Particle Learning for Bayesian Non-Parametric Markov Switching Stochastic Volatility Model

Audronė Virbickaitė *, Hedibert F. Lopes †, Concepción Ausín ‡ and Pedro Galeano §

Abstract. This paper designs a Sequential Monte Carlo (SMC) algorithm for estimation of Bayesian non-parametric Stochastic Volatility (BNP-SV) models for financial data. In particular, it makes use of one of the most recent particle filters called Particle Learning (PL). SMC methods are especially well suited for state-space models and can be seen as a cost-efficient alternative to MCMC, since they allow for online type inference. The posterior distributions are updated as new data is observed, which is prohibitively costly using MCMC. Further, a new non-parametric SV model is proposed that incorporates Markov switching jumps. The proposed model is estimated by using PL and tested on simulated data. Finally, the performance of the two non-parametric SV models, with and without Markov switching, is compared by using real financial time series. The results show that including a Markov switching specification provides higher predictive power in the tails of the distribution.

Keywords: Dirichlet Process Mixture, Markov Switching, MCMC, Particle Learning, Sequential Monte Carlo, Stochastic Volatility

1 Introduction

Understanding, modeling and predicting volatility of financial time series has been extensively researched for more than 30 years and interest in the subject is far from decreasing. Volatility prediction has a very wide range of applications in finance, for example, in portfolio optimization, risk management, asset allocation, asset pricing, etc. The two most popular approaches to model volatility are based on the Autoregressive Conditional Heteroscedasticity (ARCH) type models, first introduced by Engle (1982), and the SV type models, first introduced by Taylor (1982). There is evidence in the literature that SV models provide more flexibility than Generalized ARCH (GARCH, Bollerslev 1986) specifications, see e.g. Broto and Ruiz (2004).

The SV model, as introduced by Taylor (1982), allows for time-varying volatility but it is unable to capture the usual heavy-tailed behavior of conditional distribution of the returns, since they are assumed to be Gaussian. One alternative is to abandon parametric assumptions for the returns altogether and consider a semi-parametric SV model, where the distribution of the returns is modeled non-parametrically, at the same
time conserving the parametric discrete representation of the SV model.

The Bayesian non-parametric approach in SV models is quite a new field of research, with growing popularity due to its flexibility and superior performance, see Jensen and Mahen [2010, 2014] and Delatola and Griffin [2011, 2013]. In these works it is assumed that the distribution of the returns follows an infinite mixture of Normals via Dirichlet Process Mixture (DPM) models (see Ferguson [1983] and Lo [1984] among others). This is a very general class of models, since it nests other parametric specifications for the error term, used in SV setting. For instance, the Normal distribution was considered by Taylor [1986, 1994], Jacquier et al. [1994], Kim et al. [1998], the Student-t distribution was employed by Harvey et al. [1994], Gallant et al. [1997], Sandmann and Koopman [1998], Chib et al. [2002], Jacquier et al. [2004], Nakajima and Omori [2009], the Normal-Inverse Gaussian by Barndorff-Nielsen [1997], the Mixture of Normals by Mahieu and Schotman [1998], and the Generalized error distribution by Liesenfeld and Richard [2005], among many others. The infinite mixture of Normals can model other distributions, frequently used in financial time series context, see e.g. Tokdar [2006] and Mencía and Sentana [2009], because of its universal approximation property (Titterington et al. 1985).

The MCMC estimation approach for SV models is the usual methodology since the seminal work by Jacquier et al. [1994], where Bayesian inference for standard SV models was firstly developed. For a survey on Bayesian estimation of time-varying volatility models see Virbickaité et al. [2015b]. However, MCMC methods in general are computationally demanding for high-frequency data and inherently non-sequential (Lopes and Polson 2010). Alternatively, one can rely on Sequential Monte Carlo (SMC) methods, also known as particle filters, that allow for online type inference by updating the posterior distribution as the new data is observed. Stochastic volatility (parametric or non-parametric) models are state-space models, naturally suggesting SMC scheme. Moreover, models considered in this paper belong to such a class, that have the availability of sufficient statistics of the parameters. This naturally suggests using a filter that instead of tracking a high-dimensional vector of the parameters tracks a low-dimensional set of sufficient statistics that can be recursively updated. The use of sufficient statistics has been shown to increase the efficiency of the algorithm by reducing the variance of sampling weights, see Carvalho et al. (2010a).

In general, particle filters provide a simulation based approach where a set of particles represent the posterior density. For instance, consider the following state-space model, where \( x_t \) are latent states and \( \Theta \) are static parameters:

\[
\begin{align*}
    r_t | x_t, \Theta & \sim p(r_t | x_t, \Theta), \\
    x_t | x_{t-1}, \Theta & \sim p(x_t | x_{t-1}, \Theta),
\end{align*}
\]

for \( t = 1, \ldots, T \), with initial probability density \( p(x_0 | \Theta) \) and prior \( p(\Theta) \). Each particle has an associated weight that is proportional to the predictive \( p(r_t | x_t, \Theta) \). The sequential state filtering and parameter learning problem is solved by a sequence of joint posterior distributions \( p(x_t, \Theta | r^t) \), where \( r^t = (r_1, \ldots, r_t) \). Assume for the time being that \( \Theta \) is known, which leaves us with pure filtering problem. Gordon et al. [1993] and
Pitt and Shephard (1999) propose bootstrap and auxiliary particle filters, respectively, which are among the most popular ones. However, when Θ is unknown and also needs to be sequentially estimated, the problem becomes more difficult. The approach of directly introducing and resampling Θ breaks down in a few steps, since all the particles collapse into a single point. In order to delay particle degeneracy, Gordon et al. (1993), and later Liu and West (2001), consider artificial evolution for the parameters. On the other hand, Storvik (2002) and Carvalho et al. (2010a) rely on a low-dimensional set of sufficient statistics, instead of the parameters, to be tracked in time. For discussions and illustrations of some of the particle methods or reviews of particle methods in general, see Johansen and Doucet (2008), Kantas et al. (2009), Douc et al. (2009), Lopes and Tsay (2011), Lopes et al. (2011) together with Chopin et al. (2011) for a lively discussion, Lopes and Carvalho (2013) and Rios and Lopes (2013), among many others. Even if particle filters are known to suffer from a fundamental problem called particle degeneracy, i.e., an ever-decreasing set of atoms in the particle approximation of the density of interest (see Section 2.6), the online property of particle filters is definitely an advantage over MCMC.

Among all available Sequential Monte Carlo methods, in this paper we make use of the particle learning (PL) approach, which is a particle based method, firstly introduced by Carvalho et al. (2010a). Surely, alternative particle filters are in order. Nevertheless, comparison of SMC methods in this setting is out of the scope of this paper. One can find extensive empirical results for comparisons of a variety competing filters in Carvalho et al. (2010a), Lopes and Tsay (2011) and Rios and Lopes (2013) in more general settings. PL incorporates sequential parameter learning, state filtering and smoothing, thus providing an online estimation alternative to MCMC/FFBS methods. For PL comparison with MCMC see Carvalho et al. (2010a), Lopes and Polson (2010), among others. An essential feature of PL is the presence of conditional sufficient statistics for the parameters to be tracked in time. It also makes model comparison easy, since at each step we have the predictive likelihood as a by-product.

In the first part of the paper we design a PL algorithm for a SV model with DPM innovations, referred to as a SV-DPM, similar to that of Delatola and Griffin (2011). We estimate the simulated data via PL and MCMC in order to illustrate the differences in the computational aspects of the estimation approaches. PL method provides the advantage of easily incorporating the information from the new observation, while MCMC requires to re-run the algorithm again.

In the second part of the paper we augment the SV-DPM model by incorporating Markov switching jumps, as seen in So et al. (1998) and Carvalho and Lopes (2007), resulting into a new MSSV-DPM model. We extend the previously developed PL algorithm to this new model and apply it on simulated data. Then, the performance of the new MSSV-DPM model is compared with the SV-DPM specification using three real financial time series and we obtain that the new model provides better predictive power in the tails of the distribution.

The paper is structured as follows. Section 2 presents the linearized SV model with non-parametric errors and designs a PL algorithm. It also includes a discussion on
the limitations of the particle methods in general. Then, Section 3 introduces a new MSSV-DPM model and designs a PL algorithm for inference and prediction. Section 4 compares the performance of the two non-parametric models by using real data. Finally, Section 5 concludes.

2 SV-DPM Model

In this section we briefly review a commonly used version of the standard stochastic volatility model with Normal errors. We then drop the normality hypothesis and introduce a novel particle learning scheme to perform sequential Bayesian learning in the class of non-parametric SV models. The innovation distribution is assumed to follow an infinite mixture of Gaussians via Dirichlet Process Mixture models, giving rise to a SV-DPM model. We show the differences in the computational aspects between PL and MCMC, which is a gold standard in this type of models. Meanwhile, PL has the advantage of producing online inference and, as a by product, online model comparison/selection statistics.

2.1 Normal errors

The standard SV model looks as follows:

\[
y_t = \exp \left(\frac{h_t}{2}\right) v_t,
\]

\[
h_t = \alpha + \beta h_{t-1} + \tau \eta_t,
\]

where we impose \(|\beta| < 1\) for the stationarity of the volatilities; \(v_t\) and \(\eta_t\) are uncorrelated error terms, such that \(\eta_t \sim \mathcal{N}(0, 1)\). The distribution of the \(v_t\) with zero mean and unit variance takes many different forms in the existing literature: from a standard Normal, to heavy-tailed Student-t and others (see Kim et al. 1998, Chib et al. 2002, Mahieu and Schotman 1998, Liesenfeld and Richard 2005, for example).

Kim et al. (1998) proposed a linearization of the standard SV model by defining \(r_t = \log y_t^2\) and \(\epsilon_t = \log v_t^2\), resulting into the following dynamic linear model:

\[
r_t = h_t + \epsilon_t, \quad \text{where} \ \epsilon_t \sim \mathcal{F},
\]

\[
h_t = \alpha + \beta h_{t-1} + \tau \eta_t, \quad \text{where} \ \eta_t \sim \mathcal{N}(0, 1).
\]

Observe that the distribution \(\mathcal{F}\) is a \(\log \chi^2_1\) if \(v_t\) is Normally distributed. Kim et al. (1998) and Omori et al. (2007) use carefully tuned finite mixtures of Normals to approximate the \(\log \chi^2_1\) distribution and use a data augmentation argument to propose fast MCMC schemes that jointly sample \(\{h_1, \ldots, h_T\}\) based on the well-known forward filtering, backward sampling (FFBS) algorithm of Carter and Kohn (1994) and Frühwirth-Schnatter (1994).

However, the recent literature is abundant in showing that the distribution of \(v_t\) has heavier tails than the Normal distribution, rendering the above approximations useless.
Below we introduce the simple linearized SV model with non-parametric errors to model the unknown return distribution.

Another important issue concerns the moments of the distribution of $\epsilon_t$. Even though the original errors $v_t$ are generated by a process with zero mean and unit variance, the resulting moments of $\epsilon_t$ can vary greatly, depending on the distribution of $v_t$. For example, if $v_t \sim N(0,1)$, then $E[\epsilon_t] = -1.272$, $V[\epsilon_t] = 4.946$, $S[\epsilon_t] = -1.539$ and $K[\epsilon_t] = 7.015$, where $E[\cdot]$, $V[\cdot]$, $S[\cdot]$ and $K[\cdot]$ denote mean, variance, skewness and kurtosis, respectively. On the other hand, if $v_t \sim ST(1)$, scaled in such a way that $E[v_t] = 0$ and $V[v_t] = 1$, then $E[\epsilon_t] = -1.428$, $V[\epsilon_t] = 5.218$, $S[\epsilon_t] = -1.404$ and $K[\epsilon_t] = 6.583$. However, Student-t and Normal are not the only possible distributions for the errors. There is an infinite number of possibilities for the distribution of the error term, whose moments are impossible to “map” backwards in order to recover the true error distribution. Actually, the main interest usually lies in the distribution of the returns and its filtered volatilities.

2.2 Non-Normal errors

We do not specify a parametric model for the error density, but instead, we assume a Dirichlet Process Mixture prior, firstly introduced by Lo (1984). DPM models have been widely used for modeling time-varying volatilities, see Jensen and Maheu (2010, 2013, 2014), Delatola and Griffin (2011, 2013), Kalli et al. (2013), Ausín et al. (2014) and Virbickaitė et al. (2015a). This type of approach is known as time-invariant (independent) DPM.

Delatola and Griffin (2011, 2013), for example, propose to approximate the log-square of the unknown return distribution $F$ as an infinite mixture of Normals by relying on DPM models. The simple SV-DPM model presented in this section is of the same spirit as the model in Delatola and Griffin (2011). The model specification in (1) and (2) is slightly different from the one in Delatola and Griffin (2011), since we do not sum the constant volatility parameter $\alpha$ into the mixture. We leave this constant separate since in Section 3 we augment the model by considering two different volatility levels.

As seen in Escobar and West (1995), the DPM model has the following density function:

$$f(\epsilon_t; G) = \int k(\epsilon_t; \theta_t) dG(\theta_t),$$

where $k$ is some density kernel with parameters $\theta_t$ and the mixing distribution $G$ has a DP prior, denoted here by $G \sim DP(c, G_0(\theta; \varrho))$. Each observation $\epsilon_t$ comes from a different kernel density with some parameters $\theta_t$, following the mixing distribution $G$. Parameter $c$ is called the concentration parameter and $G_0(\theta; \varrho)$ is called the base distribution that depends on certain hyperparameters $\varrho$. The concentration parameter $c$ can be interpreted as the prior belief about the number of clusters in the mixture. Small values of $c$ assume a priori an infinite mixture model with a small number of components with large weights. On the contrary, large values of $c$ assume a priori an
infinite mixture model with all the weights being very small. \( c \) is also called a precision parameter and indicates how close \( G \) is to the base distribution \( G_0 \), where larger \( c \) indicates that \( G \) is closer to \( G_0 \).

**Gaussian kernel and conjugate base prior.** Considering a Gaussian kernel density, \( \epsilon_t \sim N(\mu_t, \sigma_t^2) \), the conjugate base prior \( G_0(\mu, \sigma^2; \varrho) \) is a Normal - Inverse Gamma prior, denoted here by \( G_0 \sim NIG(\mu, \sigma^2; m_0, V_0, a_0, a_0 \sigma_0^2/2) \). Here \( m_0, V_0, a_0 \) and \( a_0 \sigma_0^2 \) are the hyper-parameters in \( \varrho \).

Define \( \Phi = (\alpha, \beta, \tau^2) \) as the set of parameters associated with the parametric part of the model, \( \Omega = \{(\mu, \sigma^2)_{(j)}\}_{j=1}^{\infty} \) as a set of parameters associated with the distribution of the error term, and \( \Theta = (\Phi, \Omega) \) as a complete set of all model parameters. Therefore, the model in (1) and (2) can be rewritten as follows:

\[
\begin{align*}
  r_t | h_t, \Theta & \sim \frac{c}{c + t - 1} N(r_t; \mu_0 + h_t, \sigma_0^2) + \frac{1}{c + t - 1} \sum_{j=1}^{L_t-1} n_{t-1,j} N(r_t; \mu_j + h_t, \sigma_j^2), \\
  h_t | h_{t-1}, \Theta & \sim N(h_t; \alpha + \beta h_{t-1}, \tau^2),
\end{align*}
\]

where \( n_{t,j} \) is a number of observations assigned to the \( j \)th component at time \( t \) and \( L_t \) is a number of non-empty components in the mixture at time \( t \). Given this missing information, the mixture becomes finite, where the maximum number of components theoretically is limited by the number of observations. In practice, data tends to cluster, meaning that some observations come from the same component, therefore \( L_t \leq t \).

### 2.3 MCMC for SV-DPM

The standard Bayesian estimation of SV models, parametric or non-parametric, relies on MCMC methods, which, however, can be costly, because, additionally to the parameter estimation, they have to consider a sampler for latent volatilities.

Jensen and Mahen (2010) construct an MCMC scheme for their proposed SV-DPM model, where latent volatilities are sampled via random length block sampler, which helps to reduce correlation between draws. The authors found that the semi-parametric SV model is more robust to non-Normal data and provides better forecasts. In another paper, Jensen and Mahen (2014) consider an asymmetric SV-DPM model. The authors extend their previous semi-parametric sampler to a bivariate setting, where the innovations of the returns and volatilities are modeled jointly via infinite scale mixture of bivariate Normals.

Meanwhile, Delatola and Griffin (2011) use a linearized version of SV model. Conditional on knowing which mixture component the data belongs to, the linearized SV model is just a Normal Dynamic Linear Model (NDLM) and the latent volatilities are updated by FFBS (see the discussion at the end of Section 2.1). The remainder of the model parameters are sampled via an extension of Gibbs sampler, called hybrid Gibbs
sampler. In their subsequent paper, Delatola and Griffin (2013) consider an asymmetric SV model. Same as before, they make use of the linearization and update the latent log-volatilities via FFBS and the other parameters via Metropolis-Hastings. All above MCMC schemes are costly in the context of SV models for high-frequency data for at least two reasons: (1) the MCMC sampler has to include a filter for latent volatilities, and (2) the sampler has to be re-run each time a new observation arrives.

### 2.4 PL for SV-DPM

In this section we present the algorithm to perform PL estimation for a SV model with non-parametric errors. PL, as mentioned before, is one of several particle filters that consider sequential state filtering and parameter learning. PL, which was firstly introduced by Carvalho et al. (2010a), allows for sequential filtering, smoothing and parameter learning by including state-sufficient statistics in a set of particles. The Appendix I at the end of this paper includes a brief description of the main idea behind PL. For a more detailed explanation of PL with illustrations refer to Carvalho et al. (2010a) and Lopes et al. (2011), among others.

The priors for model parameters are chosen to be conditionally conjugate: $h_0 \sim \mathcal{N}(c_0, C_0)$, $\sigma^2 \sim \mathcal{IG}(\alpha_0^2/2, \beta_0^2/2)$, $\mu|\sigma^2 \sim \mathcal{N}(m_0, V_0)$, $\tau^2 \sim \mathcal{T}\mathcal{N}_{(-1,1)}(m_0, V_0)$ and $\alpha \sim \mathcal{N}(m_0, V_0)$. Here $\mathcal{T}\mathcal{N}_{(a,b)}$ represents Normal distribution, truncated at $a$ and $b$, while $c_0, C_0, \alpha_0, \beta_0$ are initial hyper-parameters, necessary for the parameter simulation, as well as filtered state variables, which are of two kinds: the latent log-volatilities $h_0$ and the indicator variable $k_t$, which tells us to which mixture component the error data point belongs to. The object we call particle at time $t$ thus will contain $S_t$ and corresponding parameters, simulated from the hyper-parameters in $S_t$. At each time $t$, we have a collection of $N$ particles. When this set of $N$ particles passes from $t$ to $t+1$, some of the particles disappear, some are repeated (sampling with replacement, corresponds to the Resampling step defined below) and then modified (Sampling and Propagating steps).

In order to initiate the algorithm, we need to have the initial set of sufficient statistics $S_0$ and initial parameter values. The initial parameter values are simulated from their corresponding priors. The set $S_0$ consists of: initial $\{h_0^{(i)}\}_{i=1}^N$, that has been simulated from its prior, initial $\{k_t^{(i)}\}_{i=1}^N$, which at $t=0$ are all set equal to 1, since when the first observation arrives, it will belong to the first and only component, and initial hyper-parameters $\{a_0^{(i)}\}_{i=1}^N, \{a_0^{2(i)}\}_{i=1}^N$, which at time $t=0$ are all the same across all particles.

For $t = 1, \ldots, T$ and for each particle $(i)$ the algorithm iterates through three steps (see Appendix II for the derivation of the equations):

1. **Resampling.**
Resample the particles from the previous period $t-1$ with weights

$$w \propto \frac{1}{c + t - 1} \sum_{j=0}^{L_t-1} n_j f_N(r_t; \alpha + \beta h_t-1 + \mu_j, \tau_j^2 + \sigma_j^2),$$

proportional to the predictive density of the returns ($n_0 = c$). The components of $\Theta = (\alpha, \beta, \tau^2, \mu_1, \ldots, \mu_{L_t-1}, \sigma_1^2, \ldots, \sigma_{L_t-1}^2)$ have been simulated at the end of the previous period. The resampled particles are denoted by a tilde above the particle, as in $\tilde{\Theta}$, for example.

2. Sampling.

(a) Sample new log-volatilities $h_t$ from

$$h_t|\tilde{h}_{t-1}, \tilde{\Theta}, \tilde{\alpha}, \tilde{\beta}, \tilde{\tau} \sim \sum_{j=0}^{L_t-1} \frac{n_j}{c + t - 1} N(h_t; m_{hj}, V_{hj}),$$

where, $V_{hj} = A_j \sigma_j^2$, $m_{hj} = A_j (r_t - \tilde{\mu}_j) + (1 - A_j) (\tilde{\alpha} + \tilde{\beta} h_{t-1})$, and $A_j = \tilde{\tau}_j^2 / (\tilde{\tau}_j^2 + \sigma_j^2)$.

(b) Sample new indicators $k_t$ from $\{1, \ldots, L_t^* + 1\}$, with weights proportional to

$$\tilde{n}_j f_N(r_t; \tilde{\alpha} + \tilde{\beta} h_{t-1} + \tilde{\mu}_j, \tilde{\tau}_j^2 + \sigma_j^2), \; j = 1, \ldots, L_t^* + 1,$$

where $\tilde{n}_{L_t^*+1} = c$ and $\sigma_{L_t^*+1}^2 = \sigma_0^2$.

3. Propagating sufficient statistics and learning $\Theta$.

(c.1) Sample $\tau^2$ from $\mathcal{IG}(\tau^2; b_0^*/2, b_0^* \tilde{\tau}_0^2/2)$, where

$$b_0^* = \tilde{b}_0 + 1 \quad \text{and} \quad b_0^* \tau_0^2 = \tilde{b}_0 \tilde{\tau}_0^2 + \frac{(\tilde{m}_\beta \tilde{h}_{t-1} - (h_t - \tilde{\alpha}))^2}{1 + \tilde{V}_\beta \tilde{h}_{t-1}^2}.$$

(c.2) Sample $\beta$ from $\mathcal{TN}_{(-1,1)}(\beta; m_\beta^*, V_\beta^* \tau^2)$, where

$$m_\beta^* = \frac{\tilde{m}_\beta + \tilde{V}_\beta \tilde{h}_{t-1} (h_t - \tilde{\alpha})}{1 + \tilde{V}_\beta \tilde{h}_{t-1}^2} \quad \text{and} \quad V_\beta^* = \frac{\tilde{V}_\beta}{1 + \tilde{V}_\beta \tilde{h}_{t-1}^2}.$$

(c.3) Sample $\alpha$ from $\mathcal{N}(\alpha; m_\alpha^*, V_\alpha^*)$, where

$$m_\alpha^* = \frac{\tilde{m}_\alpha \tau^2 + \tilde{V}_\alpha (h_t - \tilde{\beta} h_{t-1})}{\tau^2 + V_\alpha} \quad \text{and} \quad V_\alpha^* = \frac{\tau^2 \tilde{V}_\alpha}{\tau^2 + V_\alpha}.$$

(c.4) Sample $\sigma_k^2$ from $\mathcal{IG}(\sigma_k^2; a_0^*/2, a_0^* \sigma_0^2/2)$, where

$$a_0^* = \tilde{a}_0 + 1 \quad \text{and} \quad a_0^* \sigma_0^2 = \tilde{a}_0 \tilde{\sigma}_0^2 + \frac{(y_t - h_t - \tilde{m}_0)^2}{1 + \tilde{V}_0}.$$
Sample $\mu_k$ from $\mathcal{N}(\mu_k; m_0^*, V_0^* \sigma^2)$, where

$$m_0^* = \frac{\bar{m}_0 + \tilde{V}_0 (y_t - h_t)}{1 + \tilde{V}_0} \quad \text{and} \quad V_0^* = \frac{\tilde{V}_0}{1 + \tilde{V}_0}.$$  

2.5 Simulation exercise

We perform a short exercise based on simulated data to illustrate computational aspects of MCMC and PL approaches. A time series of length $T = 3000$ was simulated directly from the linearized model with $\alpha = 0, \beta = 0.98$ and $\tau^2 = 0.05$, where the log-square of the returns $\epsilon_t$ comes from the mixture of 7 Normals proposed by [Kim et al. (1998)] to approximate the log $\chi^2$ distribution. MCMC results are obtained via Matlab code of [Delatola and Griffin (2011)], which is available on Jim Griffin’s website[^1]. The MCMC algorithm was run for a total of 100k iterations, with the first 50k discarded as burn-in ([Delatola and Griffin (2011)] use 200k with 100k as burn-in, but we found that for our simple simulated example 100k in total was enough to ensure convergence). The prior on the persistence parameter is $\beta \sim TN(-1,1)(0.95,0.1)$ and the prior for the volatility of the volatility is $\tau^2 \sim IG(4/2,0.2/2)$. The prior for the mixture components is different than in this paper, since [Delatola and Griffin (2011)] use an alternative specification for the base distribution (see Griffin (2010)). Our PL algorithm, written in R, was run for a total of 300k particles. The hyper-parameters in the PL scheme are set as $c_0 = 0, C_0 = 0.1, m_\alpha = 0, V_\alpha = 0.25, m_\beta = 0.95, V_\beta = 0.1, b_0 = 4, b_0 \tau_0^2 = 0.2, a_0 = 5, a_0 \sigma_0^2 = 15, m_0 = -1.27, V_0 = 0.1$. The concentration parameter $c$ in both codes is set to be equal to one. For volatility process and the parameters we report the median particle as an estimate together with 97.5% and 2.5% percentile particles for 95% credible intervals.

We have split the sample into three data sets of $T = 1000, 2000$ and 3000 observations. The true advantage of the SMC becomes evident at the moment when the new observation arrives. In MCMC setting we need to re-run the entire chain all over again in order to incorporate this new information, meanwhile in SMC we just add this new information to the existing output to obtain new updated parameters and states, which is just a matter of seconds. The CPU time for both estimation approaches is presented in Table 1[^1]. Nevertheless, the computational cost of MCMC vs. PL is clearly secondary given the qualitative “online” advantage of PL.

<table>
<thead>
<tr>
<th>$T$</th>
<th>MCMC (50k+50k)</th>
<th>PL (300k particles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>23356</td>
<td>18999</td>
</tr>
<tr>
<td>2000</td>
<td>51796</td>
<td>37999</td>
</tr>
<tr>
<td>3000</td>
<td>80401</td>
<td>56999</td>
</tr>
</tbody>
</table>

[^1]: [http://www.kent.ac.uk/smsas/personal/jeg28/index.htm](http://www.kent.ac.uk/smsas/personal/jeg28/index.htm)
Here we do not compare the posterior densities produced by MCMC and PL, since the models are essentially different. Table 2 presents the estimated median parameter values with their corresponding 95% credible intervals for the PL and MCMC estimation procedures, since the interpretation of these parameters is generally the same for both models.

Table 2: Parameter estimates and their corresponding 95% CIs for PL and MCMC for $T = 1000, 2000, 3000$, where the priors are $\beta \sim \mathcal{N}(-1,1)(0.95, 0.1)$ and $\tau^2 \sim \mathcal{IG}(4/2, 0.2/2)$.

<table>
<thead>
<tr>
<th>$T$</th>
<th>MCMC Mean</th>
<th>95%CI</th>
<th>PL Mean</th>
<th>95%CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.9715</td>
<td>(0.9476, 0.9904)</td>
<td>0.9684</td>
<td>(0.9501, 0.9860)</td>
</tr>
<tr>
<td>$\beta = 0.98$</td>
<td>2000</td>
<td>0.9746</td>
<td>(0.9593, 0.9874)</td>
<td>0.9730</td>
</tr>
<tr>
<td>3000</td>
<td>0.9759</td>
<td>(0.9638, 0.9862)</td>
<td>0.9869</td>
<td>(0.9768, 0.9937)</td>
</tr>
<tr>
<td>$\tau^2 = 0.05$</td>
<td>2000</td>
<td>0.0626</td>
<td>(0.0393, 0.0941)</td>
<td>0.0697</td>
</tr>
<tr>
<td>3000</td>
<td>0.0555</td>
<td>(0.0376, 0.0787)</td>
<td>0.0541</td>
<td>(0.0432, 0.0896)</td>
</tr>
</tbody>
</table>

In order to obtain the posterior distributions for the log-volatilities at three different data points $T = 1000, 2000,$ and $3000$, MCMC had to be re-run three times for three “different” data sets, meanwhile PL just incorporated new information sequentially and the posterior distribution of any $p(h_t|r_t)$ is readily available in the estimation output.

Next, we carry out a short Monte Carlo exercise by estimating the same data set 12 times. Figure 1 draws the 12 paths of PL parameter estimation with 95% confidence bounds, as compared with the true parameter values, for different number of particles (150k top panel and 300k bottom panel). Once the chain has been run, at the arrival of the new observation the posterior distributions can be updated at a very low computational cost. The top panel of Figure 1 shows more variability across runs than the bottom panel, since the number of particles is smaller. This is due to a well known MC error, see Lopes and Carvalho (2013) for example. The mean sample standard deviation across the 12 paths of the estimated mean is 0.0164, 0.0031 and 0.0105 for 150k particles and 0.0132, 0.0023 and 0.0066 for 300k particles for $\alpha$, $\beta$ and $\tau^2$, respectively.

### 2.6 Limitations of particle filters

Particle filters, PL included, are known to suffer from a problem called particle degeneracy: an ever-decreasing set of atoms in the particle approximation of the density of interest. As noted by Chopin et al. (2011), increasing the number of observations will lead to degenerating paths, unless the number of particles is being increased simultaneously. This has to be monitored carefully for the chosen filter and can be seen as a trade-off between the sequential nature of the algorithm and stability of MCMC for very large samples. Therefore, the a priori consideration of the sample size of interest
A. Virbickaitė, H. F. Lopes, C. Ausín and P. Galeano

Figure 1: PL parameter estimates with 95% credible intervals for 12 runs, compared to the true parameter values; 150k particles (top) and 300k particles (bottom).

directly influences the choice of number of particles in order not to reach the stage where particles start to degenerate. In this work we consider time series of up to 5000 obs., which is standard in financial literature. We have chosen such number of particles, that for the series in consideration there is no sign in decrease of particle representativeness. In case of a very long series, an increase in the number of particles should suffice.

The development of particle filters is a relatively new and active field of research. The ever going quest to avoid or at least postpone particle degeneracy has lead to Gordon et al. (1993) and Liu and West (2001) introducing artificial evolution in the parameters. Another strategy is to use resample – propagate strategy rather than propagate – resample, as seen in Carvalho et al. (2010a), Lopes and Tsay (2011). Finally, the use of sufficient statistics produces lower MC error than other filters (given the same number of particles), which in turn implies that filters, making use of sufficient statistics – such as PL or Storvik (2002), can reach the same accuracy with a smaller number of particles as other filters. This leaves more room for increase in a number of particles to accommodate desired time-horizon before the particles start vanishing.
Finally, if the interest is not online type inference, MCMC is still a gold standard in the area. Recently other approaches, such as Particle MCMC, that combine MCMC and particle filters, have been emerging, see Andrieu et al. (2010) and Pitt et al. (2012), among others.

In the next section we extend the non-parametric SV model to include Markov switching jumps and design a SMC algorithm for inference and prediction.

3 MSSV-DPM Model

The simple SV model has some limitations such as it does not account for structural changes in the volatility process, which we have to take into consideration, otherwise the persistence parameter might be overestimated. In this paper we consider a two-state Markov switching SV (MSSV) model, based on the model of So et al. (1998), where the log-volatility equation is of the following form:

\[ h_t = \alpha_{s_t} + \beta h_{t-1} + \tau \eta_t, \quad \eta_t \sim \mathcal{N}(0,1), \]

where \( s_t \) are the regime variables following a two-state first order Markov Process:

\[ p_{ij} = P[s_t = j|s_{t-1} = i], \quad \text{for } i,j = 0,1. \]

As seen in Carvalho and Lopes (2007), we have to introduce the following reparameterization for \( \alpha_{s_t} \) in order to avoid identification issues:

\[ \alpha_{s_t} = \gamma_0 + \gamma_1 \mathbb{1}\{s_t = 1\}, \quad \gamma_0 \in \mathbb{R} \text{ and } \gamma_1 > 0. \]

Here \( \mathbb{1}\{s_t = 1\} \) is an indicator function that takes values equal to one if the volatility is in the high state \( (s_t = 1) \) and zero otherwise \( (s_t = 0) \). We also need to define the transition matrix between the states 0 and 1:

\[ T = \begin{bmatrix} P(s_t = 0|s_{t-1} = 0) & P(s_t = 1|s_{t-1} = 0) \\ P(s_t = 0|s_{t-1} = 1) & P(s_t = 1|s_{t-1} = 1) \end{bmatrix} = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}. \]

There are several papers that consider regime switching SV models in Bayesian context. Kalimipalli and Susmel (2004) have proposed a two-factor SV model with regime switching and estimated it using Gibbs sampler. They find that the high volatility persistence is reduced when the regimes are incorporated in the model. Also, the authors compare the new model with other two alternative two-factor models, simple SV and GARCH, and find that SV always outperforms GARCH, both in sample and out of sample. The regime switching SV performs better than the simple SV in sample, however, out of sample, it is only marginally better. On the other hand, Gerlach and Tuyt (2006) have also compared SV and GARCH models, with and without Markov switching, where the authors have found that a GARCH model with Student innovations and without regime switches outperforms its SV counterpart. Lopes and Carvalho (2007) extend SV model to multivariate case and present a Factor Stochastic Volatility (FSV) model with Markov switching jumps. They construct a novel MCMC scheme for inference and find
that the new model can capture market crashes in an instantaneous way, as opposed to the traditional FSV models. Carvalho and Lopes (2007) have constructed a sequential Monte Carlo filter by combining auxiliary particle filter (APF) with the filter of Liu and West (2001) to estimate a SV model with Markov switching regimes. They found that in terms of prediction the Markov switching SV specification outperforms a simple SV model. It is also possible to consider more than one model parameter subjected to different regimes.

Here we extend the SV-DPM model in (3) and (4) to accommodate the above regime-shifting structure:

\[
\begin{align*}
  r_t | h_t, \Theta &\sim \mathcal{N}(r_t; \mu_0 + h_t, \sigma_0^2) + \frac{1}{c + t - 1} \sum_{j=1}^{L_t-1} n_{t-1,j} \mathcal{N}(r_t; \mu_j + h_t, \sigma_j^2), \\
  h_t | h_{t-1}, \lambda_t, \Theta &\sim \mathcal{N}(h_t; \gamma_0 + \gamma_1 \lambda_t + \beta h_{t-1}, \tau^2), \\
  \lambda_t | \Theta &\sim \mathcal{BR}( (1 - p)^{1 - \lambda_{t-1}} q^{\lambda_{t-1}} ),
\end{align*}
\]

where **BER** denotes a Bernoulli distribution with parameter *p* and \(\lambda_t\) is a Bernoulli distributed state variable.

### 3.1 PL for MSSV-DPM

We extend the previous PL algorithm of SV-DPM for MSSV-DPM, by incorporating the estimation of three extra parameters and filtering of one more state variable \(\lambda_t\). The set of the parameters for the parametric part of the model is \(\Phi = (\gamma_0, \gamma_1, \beta, \tau^2, p, q)\).

Also, priors for the new parameters are: \(\gamma_0 \sim \mathcal{N}(m_{\gamma_0}, V_{\gamma_0})\), \(\gamma_1 \sim \mathcal{TN}(0, +\infty)(m_{\gamma_1}, V_{\gamma_1})\), \(p \sim \mathcal{B}(\alpha_p, \beta_p)\) and \(q \sim \mathcal{B}(\alpha_q, \beta_q)\). For \(t = 1, \ldots, T\) and for each particle \((i)\) the algorithm iterates through three steps (see Appendix II for the derivation of the equations):

1. **Resampling.**
   Resample with weights proportional to the predictive density of the returns:

   \[
   w^{(i)} \propto \frac{1}{c + t - 1} \sum_{j=1}^{L_t-1} n_j f_N(r_t; \gamma_0 + \gamma_1 \lambda_{t-1} + \beta h_{t-1} + \mu_j, \tau^2 + \sigma^2_j) + \frac{c}{c + t - 1} f_N(r_t; \gamma_0 + \gamma_1 \lambda_{t-1} + \beta h_{t-1} + \mu_0, \tau^2 + \sigma^2_0).
   \]

   Here \(\Theta = (\gamma_0, \gamma_1, \beta, \tau^2, p, q, \mu_1, \ldots, \mu_{L_{t-1}}, \sigma^2_1, \ldots, \sigma^2_{L_{t-1}})\) have been simulated at the end of the previous period. The resampled particles are denoted by a tilde above the particle, as in \(\tilde{\Theta}\), for example.

2. **Sampling.**

   (a) Sample new states of the log-volatilities \(\lambda_t\):

   \[
   \lambda_t | \lambda_{t-1}, h_{t-1}, \Theta, r_t \sim \mathcal{BR}\left( \frac{z_2}{z_1 + z_2} \right),
   \]

   where \(z_1, z_2\) are the values of the Bernoulli distribution for \(\lambda_t\).
where
\begin{equation}
   z_1 = \left[ \frac{1}{c + t - 1} \sum_{j=1}^{L_t-1} n_j f_N(r_t; \gamma_0 + \beta h_{t-1} + \mu_j, \tau^2 + \sigma_j^2) + \right. \\
   \left. \frac{c}{c + t - 1} f_N(r_t; \gamma_0 + \beta h_{t-1} + \mu_0, \tau^2 + \sigma_0^2) \right] \times \Pr(\lambda_t = 0 | \lambda_{t-1}, \Theta),
\end{equation}
\begin{equation}
   z_2 = \left[ \frac{1}{c + t - 1} \sum_{j=1}^{L_t-1} n_j f_N(r_t; \gamma_0 + \gamma_1 \lambda_{t-1} + \beta h_{t-1} + \mu_j, \tau^2 + \sigma_j^2) + \right. \\
   \left. \frac{c}{c + t - 1} f_N(r_t; \gamma_0 + \gamma_1 \lambda_{t-1} + \beta h_{t-1} + \mu_0, \tau^2 + \sigma_0^2) \right] \times \Pr(\lambda_t = 1 | \lambda_{t-1}, \Theta).
\end{equation}

Then call $\hat{\alpha} = \gamma_0 + \gamma_1 \lambda_t$.

(b) Sample new log-volatilities $h_t$:
\[ h_t | h_{t-1}, \tilde{\Theta}, \tilde{n}, \tilde{L}_{t-1}, r_t \sim \sum_{j=1}^{L_t-1} \frac{n_j}{c + t - 1} N(h_t; m_{hj}, V_{hj}) + \frac{c}{c + t - 1} N(h_t; m_{h0}, V_{h0}), \]
where
\[ m_{hj} = \frac{\tilde{\sigma}_j^2 (r_t - \tilde{\mu}_j) + \hat{\sigma}_j^2 (\hat{\alpha} + \hat{\beta} h_{t-1})}{\tilde{\sigma}_j^2 + \hat{\sigma}_j^2} \quad \text{and} \quad V_{hj} = \frac{\hat{\sigma}_j^2 \tilde{\sigma}_j^2}{\tilde{\sigma}_j^2 + \hat{\sigma}_j^2}. \]

For each particle we sample $h_t$ from a mixture of $L_{t-1}^* + 1$ components with the corresponding weights from the previous period.

(c) Sample new indicators $k_t$ from $\{1, \ldots, L_{t-1}^* + 1\}$, with weights proportional to:
\[ \tilde{n}_j f_N(r_t; \alpha + \beta h_{t-1} + \mu_j, \tau^2 + \sigma_j^2), \quad j = 1, \ldots, L_{t-1}^*, \]
where $\tilde{n}_{L_{t-1}^*+1} = c$ and $\sigma_{L_{t-1}^*+1}^2 = \sigma_0^2$.

3. Propagating sufficient statistics and learning $\Theta$.

(c1) Sample $\gamma_0$ from $N(\gamma_0; m_{\gamma_0}^*, V_{\gamma_0}^*)$, where
\[ m_{\gamma_0}^* = \frac{\tilde{m}_{\gamma_0} \tilde{\tau}^2 + \tilde{V}_{\gamma_0} (h_t - (\gamma_1 \lambda_t + \hat{\beta} h_{t-1}))}{\tilde{\tau}^2 + \tilde{V}_{\gamma_0}} \quad \text{and} \quad V_{\gamma_0}^* = \frac{\tilde{\tau}^2 \tilde{V}_{\gamma_0}}{\tilde{\tau}^2 + \tilde{V}_{\gamma_0}}. \]

(c2) Sample $\gamma_1$ from $\mathcal{T}N(0, +\infty)(\gamma_1; m_{\gamma_1}^*, V_{\gamma_1}^*)$, where
\[ m_{\gamma_1}^* = \frac{\tilde{m}_{\gamma_1} \tilde{\tau}^2 + \tilde{V}_{\gamma_1} \lambda_t (h_t - (\gamma_0 + \hat{\beta} h_{t-1}))}{\tilde{V}_{\gamma_1} \lambda_t + \tilde{\tau}^2} \quad \text{and} \quad V_{\gamma_1}^* = \frac{\tilde{\tau}^2 \tilde{V}_{\gamma_1}}{\tilde{\tau}^2 + \lambda_t \tilde{V}_{\gamma_1}}. \]

Call $\alpha = \gamma_0 + \gamma_1 \lambda_t$. 

\textbf{PL for Non-Parametric SV}
(c.3) Sample \( p \) from \( \mathcal{B}(p; \alpha_p^*, \beta_p^*) \), where
\[
\alpha_p^* = \alpha_p + 1 \text{ if } \lambda_t = 0|\lambda_{t-1} = 0 \text{ and } \beta_p^* = \beta_p + 1 \text{ if } \lambda_t = 1|\lambda_{t-1} = 0.
\]

(c.4) Sample \( q \) from \( \mathcal{B}(q; \alpha_q^*, \beta_q^*) \), where
\[
\alpha_q^* = \alpha_q + 1 \text{ if } \lambda_t = 1|\lambda_{t-1} = 1 \text{ and } \beta_q^* = \beta_q + 1 \text{ if } \lambda_t = 0|\lambda_{t-1} = 1.
\]

(c.5) Sample \( \tau^2 \) from \( \mathcal{IG}(\tau^2; b_0^*/2, b_0^*/\tau_0^*/2) \), where
\[
b_0^* = \tilde{b}_0 + 1 \text{ and } b_0^*/\tau_0^* = \tilde{b}_0 \tau_0^2 + \frac{(\tilde{m}_\beta \tilde{h}_{t-1} - (h_t - \alpha))^2}{1 + \tilde{V}_\beta \tilde{h}_{t-1}^2}.
\]

(c.6) Sample \( \beta \) from \( \mathcal{TN}_{(-1,1)}(\beta; m_\beta^*, V_\beta^* \tau^2) \), where
\[
m_\beta^* = \frac{\tilde{m}_\beta + \tilde{V}_\beta \tilde{h}_{t-1}(h_t - \alpha)}{1 + \tilde{V}_\beta \tilde{h}_{t-1}^2} \quad \text{and} \quad V_\beta^* = \frac{\tilde{V}_\beta}{1 + \tilde{V}_\beta \tilde{h}_{t-1}^2}.
\]

(c.7) Sample \( \sigma_k \) from \( \mathcal{IG}(\sigma_k^2; a_0^*/2, a_0^*/\sigma_0^2/2) \), where
\[
a_0^* = \tilde{a}_0 + 1 \text{ and } a_0^*/\sigma_0^2 = \tilde{a}_0 \sigma_0^2 + \frac{(r_t - h_t - \tilde{m}_0)^2}{1 + \tilde{V}_0}.
\]

(c.8) Sample \( \mu_k \) from \( \mathcal{N}(\mu_k; m_0^*, V_0^* \sigma^2) \), where
\[
m_0^* = \frac{\tilde{m}_0 + \tilde{V}_0 (r_t - h_t)}{1 + \tilde{V}_0} \quad \text{and} \quad V_0^* = \frac{\tilde{V}_0}{1 + \tilde{V}_0}.
\]

### 3.2 Simulated Data

In order to test the proposed model we use a simulated data set with the following parameters: \( \gamma_0 = -0.06, \gamma_1 = 0.20, \beta = 0.92, \tau^2 = 0.05, p = 0.996, q = 0.996 \). The errors follow a standard Normal distribution \( \varepsilon_t \sim \mathcal{N}(0, 1) \). The hyper-parameters are the same as in SV-DPM model, plus the new parameters: \( m_{\gamma_0} = 0, V_{\gamma_0} = 1, m_{\gamma_1} = 0, V_{\gamma_1} = 0.1, \alpha_p = 3, \beta_p = 0.1, \alpha_q = 3 \) and \( \beta_q = 0.1 \). We estimate this data with MSSV-DPM model using PL, number of particles \( N = 1m \). The 95% credible intervals for the estimated filtered log-volatility always capture the true realization of the log-volatility, same as the changes in the regimes are captured very well. The filtering estimates (volatilities and regimes) are much more stable than parameter estimates owing to the mixing properties of the algorithm. Figure 3 draws the sequential estimation of the model parameters and their 95% credible intervals for 32 independent runs. Overall, the obtained estimation results seem quite reasonable and PL is able to correctly identify volatility regimes, filter log-volatilities, estimate the density of the errors and the parameters in an efficient sequential manner.
4 Real Data Application

In this section we present a real data application using return time series for various financial assets, in particular one index - S&P500, one company - Ford - and one commodity - natural gas. The S&P500 and Ford prices are from Jan 2nd 1997 till Sept 9th 2014 and Henry Hub natural gas spot prices (dollars per million btu) are from Jan 5th 1997 till Sept 9th 2014. The summary of descriptive statistics can be seen in Table 3 and Figure 3.

Next, we estimate the data with two non-parametric models, SV-DPM and MSSV-DPM. The hyper-parameters for the priors are the same as in simulation studies for SV-DPM and MSSV-DPM models. The codes were run for 500k particles each.

To compare the performance of the models, we use the average log-predictive score (LPS) and average log-predictive tail score (LPTS_α), which restricts attention to the events in the upper 100α% of the empirical distribution of the squared returns, as seen...
A. Virbickaitė, H. F. Lopes, C. Ausín and P. Galeano

Table 3: Descriptive statistics for S&P500, Ford and Gas data.

<table>
<thead>
<tr>
<th></th>
<th>S&amp;P500</th>
<th>Ford</th>
<th>Gas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.0223</td>
<td>0.0182</td>
<td>0.0104</td>
</tr>
<tr>
<td>Median</td>
<td>0.0690</td>
<td>-0.0778</td>
<td>0.0668</td>
</tr>
<tr>
<td>St.dev.</td>
<td>1.2752</td>
<td>2.8026</td>
<td>4.4554</td>
</tr>
<tr>
<td>Skewness</td>
<td>-0.2237</td>
<td>-0.0220</td>
<td>0.7370</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>10.4789</td>
<td>15.8981</td>
<td>28.3024</td>
</tr>
<tr>
<td>T</td>
<td>4447</td>
<td>4329</td>
<td>4193</td>
</tr>
</tbody>
</table>

The LPS is defined as follows:

\[
LPS = -\frac{1}{T} \sum_{t=1}^{T} \log p(r_t|r_{t-1}),
\]

and \(LPTS_\alpha\) is defined as:

\[
LPTS_\alpha = -\frac{1}{T} \sum_{t=1}^{T} \sum_{z_\alpha} 1\{r_t > z_\alpha\} \log p(r_t|r_{t-1}),
\]

where \(z_\alpha\) is the upper 100\(\alpha\) percentile of the empirical distribution of \(r_t\). As Delatola and Griffin (2011) point out, the \(LPTS_\alpha\) is not considered a proper scoring rule, however, it can be very useful for understanding how the model performs in the tails.

The log-predictive densities are very easy to obtain in the SMC setting, since they are a by-product of the estimation procedure and, for each \(t = 1, \ldots, T\), are calculated as

\[
\log p(r_t|r_{t-1}) = \frac{1}{N} \sum_{i=1}^{N} \log p(r_t|\Theta, h_t, k_t)^{(i)}.
\]

Differently than in Delatola and Griffin (2011), there is no need to fix a certain \(\hat{\Theta}\) for the calculation of the LPS and \(LPTS_\alpha\), since we can account for parameter and state uncertainty by using the approximation in \(\Theta\).

Next, we present the estimation results for the S&P500 data set. Figures 4 and 5 present estimated predictive densities, filtered volatilities and volatility states and Table 4 presents the estimated parameters. Figure 4 shows the estimated densities for the error term as compared to the frequently used mixture of 7 Normals, as an approximation of \(\log \chi^2_1\). SV-DPM and MSSV-DPM models estimates are very similar to each other and different from the 7N approximation. This shows that the assumption of Normality is very restrictive and in most cases incorrect. As we can see in Figure 5, the filtered volatilities for both models are very similar (second and third graphs).
Additionally, the MSSV-DPM model is able to identify some different volatility regimes, especially in the second half of the data series when the algorithm had time to learn (bottom graph). As for the parameter estimation in Table 4, the volatility persistence parameter tends to be larger for the SV-DPM model, as expected, see So et al. (1998) and Kalimipalli and Susmel (2004), among others.

Table 5 presents the LPS and LPTS, for the S&P500 data. Same as in the paper by Delatola and Griffin (2011), where the authors compare parametric vs non-parametric SV models, the LPS are very similar thus making the models virtually indistinguishable. However, once we concentrate on the tails, the MSSV-DPM model tends to perform better, especially if we consider the very extreme events (the 99th percentile).

Similar results can be seen in the estimation of the other two data sets, see Tables 6, 7, 8 and 9 and Figures 6 and 7. For Ford and Natural gas data the SV-DPM model estimates larger persistence parameter, same as in the S&P500 data set. Also, the LPS for both models are very similar, but the differences appear when we consider only the tails of the distribution. Important to mention that Figures 5, 6 and 7 present only filtered, but not smoothed volatility state estimates. In general, the purpose of the analysis plays essential role. If one is interested in understanding the historical behavior of the series and the effects of, say, economic factors on the changes in volatility, it is
Figure 4: Estimated densities for the log-squared error term for SV-DPM and MSSV-DPM models, as compared to the approximation of 7 Normals.

Table 4: Parameter estimation for SV-DPM and MSSV-DPM models for S&P500 data at time $T$.

<table>
<thead>
<tr>
<th></th>
<th>SV-DPM</th>
<th>MSSV-DPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>0.0019</td>
<td>-</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.9887</td>
<td>0.9712</td>
</tr>
<tr>
<td>$\tau^2$</td>
<td>0.016</td>
<td>0.0319</td>
</tr>
<tr>
<td>$\gamma_0$</td>
<td>-</td>
<td>0.0074</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>-</td>
<td>0.0737</td>
</tr>
<tr>
<td>$p$</td>
<td>-</td>
<td>0.9998</td>
</tr>
<tr>
<td>$q$</td>
<td>-</td>
<td>0.9971</td>
</tr>
</tbody>
</table>

important to consider the information from the entire sequence (as done by MCMC, for example). In SMC setting this can be achieved by performing the backwards smoothing procedure, where PL provides a direct backward sequential pass to sample from the target smoothing distribution, see Carvalho et al. (2010a). Since MCMC produces smoothed states, in order to be comparable, SMC output also has to be smoothed. On the other hand, if one is interested in the prediction of the volatility state in the next period $r_{t+1}$, backwards smoothing does not apply. This can be clearly seen in the bottom panel of Figure 5 where observations from 1999 to 2002 are assigned to state 1, but given all of the observation sequence it is possible that smoothed states would exhibit smaller probability of being in state 1. As noted in Lopes et al. (2011), in most models estimated using PL smoothing can effectively be performed at the end.

To conclude, it seems that the SV-DPM and MSSV-DPM models tend to perform
Table 5: LPS and LPTS for SV-DPM and MSSV-DPM for S&P500 data

<table>
<thead>
<tr>
<th></th>
<th>SV-DPM</th>
<th>MSSV-DPM</th>
<th>difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPS</td>
<td>2.1956</td>
<td>2.1991</td>
<td>-0.0036</td>
</tr>
<tr>
<td>LPTS_{0.10}</td>
<td>2.5953</td>
<td>2.5527</td>
<td>0.0426</td>
</tr>
<tr>
<td>LPTS_{0.05}</td>
<td>2.8400</td>
<td>2.7826</td>
<td>0.0574</td>
</tr>
<tr>
<td>LPTS_{0.01}</td>
<td>3.3949</td>
<td>3.2398</td>
<td>0.1550</td>
</tr>
</tbody>
</table>

Table 6: Parameter estimation for SV-DPM and MSSV-DPM models for Ford data at time \( T \).

<table>
<thead>
<tr>
<th></th>
<th>SV-DPM Mean</th>
<th>95% CI</th>
<th>MSSV-DPM Mean</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>0.0707</td>
<td>(0.0641, 0.0774)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \beta )</td>
<td>0.9514</td>
<td>(0.9461, 0.9563)</td>
<td>0.908</td>
<td>(0.9041, 0.9119)</td>
</tr>
<tr>
<td>( \tau^2 )</td>
<td>0.0432</td>
<td>(0.0339, 0.0487)</td>
<td>0.0395</td>
<td>(0.0374, 0.0417)</td>
</tr>
<tr>
<td>( \gamma_0 )</td>
<td>-</td>
<td>-</td>
<td>0.1287</td>
<td>(0.1223, 0.1351)</td>
</tr>
<tr>
<td>( \gamma_1 )</td>
<td>-</td>
<td>-</td>
<td>0.139</td>
<td>(0.113, 0.167)</td>
</tr>
<tr>
<td>( p )</td>
<td>-</td>
<td>-</td>
<td>0.9994</td>
<td>(0.9981, 0.9999)</td>
</tr>
<tr>
<td>( q )</td>
<td>-</td>
<td>-</td>
<td>0.9949</td>
<td>(0.9847, 0.9992)</td>
</tr>
</tbody>
</table>

similarly, if we consider the entire predictive distribution of the returns. However, the identification of different volatility regimes becomes important if we consider the tails of the distributions, where the MSSV-DPM model performs better. This is of major interest to the investors, who are usually more interested in the tails than the entire distribution.

5 Discussion

This paper designs a sequential estimation procedure, based on PL, for a non-parametric SV-DPM model. PL is comparable to MCMC and allows for sequential inference, which is important in high-frequency data context. SMC also enables us to see the evolution of parameter learning and provides the predictive likelihoods at each data point as a by-product. Next, the existing SV-DPM model is augmented with Markov switching jumps to capture different volatility regimes. We test the new model on simulated data and find that it is able to identify different volatility regimes. Finally, we present a real data application using three financial time series of the returns for one index - S&P500, one company - Ford, and one commodity - Natural gas. We find that the new MSSV-DPM model performs as well as the SV-DPM model if we consider the entire predictive distribution of the returns. However, the MSSV-DPM model outperforms the SV-DPM model if we consider only the tails of the distribution, especially, very rare events (the 99th percentile).

As noted in the introduction, we use PL to perform sequential Monte Carlo for non-parametric SV models. Nevertheless, other particle filter alternatives are in order.
Table 7: LPS and LPTS for SV-DPM and MSSV-DPM for Ford data

<table>
<thead>
<tr>
<th></th>
<th>SV-DPM</th>
<th>MSSV-DPM</th>
<th>difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPS</td>
<td>2.0783</td>
<td>2.0756</td>
<td>0.0027</td>
</tr>
<tr>
<td>LPTS0.10</td>
<td>2.8134</td>
<td>2.7260</td>
<td>0.0873</td>
</tr>
<tr>
<td>LPTS0.05</td>
<td>3.1713</td>
<td>3.0334</td>
<td>0.1379</td>
</tr>
<tr>
<td>LPTS0.01</td>
<td>4.2408</td>
<td>3.9699</td>
<td>0.2709</td>
</tr>
</tbody>
</table>

Table 8: Parameter estimation for SV-DPM and MSSV-DPM models for Gas data at time $T$.

<table>
<thead>
<tr>
<th></th>
<th>SV-DPM</th>
<th>MSSV-DPM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean 95% CI</td>
<td>Mean 95% CI</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.2262 (0.2125, 0.2408)</td>
<td>-</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.8889 (0.8812, 0.8961)</td>
<td>0.8137 (0.807, 0.8206)</td>
</tr>
<tr>
<td>$\tau^2$</td>
<td>0.1219 (0.1129, 0.1458)</td>
<td>0.1338 (0.1245, 0.1492)</td>
</tr>
<tr>
<td>$\gamma_0$</td>
<td>-</td>
<td>0.4161 (0.397, 0.4368)</td>
</tr>
<tr>
<td>$\gamma_1$</td>
<td>-</td>
<td>0.2758 (0.2255, 0.3205)</td>
</tr>
<tr>
<td>$p$</td>
<td>-</td>
<td>0.9951 (0.9907, 0.9976)</td>
</tr>
<tr>
<td>$q$</td>
<td>-</td>
<td>0.9802 (0.9655, 0.9904)</td>
</tr>
</tbody>
</table>

Comparison of these methodologies for the particular models considered in this paper is of interest and we believe it deserves its own space.

Finally, both non-parametric models could be augmented by incorporating asymmetric volatility effect as well. Comparison of such models is of interest and could be considered for future research. Another extension is to consider models with change points that are more complex than just the intercept. Sequential inference for such models can be performed by extending the presented PL algorithm in a straightforward manner.

Appendix I: A brief review of particle learning

Define $S_t$ as an essential state vector to be tracked in time. $S_t$ is sufficient for the computation of $p(r_{t+1}|S_t)$, $p(S_{t+1}|S_t, r_{t+1})$ and $p(\Theta|S_{t+1})$. Usually it contains the filtered states and the hyper-parameters for the distributions of the model parameters $\Theta$. PL, differently than other particle methods, relies on a resample-propagate scheme, that can be understood by rewriting the Bayes theorem:

$$p(S_t|r_{t+1}) \propto p(r_{t+1}|S_t)p(S_t|r^t) :$$

Resample $p(S_t|r^t)$ with weights $p(r_{t+1}|S_t)$,

$$p(S_{t+1}|r_{t+1}) = \int p(S_{t+1}|S_t, r_{t+1})dP(S_t|r_{t+1}) :$$

Propagate $S_{t+1}$ via some propagation rules.
Table 9: LPS and LPTS for SV-DPM and MSSV-DPM for Gas data

<table>
<thead>
<tr>
<th></th>
<th>SV-DPM</th>
<th>MSSV-DPM</th>
<th>difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPS</td>
<td>2.1592</td>
<td>2.1529</td>
<td>0.0063</td>
</tr>
<tr>
<td>LPTS&lt;sub&gt;0.10&lt;/sub&gt;</td>
<td>2.8845</td>
<td>2.7875</td>
<td>0.0970</td>
</tr>
<tr>
<td>LPTS&lt;sub&gt;0.05&lt;/sub&gt;</td>
<td>3.2812</td>
<td>3.1095</td>
<td>0.1717</td>
</tr>
<tr>
<td>LPTS&lt;sub&gt;0.01&lt;/sub&gt;</td>
<td>4.5151</td>
<td>4.2300</td>
<td>0.2851</td>
</tr>
</tbody>
</table>

Here \( r_{t+1} = (r_1, \ldots, r_{t+1}) \). At \( t = 0 \) initial values for parameters and states are simulated from their prior distributions: \( \Phi_0 \) of dimension \( K \times N \) (\( N \) is the number of particles and \( K \) is the number of model parameters), \( \Omega_0 \) of dimension \( 2 \times N \) (at time \( t = 1 \) there is only one mixture component, having only two parameters) and \( h_0 \) of dimension \( 1 \times N \). Also, an essential state vector \( S_0 \) is constructed, containing all the hyper-parameters for the parameters of the model and mixture components, the volatility states and other information about the mixture. This vector is of dimension \( Z_t \times N \), where \( Z_t \) changes in time depending on the number of the mixture components. Then, PL iterates through three steps, for each particle \( (i) \), for \( i = 1, \ldots, N \):

1. **Resample** the particles with weights proportional to the posterior predictive density \( w^{(i)} \propto p(r_{t+1}|S_t^{(i)}) \) to obtain resampled particles \( \tilde{S}_t^{(i)} \). In other words, we obtain a new essential state vector \( \tilde{S}_t \) by sampling from the existing essential state vector \( S_t \) with weights that give more importance to the particles that produce higher likelihood with respect to the new data point.

2. **Propagate** the particles \( S_t^{(i)} \sim p(S_{t+1}|\tilde{S}_t^{(i)}, r_{t+1}) \). In this step we update all the elements of the essential state vector through some propagation rules.

3. **Learn** about the parameters online or off-line by approximating \( p(\Theta|r_{t+1}) \) as follows:

\[
p(\Theta|r_{t+1}) \approx 1/N \sum_{i=1}^{N} p(\Theta|S_{t+1}^{(i)}).
\]

In this step, once the elements of the essential state vector have been propagated, we use those updated hyper-parameters to sample from the posterior distributions of the parameters, obtaining new samples for the parameters \( \Theta \). In some cases it is possible to integrate out the parameter uncertainty in resample step. Then, the predictive density depends only on the essential state vector \( p(r_{t+1}|S_t^{(i)}) \). However, in many other cases it is not possible to integrate out the parameter uncertainty analytically. Then, in order to calculate the predictive density in the resample step, we use the sampled parameters, obtained from the hyper-parameters in the essential state vector: \( p(r_{t+1}|\Theta_t^{(i)}, S_t^{(i)}) \).

[Carvalho et al. (2010b)] presented a detailed explanation of PL methods for general mixtures, including DPM models. As before, \( n_{t,j} \) is a number of observations assigned
to the \( j^{th} \) mixture component at time \( t \) and \( k_t \) is an allocation variable that indicates which mixture component the observation belongs to. We can augment the essential state vector \( S_t \) by including \( n_{t,j} \) and \( k_t \). Then density estimation by using a infinite location-scale mixture of Normals via PL can be carried out by iterating through the following two steps, for each particle \( i \):

1. **Resample** with weights proportional to the predictive density \( w^{(i)} \propto p(r_{t+1}|S_t^{(i)}) \) to obtain resampled particles \( S_{t+1}^{(i)} \):

2. **Propagate** allocation variable \( k_{t+1}^{(i)} \sim p(k_{t+1}|S_{t+1}^{(i)}, y_{t+1}) \), and the rest of the sufficient statistics \( S_{t+1}^{(i)} = p(S_{t+1}|S_t^{(i)}, k_{t+1}, y_{t+1}) \), including \( n_{t+1}^{(i)} \).

The third step, parameter learning, can be performed off-line since the parameter uncertainty, as mentioned before, can be integrated out. In various simulation studies, presented in the papers above, the authors show that PL outperforms other particle filtering approaches, and is a cost-efficient alternative to MCMC methods.

**Appendix II: PL for SV-DPM and MSSV-DPM**

1. **Resampling.** Resample old particles (parameters and the set of sufficient statistics, including all state variables) with weights proportional to the predictive density of the returns, that can be obtained as below, where \( p(r_t|h_t, \Theta) \) and \( p(h_t|h_{t-1}, \Theta) \) are as in (3) and (4):

\[
p(r_t|h_{t-1}, \Theta) = \int p(r_t|h_t, \Theta)p(h_t|h_{t-1}, \Theta)dh_t
\]

\[
= \frac{1}{c + t - 1} \sum_{j=0}^{L_t-1} n_j \int f_N(r_t; h_t + \mu_j, \sigma_j^2) f_N(h_t; \alpha + \beta h_{t-1}, \tau^2)dh_t
\]

\[
= \ldots \int \frac{\exp\left(-\frac{(r_t - (h_t + \mu_j))^2}{2\sigma_j^2}\right)}{\sqrt{2\pi\sigma_j^2}} \frac{\exp\left(-\frac{(h_t - (\alpha + \beta h_{t-1}))^2}{2\tau^2}\right)}{\sqrt{2\pi\tau^2}} dh_t
\]

\[
= \frac{1}{c + t - 1} \sum_{j=0}^{L_t-1} n_j f_N(r_t; \alpha + \beta h_{t-1} + \mu_j, \tau^2 + \sigma_j^2),
\]

where \( n_0 = c \) and \( (\alpha, \beta, \tau^2, \mu_1, \ldots, \mu_{L_t-1}, \sigma_1^2, \ldots, \sigma_{L_t-1}^2) \) have been simulated at the end of the previous period.

2. **Sampling.** In this step we propagate the latent states \( h_t \) and indicator variables \( k_t \), that indicate to which mixture component the observation belongs to. Note that \( \tilde{z} \) indicates that the particle has been resampled in the first step.

(a) For sampling the \( h_t \) make use of \( p(h_t|\tilde{h}_{t-1}, r_t, \Theta) \propto p(r_t|h_t, \Theta)p(h_t|\tilde{h}_{t-1}, \Theta) \),
where $p(r_t|h_t, \Theta)$ and $p(h_t|h_{t-1}, \Theta)$ are as in (3) and (4):

$$p(h_t|\hat{h}_{t-1}, \hat{\Theta}, \hat{n}, \hat{L}_{t-1}^*, r_t) \propto \sum_{j=0}^{\hat{L}_{t-1}^{*}} \frac{\hat{n}_j}{c + t - 1} f_N(r_t; h_t + \hat{\mu}_j, \hat{\sigma}^2_j) f_N(h_t; \hat{\alpha} + \hat{\beta} h_{t-1}, \hat{\tau}^2)$$

$$h_t|\hat{h}_{t-1}, \hat{\Theta}, \hat{n}, \hat{L}_{t-1}^*, r_t \sim \sum_{j=0}^{\hat{L}_{t-1}^{*}} \frac{\hat{n}_j}{c + t - 1} N(h_t; m_{hj}, V_{hj}),$$

where, $V_{hj} = A_j \hat{\sigma}^2_j$, $m_{hj} = A_j (r_t - \hat{\mu}_j) + (1 - A_j)(\hat{\alpha} + \hat{\beta} h_{t-1})$, and $A_j = \hat{\tau}^2/\hat{\tau}^2 + \hat{\sigma}_j^2$.

(b) For sampling new indicators $k_t$, make use of $p(k_t = j|r_t, \hat{h}_{t-1}, \hat{\Theta}) \propto p(r_t|k_t = j, \hat{h}_{t-1}, \hat{\Theta}) p(k_t = j|\hat{h}_{t-1}, \hat{\Theta})$, where

$$p(r_t|k_t = j, \hat{h}_{t-1}, \hat{\Theta}) = \int p(r_t|h_t, \hat{\Theta}) p(h_t|\hat{h}_{t-1}, \hat{\Theta}) dh_t$$

and

$$p(k_t = j|\hat{h}_{t-1}, \hat{\Theta}) \propto \frac{\hat{n}_j}{c + t - 1},$$

therefore,

$$p(k_t = j|r_t, \hat{h}_{t-1}, \hat{\Theta}) \propto \hat{n}_j f_N(r_t; \hat{\alpha} + \hat{\beta} h_{t-1} + \hat{\mu}_j, \hat{\tau}^2 + \hat{\sigma}^2_j), \ j = 1, \ldots, \hat{L}_{t-1}^* + 1,$$

where $\hat{n}_{\hat{L}_{t-1}^* + 1} = c$ and $\sigma^2_{\hat{L}_{t-1}^* + 1} = \sigma_0^2$.

3. Propagating sufficient statistics and learning $\Theta$.

(c.1) & (c.2) Sampling $\tau^2$ and $\beta$:

$$p(\beta, \tau^2|h_t) \propto p(h_t|\beta, \tau^2) p(\beta, \tau^2)$$

$$\propto f_N(h_t; \hat{\alpha} + \hat{\beta} h_{t-1}, \hat{\tau}^2) f_TN_{(\beta, \hat{m}_\beta, \hat{V}_\beta \tau^2)} f_{IG}(\tau^2; \hat{b}_0/2, \hat{b}_0 \hat{\tau}_0^2/2)$$

$$\tau^2 \sim IG(\tau^2; \hat{b}_0 + 1, \frac{\hat{b}_0 \hat{\tau}_0^2}{2})$$

$$\beta \sim TN_{(\beta, \hat{m}_\beta, \hat{V}_\beta \tau_{t-1}^2)}(\beta; \frac{\hat{m}_\beta + \hat{V}_\beta h_{t-1}(h_t - \hat{\alpha})}{1 + \hat{V}_\beta h_{t-1}^2}, \frac{\hat{V}_\beta \tau_{t-1}^2}{1 + \hat{V}_\beta h_{t-1}^2})$$

Sufficient statistics updates and sampling for the rest of the parameters is analogous to (c.1) & (c.2).

PL algorithm for MSSV-DPM model is very similar to the SV-DPM, except for an extra state variable $\lambda_t$ that needs to be propagated, according to the following:

$$p(\lambda_t|\hat{\lambda}_{t-1}, \hat{h}_{t-1}, \hat{\Theta}, r_t) \propto p(r_t|\hat{\lambda}_{t-1}, \hat{h}_{t-1}, \hat{\Theta}) p(\lambda_t|\hat{\lambda}_{t-1})$$

$$\lambda_t|\hat{\lambda}_{t-1}, \hat{h}_{t-1}, \hat{\Theta}, r_t \sim BEG \left( \frac{z_2}{z_1 + z_2} \right),$$
where $z_1$ and $z_2$ as in (5).

Acknowledgments

We are very grateful to the Associate Editor and Referees for insightful comments and suggestions that helped to improve the manuscript.

Virbickaitė, A. and Ausín, C.M. are grateful for the financial support from MEC grant ECO2011-25706. Galeano, P. acknowledges financial support from MEC grant ECO2012-38442.

References


A. Virbickaitė, H. F. Lopes, C. Ausín and P. Galeano


Figure 5: Filtered volatilities and volatility states for S&P500 data for SV-DPM and MSSV-DPM models.
Figure 6: Filtered volatilities and volatility states for Ford data for SV-DPM and MSSV-DPM models.
Figure 7: Filtered volatilities and volatility states for Gas data for SV-DPM and MSSV-DPM models.