides a flexible forecast, even given the small sample size.

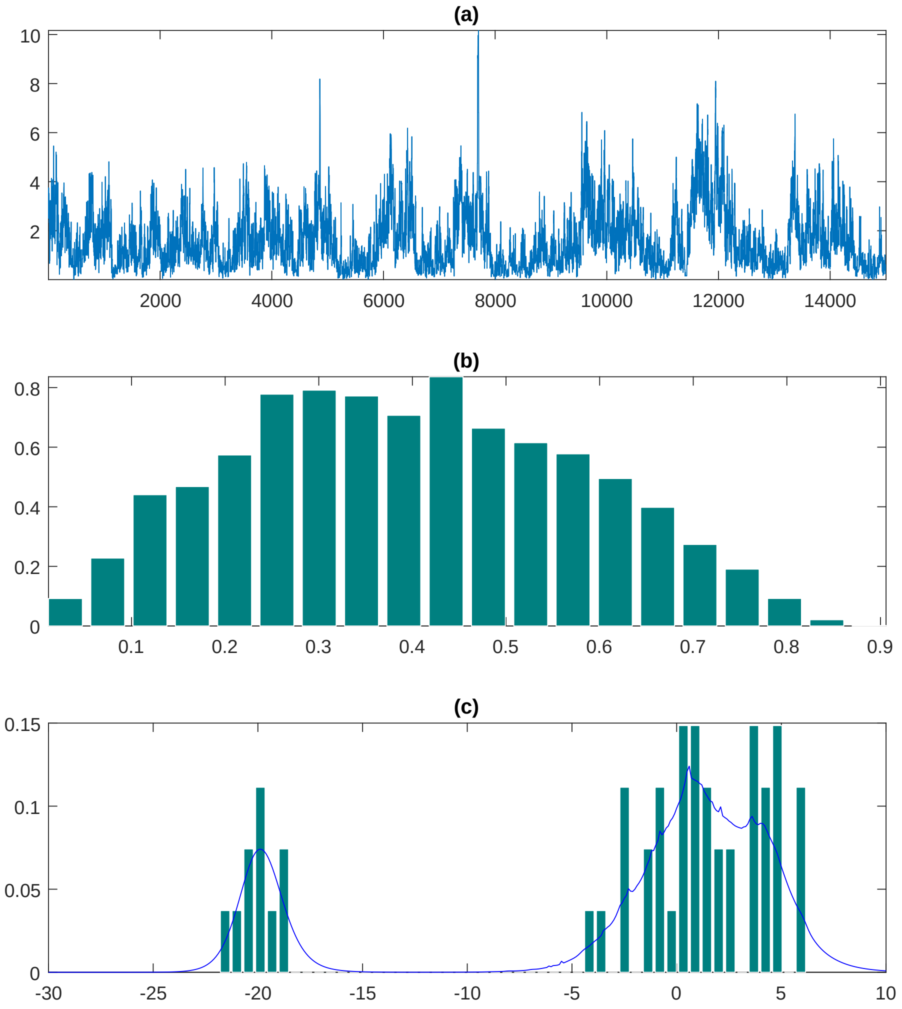

Figure A.1: Density estimation: (a) trace plot of α, (b) Bayes factors and (c) data histogram and posterior predictive density (blue line).

B. Latent State & Parametric Distribution

The second example is the stochastic volatility model with parametric error terms.

$$
\begin{aligned}
y_t &= \exp(s_t/2)\epsilon_t, & \epsilon_t &\sim \mathcal{N}(0, \sigma_y^2), \\
s_t &= \rho s_{t-1} + \eta_t, & \eta_t &\sim \mathcal{N}(0, \sigma_\eta^2).
\end{aligned}
$$

Our sampling algorithm proceeds as in Flury and Shephard (2011). We use Step (C) and the particle-filter approximation of the likelihood to sample the parameter set $\boldsymbol{\theta} = \{\rho, \sigma_\eta, \sigma_y\}$. We view this second example as a test of the particle filter's capability to deal with the latent state variable. We set the parameters to $\rho = 0.95$, $\sigma_\eta = 0.2$ and $\sigma_y = 1.2$, simulate 1000 data points from the model and report the marginal posterior densities in Figure B.1. Panel (a) shows the full chain trajectories of ρ (in blue), σ_η (in red) and σ_y (in yellow) followed by the corresponding marginal posterior distributions (panels (b), (c) and (d)), where the blue line indicates the flat prior and the red circles the true values. The last panel (e) shows the posterior predictive density obtained from our estimation. Once more, we used 20000 iterations, where only the last 15000 are used as posterior sample. The posterior means and the 90% Bayesian intervals of the model parameters are reported in Table (B.1).

	True	Post. Mean	Interval (0.05,0.95)
ρ	0.95	0.9334	(*0.8758, 0.9747*)
σ_η^2	0.04	0.0561	(*0.0238, 0.1171*)
σ_y^2	1.44	1.4509	(*1.1121, 1.9055*)

Table B.1: Simulated data: Posterior medians and a 90% Bayesian interval.

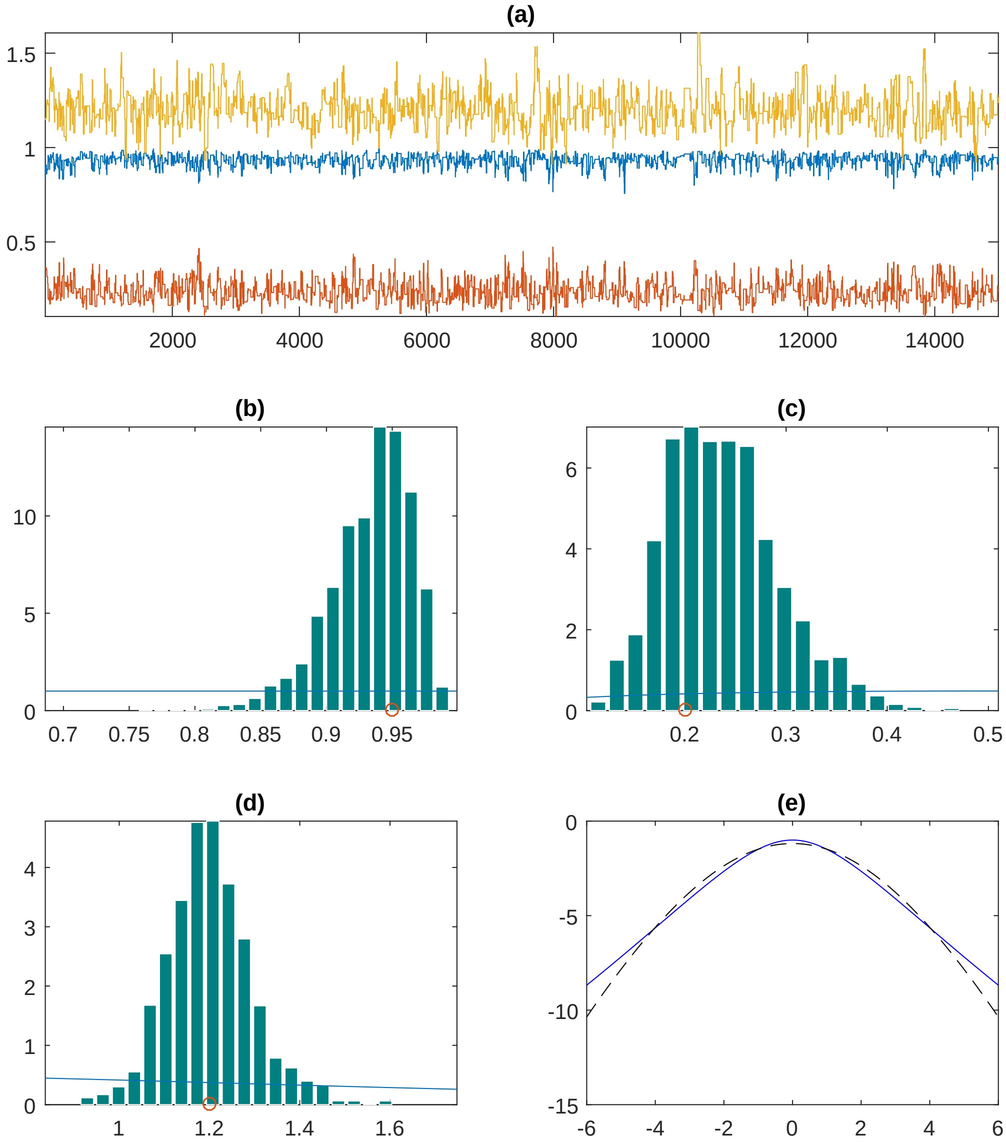

Figure B.1: Classical stochastic volatility model: (a) trace plots of ρ (in blue), σ_η (in red) and σ_y (in yellow), (b), (c) and (d) Marginal posterior of distributions of ρ, σ_η and σ_y (with priors in blue and true values indicated with red circles); (e) posterior log-predictive density (blue line) and true log-predictive density (dashed black line).

C. Particle Filter

For parsimony, we omit the parameter set from the conditioning set, i.e. we write $p(s_t|y_t) = p(s_t|y_t, \theta)$. Except for minor changes, our notation follows Herbst and Schorfheide (2015).

1. **Initialization**
   Generate a particle swarm $\{\mathbf{s}_0, \mathbf{W}_0\}$ by means of N_p i.i.d. draws from a prior distribution and set the initial weights $\mathbf{W}_0 = \mathbf{1}_{N_p}$, where $\mathbf{1}_{N_p}$ is a $N_p \times 1$ vector of ones.

2. **Recursion**. For $t = 1, \ldots, T$:

   a. **Forecast $\mathbf{s}_t$**
   Iterate $\mathbf{s}_{t-1}$ forward using the state-transition equation

   $$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \boldsymbol{\theta}, \epsilon_s).$$

   The swarm $\{\mathbf{s}_t, \mathbf{W}_{t-1}\}$ approximates the forecast density $p(s_t|\mathbf{y}_{1:t-1})$.

   b. **Forecast $\mathbf{y}_t$**
   The forecast density of y_t is

   $$p(y_t|\mathbf{y}_{1:t-1}) = \int p(y_t|s_t, \mathbf{y}_{1:t-1})p(s_t|\mathbf{y}_{1:t-1})\, ds_t$$

   with each incremental weight $p(y_t|s_t, \mathbf{y}_{1:t-1}) = w_t$ computed from the observation equation $g(\cdot)$ and the distribution $\mathcal{G}$. Consequently

   $$\hat{p}(y_t|\mathbf{y}_{1:t-1}) = \frac{1}{N_p}\mathbf{w}_t'\mathbf{W}_{t-1}$$

   is the approximate predictive density.

   c. **Updating**
   Bayes' theorem yields the updated density

   $$p(s_t|\mathbf{y}_{1:t}) = p(s_t|\mathbf{y}_{1:t-1}, y_t) = \frac{p(y_t|s_t, \mathbf{y}_{1:t-1})p(s_t|\mathbf{y}_{1:t-1})}{p(y_t|\mathbf{y}_{1:t-1})},$$

   which is approximated by the swarm $\{\mathbf{s}_t, \tilde{\mathbf{W}}_t = \frac{\mathbf{w}_t \cdot \mathbf{W}_{t-1}}{\hat{p}(y_t|\mathbf{y}_{1:t-1})}\}$.

   d. **Resampling**
   If the variation of the particles approaches a lower limit defined by the effective sample size

   $$\widehat{\mathrm{ESS}}_t = N_p / \left(\frac{\tilde{\mathbf{W}}_t'\tilde{\mathbf{W}}_t}{N_p}\right),$$

   all particles $\mathbf{s}_t$ are resampled from a multinomial distribution using weights $\tilde{\mathbf{W}}_t$. In the case of resampling, set $\mathbf{W}_t = \mathbf{1}$, and $\mathbf{W}_t = \tilde{\mathbf{W}}_t$ otherwise.

References

Blackwell, D. and J. B. MacQueen (1973). Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 353–355.

Broto, C. and E. Ruiz (2004). Estimation methods for stochastic volatility models: A survey. *Journal of Economic Surveys 18*(5), 613–649.

Delatola, E.-I. and J. E. Griffin (2011). Bayesian nonparametric modelling of the return distribution with stochastic volatility. *Bayesian Analysis 6*(4), 901–926.

Delatola, E.-I. and J. E. Griffin (2013). A Bayesian semiparametric model for volatility with a leverage effect. *Computational Statistics & Data Analysis 60*(1), 97–110.

Dickey, J. M. (1971). The weighted likelihood ratio, linear hypotheses on normal location parameters. *Annals of Mathematical Statistics 42*(1), 204–223.

Doucet, A., S. Godsill, and C. Andrieu (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing 10*(3), 197–208.

Doucet, A., M. K. Pitt, G. Deligiannidis, and R. Kohn (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika 102*, 295–313.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics 1*(2), 209–230.

Fernández-Villaverde, J. and J. F. Rubio-Ramírez (2005). Estimating dynamic equilibrium economies: Linear versus nonlinear likelihood. *Journal of Applied Econometrics 20*(7), 891–910.

Fernández-Villaverde, J., J. F. Rubio-Ramírez, and F. Schorfheide (2016). Solution and estimation methods for DSGE models. *Handbook of Macroeconomics 2*, 527–724.

Flury, T. and N. Shephard (2011). Bayesian inference based only on simulated likelihood: Particle filter analysis of dynamic economic models. *Econometric Theory 27*(5), 933–956.

Greenberg, E. (2008). *Introduction to Bayesian Econometrics.* Cambridge University Press.

Herbst, E. P. and F. Schorfheide (2015). *Bayesian Estimation of DSGE Models.* Princeton University Press.

Jacquier, E., N. G. Polson, and P. E. Rossi (2004). Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. *Journal of Econometrics 122*(1), 185–212.

Jensen, M. J. and J. M. Maheu (2010). Bayesian semiparametric stochastic volatility modeling. *Journal of Econometrics 157*(2), 306–316.

Jensen, M. J. and J. M. Maheu (2014). Estimating a semiparametric asymmetric stochastic volatility model with a Dirichlet process mixture. *Journal of Econometrics 178*, 523–538.

Kim, S., N. Shephard, and S. Chib (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *Review of Economic Studies 65*(3), 361–393.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics 9*(2), 249–265.

Pitt, M. K., R. Silva, P. Giordani, and R. Kohn (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics 171*, 134–151.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*, Volume 3. Cambridge University Press.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica 4*, 639–650.

Teh, Y. W. (2011). Dirichlet process. In *Encyclopedia of Machine Learning*, pp. 280–287. Springer.

Virbickaitė, A., H. F. Lopes, M. C. Ausín, and P. Galeano (2014). Particle learning for Bayesian non-parametric Markov switching stochastic volatility model. *Discussion papers*.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation 36*(1), 45–54.