

Lux, Thomas

Working Paper

Inference for nonlinear state space models: A comparison of different methods applied to Markov-switching multifractal models

Economics Working Paper, No. 2018-07

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Department of Economics

Suggested Citation: Lux, Thomas (2018) : Inference for nonlinear state space models: A comparison of different methods applied to Markov-switching multifractal models, Economics Working Paper, No. 2018-07, Kiel University, Department of Economics, Kiel

This Version is available at:

<https://hdl.handle.net/10419/181491>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

C | A | U

Christian-Albrechts-Universität zu Kiel

Department of Economics

Economics Working Paper
No 2018-07

Inference for Nonlinear State Space Models: A Comparison of Different Methods applied to Markov-Switching Multifractal Models

by Thomas Lux

issn 2193-2476



Inference for Nonlinear State Space Models: A Comparison of Different Methods applied to Markov-Switching Multifractal Models

Thomas Lux *
University of Kiel

August 20, 2018

Abstract

Nonlinear, non-Gaussian state space models have found wide applications in many areas. Since such models usually do not allow for an analytical representation of their likelihood function, sequential Monte Carlo or particle filter methods are mostly applied to estimate their parameters. Since such stochastic approximations lead to non-smooth likelihood functions, finding the best-fitting parameters of a model is a non-trivial task. In this paper, we compare recently proposed iterative filtering algorithms developed for this purpose with simpler online filters and more traditional methods of inference. We use a highly nonlinear class of Markov-switching models, the so called Markov-switching multifractal model (MSM), as our workhorse in the comparison of different optimisation routines. Besides the well-established univariate discrete-time MSM, we introduce univariate and multivariate continuous-time versions of MSM. Monte Carlo simulation experiments indicate that across a variety of MSM specifications, the classical Nelder-Mead or simplex algorithm appears still as more efficient and robust compared to a number of online and iterated filters. A very close competitor is the iterated filter recently proposed by Ionides et al. (2006) while other alternatives are mostly dominated by these two algorithms. An empirical application of both discrete and continuous-time MSM to seven financial time series shows that both models dominate GARCH and FIGARCH models in terms of in-sample goodness-of-fit. Out-of-sample forecast comparisons show in the majority of cases a clear dominance of the continuous-time MSM under a mean absolute error criterion, and less conclusive results under a mean squared error criterion.

Keyword: partially observed Markov processes, state space models, Markov-switching multifractal model, nonlinear filtering, forecasting of volatility

JEL codes: C20, G15

Author's Address: Department of Economics, University of Kiel, Olshausenstr. 40, 24118 Kiel, Germany, Email: lux@economics.uni-kiel.de.

*Acknowledgment: A major part of this research has been conducted during a visit to the Business School of Deakin University in Melbourne. I am most grateful to many colleagues of Deakin's Department of Finance for stimulating discussions during my visit, and I am particularly thankful to Ruipeng Liu for valuable comments, and to Boyan Yanovski for his careful reading of the code.

1 Introduction

State space models have found wide applications in many areas (e.g. Zeng and Wu, 2013; Chen, 2015). Estimating parameters of such models is straightforward if both the hidden (or latent) state variable and the observation follow linear, Gaussian processes. In this case, the time-honored Kalman filter provides the most efficient algorithm for statistical inference (Kalman, 1960). If one or both of the processes governing the state and observation obey some nonlinear, non-Gaussian law of motion, matters tend to become more complicated. Early literature has concentrated mostly on developing extensions or approximations of the Kalman filter method (c.f. Grewal and Andrews, 2008). More recently, the availability of versatile hard- and software has shifted the focus towards the development of simulation-based methods of inference. The perhaps most seminal development in this area has been the invention of the so-called particle filter by Gordon et al. (1993) and Kitagawa (1996). The particle filter has found many applications in diverse areas, and provides a straightforward avenue for filtering information on the hidden variables. It can also be applied to approximate numerically the likelihood as an input to statistical inference. The main practical obstacle here is the point-wise nature of the approximation of various densities obtained by a particle filter. This point-wise approximation leads to an optimization problem with a non-smooth objective function for which the usual derivative-based optimization routines are not applicable. Malik and Pitt (2011) have proposed a transformation of the filtering density in order to establish a smooth approximation to the likelihood. However, their approach is restricted to univariate state spaces and is also not applicable to certain classes of models for which there is an inherent lack of smoothness of the state evolution itself (cf. Lux, 2017). Hence, in most applications one would have to confront the task of finding the maximum of a non-smooth, rugged objective function for parameter inference in a nonlinear state space setting.

There exist various time-honored methods for optimization of non-smooth functions that indeed have already been used for state-space models, e.g., the Nelder-Mead or simplex method (Calvet et al., 2006; Liu and Lux, 2015) or simulated annealing (Fernández-Villaverde and Rubio-Ramirez, 2007). Besides these classical all-purpose optimization routines, some methods specifically tailored towards nonlinear state-space models have been developed in recent literature. Two methods denoted *Iterated Filtering* have been introduced by Ionides et al. (2006, 2011, 2015) and applied to ecological, epidemiological and financial models by King et al. (2008), Bhadra et al. (2011), He et al. (2010) and Bretó (2014).

The two available versions of this approach could both be demonstrated to converge to the maximum likelihood estimate of the underlying model under very mild conditions on the structure of the state space model. Both of these approaches are based upon earlier versions of *online* filtering algorithms that attempt to identify parameters via some sort of evolutionary dynamic selection executed in just one sweep through the data, but the new algorithms perform this optimization repeatedly.

Given these new methods and the simpler online estimators they have been inspired by, the applied researcher now has indeed a portfolio of numerical optimization schemes to choose from. Unfortunately, hardly any guidance exists so far on the relative strengths and weaknesses of different approaches. Even for single methods, not much evidence on the quality of the estimates they deliver is available. In order to shed light on the relative performance of different methods, we have conducted a series of Monte Carlo experiments

on which we report in this paper. Our basic interest here is in the performance of the novel iterated filtering approaches vis-à-vis more traditional optimizers. To this end we have compared the efficiency of these algorithms and their computational demands with a baseline Nelder-Mead approach using different specifications of the former. We also pursue the question whether the iterations of the underlying filtering operations are really adding precision, or whether a baseline online algorithm executed just once, but with more computational effort (more particles) would achieve the same goal.

Our workhorses in the present paper are various versions of the so-called Markov-Switching Multifractal Model (MSM) introduced by Calvet and Fisher (2001) which is based upon Mandelbrot’s (1974) formalism for turbulent flows, but adapted to the setting of asset returns in financial markets. MSM is a parsimoniously parameterized framework that allows for a large number of states of the unobserved variable (which in the applications of this model is financial volatility). It is also highly nonlinear since the volatility dynamics is modelled as a multiplicative, hierarchically structured process. MSM has been widely applied in finance recently and has mostly been found to improve upon the performance of the standard toolbox of GARCH-type models that conceive the volatility dynamics as a linear process (Calvet and Fisher, 2004; Calvet et al., 2017; Lux et al., 2014; among others, and Lux and Segnon, 2018, for a survey of this literature). We will consider both the univariate discrete MSM of Calvet and Fisher (2004) as well as the continuous-time version of Calvet and Fisher (2001) and multivariate versions of the later. While for the discrete model, we can compare the performance of numerical approximations to that of full analytical maximum likelihood, no analytical solutions are available for the continuous-time versions. Our results will, therefore, also demonstrate that continuous MSM models can be estimated more or less as efficiently as their discrete counterparts, even in the multivariate case. This is important because continuous-time models are preferred in financial engineering.

The rest of this paper will consist of the following parts: Section 2 will introduce the models, sec. 3 provides an overview over the estimation methods, and sec. 4 presents the results of the Monte Carlo simulations. Sec. 5 provides an application to a sample of seven important time series of financial returns. An out-of-sample forecasting competition shows that the continuous MSM model dominates alternatives from the GARCH and FIGARCH family in five out of seven cases for most forecasting horizons under the mean absolute error criterion. Under a mean squared error criterion results for short horizons are mostly inconclusive (i.e., a number of models provides sensible forecasts and none of those can be clearly outperformed by other competitors), but the multifractal models often turn out to provide superior forecasts over longer horizons. A comparison of different estimators confirms the insights of the proceeding Monte Carlo studies. Sec. 6 finally provides concluding remarks.

2 Markov-Switching Multifractal Models

Multifractal models date back to Mandelbrot’s (1974) work on cascades of energy flux in statistical physics. The very same combinatorial approach has been proposed and brought to fruition as a model of asset volatility twenty-five years later by Mandelbrot et al. (1997), which meanwhile has spawned a sizable new strand of literature in financial econometrics. Calvet and Fisher (2001) were the first to propose an iterative, causal version of the multifractal model that overcomes the cumbersome combinatorial nature of the original framework. Calvet and Fisher (2004) were the first to estimate a discrete-time

version of the MSM model using standard econometric techniques.

The *univariate, discrete-time* MSM models asset returns, r_t as

$$r_t = \sigma \left(\prod_{i=1}^k M_t^{(i)} \right)^{1/2} u_t \quad (1)$$

with u_t following a standard Normal distribution, and σ being a constant scale factor. $M_t^{(i)}$ are volatility factors of different average lifetimes that taken together define a Markov-switching process with 2^k states. Available studies have considered values of k around 8 so that the number of states of this model by far exceeds the numbers usually considered in baseline Markov-switching models. The interesting dynamics of the MSM models stems from the fact that the volatility components $M_t^{(i)}$ are all renewed or preserved at any time t with different probabilities γ_i and $1 - \gamma_i$. A very convenient way to allow for a large spectrum of heterogeneous mean life times is the following formulation of renewal probabilities

$$\gamma_i = 2^{-(k-i)}, \quad (2)$$

which actually corresponds almost exactly to the frequency of changes of multipliers in the original Mandelbrot (1974) framework. The multipliers $M_t^{(i)}$ themselves can be assumed to be drawn from a discrete or continuous distribution. The simplest case here is a Binomial distribution with parameters $\{m_0, 2 - m_0\}$ that also implies the normalization $E[M_t^{(i)}] = 1 \forall i, t$. Other possibilities have been explored and results have been found to differ only little across specifications (cf. Lux, 2008).

The discrete-time MSM can be estimated by full analytical maximum likelihood (cf. Calvet and Fisher, 2004) so that the efficiency of various numerical approximations to the likelihood function and optimization algorithms can be compared to this benchmark. It is worthwhile to mention that despite the availability of analytical ML for this model, numerical methods could still be of value as well. Namely, when k increases, the computational demands of the analytical approach are increasing tremendously, and it would be necessary to apply numerical approximations at some point for which the acceptable computational effort could be chosen via some of their parameters.

In *continuous-time* MSM, renewals do not occur at fixed points in time, but are modeled via Poisson processes determining the change points of all multipliers $M_t(i)$. Hence, the γ_i defined in eq. (2) would now be the parameters of Poisson processes obeyed by each of the volatility components, and returns could not only be computed over unit time intervals (as in eq. 1), but over any frequency. We assume nevertheless that discrete observations of this continuous-time process would be available at equidistant time points with unitary length. Formally, the continuous-time univariate MSM can be written as a diffusion process switching between 2^k states identified by their respective diffusion terms:

$$dp_t = \sigma(s_t)dw_t \quad (3)$$

with p_t denoting the log financial price, w_t standard Brownian motion, $\sigma(\cdot) = \sigma_i$ if $s_t = i$ ($i = 1, 2, \dots, 2^k$) and s_t a continuous-time Markov chain governed by an infinitesimal matrix Q with intensities q_{ij} whose structure obeys the multifractal principle outlined above (formalized in eq. 2).

As an example, with $k = 2$, and a Binomial distribution of volatility components with parameters m_0 and $2 - m_0$, we obtain $2^2 = 4$ states: m_0m_0 , $(2 - m_0)m_0$, $m_0(2 - m_0)$ and

$(2 - m_0)(2 - m_0)$ with intensity matrix

$$Q_{k=2} = \begin{pmatrix} -(\gamma_1 + \gamma_2) & \gamma_1 & \gamma_2 & 0 \\ \gamma_1 & -(\gamma_1 + \gamma_2) & 0 & \gamma_2 \\ \gamma_2 & 0 & -(\gamma_1 + \gamma_2) & \gamma_1 \\ 0 & \gamma_2 & \gamma_1 & -(\gamma_1 + \gamma_2) \end{pmatrix}. \quad (4)$$

For higher k , we can similarly derive the intensity matrix which would become more and more sparse because in continuous-time processes, at most one of the volatility components would change during an infinitesimal time step.

It is worthwhile to note that we can perform exact simulations of this process using the celebrated Gillespie algorithm (Gillespie, 1977) to not confound approximation errors of the underlying process with estimation uncertainty. The only attempt at estimating the parameters of the Poisson MSM in extant literature is by Leövey (2015) who has estimated this model via a simulated method of moments algorithm based upon the algorithm proposed by Lux (2008) for the discrete-time version.

Multivariate MSM models have so far only been used in the form of discrete-time processes (Calvet et al., 2006; Liu and Lux, 2015, 2017). Here we add a continuous-time bivariate and trivariate version to the multifractal zoo. To arrive at a parsimonious parametrization, we take our inspiration from Liu and Lux (2015). Their idea is to generate comovements in the volatility of different assets by assuming that they have a number of low frequency volatility components in common while their higher frequency components might be independent of each other. The motivation of their assumption is that long-term trends in volatility could be determined by economy-wide macroeconomic fluctuations while short-term fluctuations might have idiosyncratic sources. In the bivariate case, the overall volatility process $g(M_{q,t})$ of any of two assets $q = 1, 2$ is then given by

$$g(M_{q,t}) = \prod_{i=1}^{k_1} M_t^{(i)} \prod_{j=k_1+1}^k M_{q,t}^{(j)} \quad (5)$$

with the low-frequency components $1, \dots, k_1$ being the same for both assets, and the high frequency ones k_1+1, \dots, k being distinct and determined independently. We again assume that at all levels the renewal processes are driven by Poisson processes with parameters γ_i .

The bivariate continuous-time MSM would then formally be described by a bivariate switching diffusion between all common and idiosyncratic states, i.e.

$$dp_1 = \sigma(s_t, u_{1,t}) dw_1 \quad (6)$$

$$dp_2 = \sigma(s_t, u_{2,t}) dw_2 \quad (7)$$

with $s_t = i$ ($i = 1, 2, \dots, 2^{k_1}$) determining the realization of the first part of the product on the left-hand side of eq. (5), and $u_{q,t} = l$ ($l = 1, 2, \dots, 2^{k-k_1}$) determining the realization of the second part, independently for $q = 1$ and $q = 2$, but with the same probabilities of renewal. w_1 and w_2 denote a bivariate Brownian motion with covariance matrix Σ that contributes three parameters, say σ_1 , σ_2 and ρ , to our estimation problem. We again assume that the distribution of the volatility components is Binomial, with possibly different parameter values $\{m_1, 2 - m_1\}$ and $\{m_2, 2 - m_2\}$ for the two individual diffusions within the bivariate model. The extension of this setup to the trivariate case and beyond is straightforward: We only have to increase the number of assets from $q = 2$ to $q = 3$

or more preserving a number of k_1 common components and $k - k_1$ independent ones. A trivariate continuous-time MSM would have a total of nine parameters: The three Binomial parameters (say, m_1 , m_2 and m_3) and the six entries of their variance-covariance matrix.

Fig. 1 provides an illustration of the evolution of returns of two synthetic assets following a continuous-time bivariate Poisson MSM with parameters $m_1 = 1.4$, $m_2 = 1.6$, $k = 10$, $k_1 = 5$ and covariance matrix of the innovations with $\sigma_1 = \sigma_2 = 1$ and $\rho = 0.4$. We observe a certain co-movement in the volatility outbursts in agreement with the stylized facts of financial data (e.g., Forbes and Rigobon, 2002).

In the simulation of this bivariate model, we have again assumed that the k multipliers follow Poisson transition rates γ_i according to eq. (2). We could also have adopted the more complex dependency structure of the discrete bivariate MSM of Calvet et al. (2006) to the continuous-time case. For the sake of parsimonious parametrization, the simpler specification (which can be seen as a special case of Calvet et al.'s) is preferred.

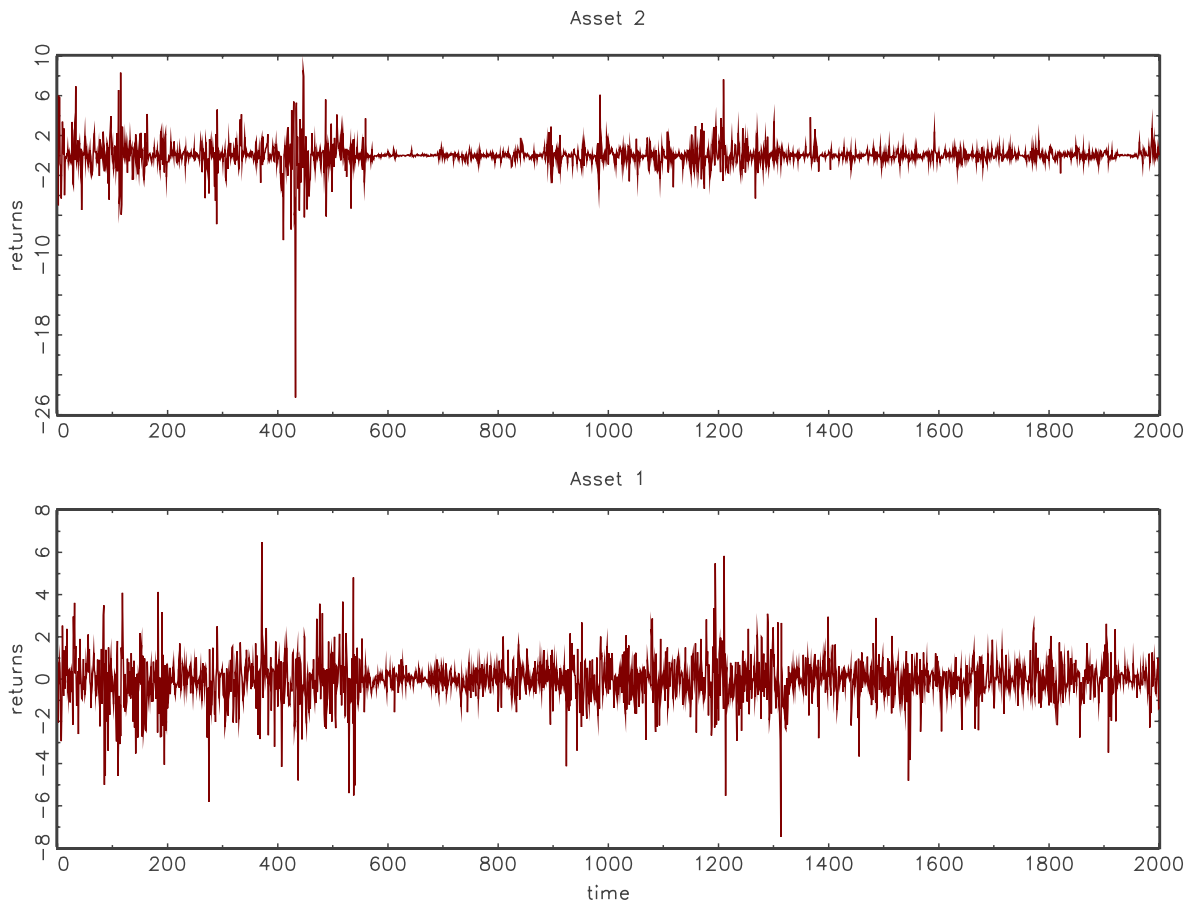


Figure 1: Realization of a bivariate continuous-time MSM process as a model of asset returns. Parameters are given in the main text.

3 Methods

Our basic goal is to compare different estimators of the parameters of the above multi-fractal processes in terms of their efficiency and computational demands. In the center of interest is the comparison between a statistical all-purpose optimization routine, the Nelder-Mead or simplex algorithm, and the more recently proposed iterative filtering approaches. Both alternatives attempt to find the parameters maximizing the likelihood, although the iterative filtering approaches pursue this goal in an unconventional manner. We also explore whether iterating the filtering operations really adds value. To do so we compare the iterated filtering algorithms with their *online* (one-time filtering) counterparts. Applying the filter only once, of course, saves on computation time which could be used to invest in the precision of the filter. Such an investment would consist in an increase of the number of particles used for the point-wise approximation of the conditional densities entering the likelihood. We will, thus, compare iterated filtering with a moderate number of particles to one-time filtering using a much larger number of particles.

In more detail, the methods applied in our Monte Carlo experiments are the following:

1. The Nelder-Mead or simplex approach is used to find the maximum of the likelihood function. The likelihood function itself will be approximated by a particle filter which leads to a lack of smoothness of the likelihood function because of simulation noise. We abstain from a presentation of the details of the Nelder-Mead algorithm which are available in every textbook on numerical optimization (e.g., Chong and Zak, 2011). We only note that we have used the following, relatively standard parametrization: reflection: 2.0, expansion: 1.0, contraction: 0.5, shrinkage: 0.5. Since the numerical approximation of the likelihood is non-smooth, we have always started the optimization with a small grid search for good initial values of the simplex algorithm. To this end, we have varied the crucial parameter m_0 (or m_1 , m_2 and m_3 in the bivariate and trivariate cases) over the interval $1.1, \dots 1.9$ with 9 equidistant steps, which in the bivariate case leads to 81 initial evaluations (729 in the trivariate case). Initial values of the parameters of the Brownian motions have been set equal to the standard deviation of the empirical series, and the off-diagonal entries of Σ in the bivariate and trivariate case equal to the sample correlations.
2. Iterated filtering I (IF I) follows the algorithm proposed in Ionides et al. (2006) for which further theoretical background was provided in Ionides et al. (2011). Both this method and the subsequent Iterated filtering II of Ionides et al. (2011) have been implemented in the R package `pomp` (King et al., 2016). We have, however, not used their software as it would probably have confronted us with limitations of computational speed that would not have allowed to conduct the present range of simulations. We have rather encoded independently all the numerical algorithms used in this paper in C.

Both IF I and IF II add the parameters to the hidden state variables of the problem (here: the asset price volatility) which are then subjected to a filtering operation alongside with the state variables. In more detail, the methods perform the following operations:

Let θ be the vector of parameters, M the number of iterations of the filter and T the length of the underlying time series. IF 1 then performs the following steps:

Algorithm: Iterated Filtering I

1. Select starting values θ_0 .
2. For iterations $m = 1, \dots, M$ perform the following steps:
 - 2.1. Perturb parameters at the beginning of each loop creating an initial parameter swarm with elements $\theta_{0,m}^{(i)}$, link each vector of particles for the parameters with another set of particles for the hidden variables, $x_{0,m}^{(i)}$, drawn from their unconditional distributions.
 - 2.2. For one filtering operation over the $t = 1, \dots, T$ realizations of the data perform the following steps:
 - 2.2.1. Iterate the particles for the hidden variables from $t-1$ to t according to their law of motion and perturb parameters obtaining a new parameter swarm $\theta_{t,m}$.
 - 2.2.2. Use the particle filter to obtain an approximation of the conditional density $f(x_{t+1,m}^{(i)} | x_{t,m}^{(i)}, \theta_{t,m}^{(i)})$ with $i = 1, \dots, I$ the elements of the swarm of particles.
 - 2.2.3. Resample the particles for the hidden variables and parameters using multinomial draws based on their relative likelihoods.
 - 2.2.4. Compute the filtering mean and variance of the parameters by reweighting them with their relative likelihood.
 - 2.2.5. Update step-wise the parameters by setting

$$\theta_m = \theta_{m-1} + V_{1,m} \sum_{t=1}^T V_{t,m}^{-1} (\theta_{t,m} - \theta_{t-1,m}). \quad (8)$$

In the last step θ_m is the starting value of the filtering process at iteration m , $\theta_{t,m}$ is the weighted average of the resampled parameter particles as computed in step 2.2.4, $V_{t,m}$ is the prediction variance of the swarm of particles, and $V_{1,m}$ is the variance of the swarm upon initialization at the beginning of iteration m in step 2.1. (cf. King et al., 2016, Algorithm 3, for more details).

While one of the virtues of this and the following IF 2 is seen in its relatively broad applicability without the need of any tailoring for particular applications (‘plug and play’), a number of parameters has, nevertheless, to be set by the user. First, convergence to the maximum likelihood estimator requires a decrease of the variability of the innovations to the parameter particles. Following Ionides et al. (2006, 2011) we used Normally distributed innovations multiplied by a factor $r_m \cdot D$ where r_m is geometrically decaying from 0.1 to 0.01 over 100 iterations and D is the predefined range of the relevant parameters. This number is chosen equal to the admissible range of those parameters that are bounded by their nature (m_0, m_1, m_2, ρ) and was chosen large equal to 5 for those without natural upper boundaries ($\sigma, \sigma_1, \sigma_2$). The initial perturbation in each iteration was multiplied by a factor 20 to achieve sufficient variability as recommended by Ionides et al. (2006). Variances $V_{1,m}$ and $V_{t,m}$ in eq. (8) were defined as univariate variances separately for each parameter. A vectorized version of this updating equation exhibited severe convergence problems and was, therefore, not pursued further. As an alternative to eq. (8) we also used simple averaging: $\theta_m = \frac{1}{T} \sum_t \theta_{t,m-1}$. Initial values were either drawn randomly from the support of the pertinent parameters (restricted to $[0, 5]$ for the variances).

Alternatively, σ in the univariate case, and all entries of the covariance matrix of the Brownian motions in the bivariate and trivariate cases were chosen initially as equal to their sample counterparts as with the Nelder-Mead approach.

We now move on to the more recently proposed second version of iterated filtering of Ionides et al. (2015):

Algorithm: Iterated Filtering II

1. Create an initial parameter swarm.
2. For iterations $m = 1, \dots, M$ perform the following steps:
 - 2.1. Perturb the elements of the swarm of particles at the beginning of each loop and link each vector from the set of particles, $\theta_{0,m}^{(i)}$, with another vector of particles for the hidden variables, $x_{0,m}^{(i)}$ drawn from their stationary distribution.
 - 2.2. For one filtering operation over the $t = 1, \dots, T$ realizations of the data perform the following steps:
 - 2.2.1. Iterate the particles for the hidden variables from $t - 1$ to t and perturb the particles for the parameters,
 - 2.2.2. Use the particle filter to obtain I approximations of the conditional density, $f(x_{t+1,m}^{(i)} | x_{t,m}^{(i)}, \theta_{t,m}^{(i)})$,
 - 2.2.3. Resample the vector of particles for parameters and hidden variables using multinomial draws based on their relative likelihood,
 - 2.2.4. At the end of each iteration, the current swarm of particles of the parameters, $\theta_{T,m}^{(i)}$, is simply used as the initial swarm $\theta_{0,m+1}^{(i)}$ of the next iteration.

Iterated filtering II is somewhat simpler in structure than IF I as the initial particle swarm is just subjected to $M \cdot T$ (iterations times length of time series) filtering and resample operations without the need to compute sample statistics as in steps 2.2.4 and 2.2.5 of IF I. The decisions to be made by the users are essentially the same as with IF I. First, the initial distribution has to be determined for the initialization of the particles for the parameters, and the distributions for the perturbations have to be determined. In our application, both the same initialization and the same distributions of the disturbances have been chosen in our implementations of IF I and IF II. Again, the same decay factor was used for the variance of disturbances in IF II. In contrast to IF I, however, no blow-up of their variance at the start of each iteration, i.e. for $\theta_{0,m}$ is required, but the new initialization is simply defined as $\theta_{0,m}^{(i)} = \theta_{T,m-1}^{(i)}$.

It is worthwhile to note that when setting $M = 1$, iterated filtering nests well-known online estimation algorithms as special case. In particular, IF I is close to a repeated application of the well known Liu-West filter (Liu and West, 2001) while IF II is an iterated version of the bootstrap filter proposed by Rios and Lopes (2013).

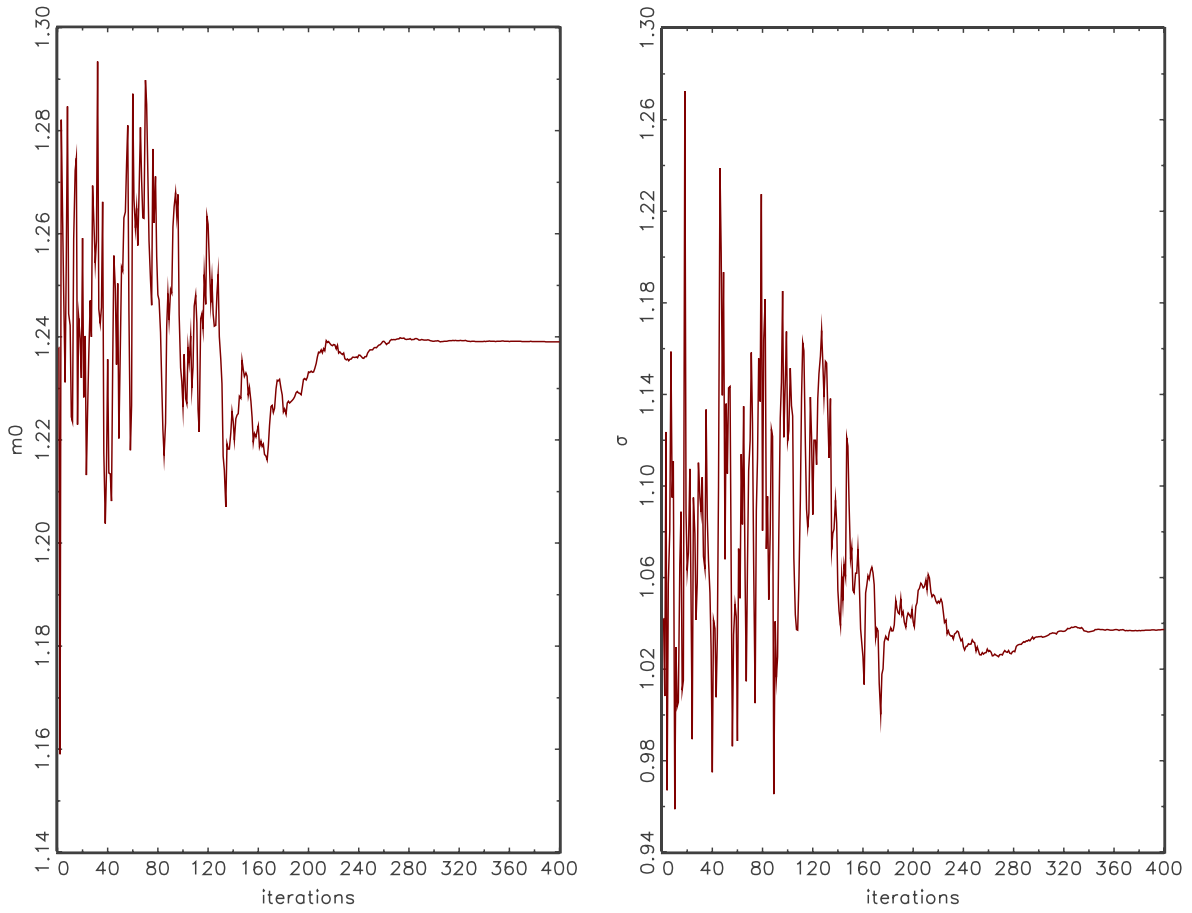


Figure 2: Illustration of iterative estimation via a repeated application of the same filtering algorithm over $M = 400$ iterations. The present case shows an application of IF 1 to the univariate discrete-time MSM with parameters $m_0 = 1.2$ and $\sigma = 1$.

4 Results of Monte Carlo simulations

4.1 Univariate discrete-time MSM

For the univariate MF model with predetermined transition probabilities as defined in eq. (2), the parameter set consists of the parameter m_0 of the Binomial distribution of the volatility components, and the scale factor σ . In the Monte Carlo simulations reported in Table 1, we have set $k = 8$, i.e. we consider a Markov-switching model with $2^8 = 256$ different states. In this and all subsequent cases we report the results of 100 Monte Carlo simulations using the same underlying simulated time series for all the estimation methods that we compare. Results are depicted by the mean, finite sample standard error (FSSE) and root-mean squared error (RMSE) across the 100 replications of each type of computational experiment. Discrete MSM models are amenable to maximum likelihood estimation so that we can establish a clear benchmark for this case. The first panel of Table 1 shows results for baseline ML estimation together with the pertinent statistics for Simulated Maximum Likelihood (SML) based on an approximation of the likelihood function using the particle filter for which we have attempted to find the maximum of the likelihood function via the Nelder-Mead or simplex algorithm. The underlying simulated time series had length $T = 1,000$, and the number of particles has been $B = 1,000$ and $B = 2,000$. We see that the numerical approximation gets on average very close to the results based on the exact analytical likelihood. Indeed, for $B = 2,000$, the accuracy of the estimates of the Binomial parameter m_0 is practically the same for ML and SML, and the accuracy of the estimates of σ is only slightly inferior with SML. Computationally, ML is much more efficient than simulated maximum likelihood in this case, but this relationship might change with increasing k .

One also notes that the gain from doubling the number of particles is tiny, while the computation time increases more than proportionally.¹ The later feature is mainly due to the multinomial sampling steps that have computational demands increasing more than proportionally with the number of particles. The remaining parts of the algorithms are, in contrast, all practically linear in computation time versus number of particles. It is also worthwhile to point out, that the disappointingly small increase of precision when moving from $B = 1,000$ to $B = 2,000$ is in no way cumbersome. Attainable precision is mainly determined by the amount of available data which is the same ($T = 1,000$) in all simulations reported in Table 1. For T constant, increasing only B need not necessarily lead to large improvements. It is known that a central limit theorem applies to the approximations of the likelihood when the number of particles goes to infinity (cf. Kantas et al., 2015), but one would certainly not expect better estimation results from SML than ML. Hence, the asymptotic limit appears to be almost reached already at $B = 2,000$ particles in the present case.

Moving on to alternative estimators, we first consider *online estimators* that provide parameter inference after only one sweep through the data (i.e., setting $M = 1$ in iterated filtering algorithms). These are denoted Filter 1 and Filter 2 (the later identical to the bootstrap filter proposed by Rios and Lopes, 2013). An even simpler approach is the so-called ‘self-organizing state space model’ (SOSM) proposed by Kitagawa (1998). In contrast to the other filters, SOSM does not perturb the vectors of particles for the

¹ Computation time refers to computations on Intel Xeon processors E5645 with 2.40 GHz. Some additional runs with AMD EPYC 7281 processors with 2.10 GHz showed a saving of computation time between 25 and 50 percent.

parameters, but just works via selection of the better fitting ones. We apply all three algorithms to our sample of 100 test series using 2,000, 10,000, 20,000, and 50,000 particles. Surprisingly, the least sophisticated algorithm, SOSM, seems the clear winner of this competition. At $B = 20,000$ it reaches the efficiency of SML (with $B = 1,000$) and at $B = 50,000$ it reaches SML (with $B = 2,000$). While there is only one iteration in SOSM compared to more than 100 in SML (the maximum number of iterations of the simplex imposed in our application), the larger number of particles comes with such a heavy computational burden that SML appears nevertheless computationally cheaper for a given level of accuracy.

As concerns the bootstrap filter and Filter 1 the computational demands of the more sophisticated algorithms are only very slightly higher than those of SOSM. However, both are almost uniformly inferior. Filter 1 remains much behind the other two in terms of precision of m_0 (arguably the more important parameter). The bootstrap filter (Filter 2) performs better than SOSM with $B = 2,000$ but with higher number of particles falls back behind the latter. Indeed, at least for parameter m_0 , the bootstrap and Filter 1 variants show practically no increase in accuracy between $B = 2,000$ and $B = 20,000$ and even a deterioration for $B = 50,000$.

The obvious next question is whether iterating the filtering operations helps to improve results. To shed light on the performance of the non-degenerate Iterated Filters 1 and 2, we have executed those over up to $M = 400$ iterations. We always find that the built-in decline of the variance of the disturbances guarantees convergence of the estimates at some point beyond about 100 and 200 iterations. As can be seen in the examples of Fig. 2, the estimates might undergo relatively wild variation initially. If the final estimates were more precise, $M = 1$ would, of course, not be a good choice, and we should expect a dismal performance of the online versions considered before.

We start with IF 2 or the Iterated Bootstrap Algorithm. As can be seen from the table, iterating helps and even with $B = 1,000$ generates better results than the online version with any number of parameters. Moving on to $B = 2,000$ provides a further improvement. The table shows estimates obtained after 100, 200, 300 and 400 rounds with practically no change happening any more beyond $M = 200$. At $M = 100$, we end up at a precision using $B = 2,000$ that ranges somewhere between that of the SML algorithm with 1,000 and 2,000 particles. Since computation time is linear in M , this precision costs about 290 seconds of computation time which is also in the range between the two implementations of SML. Hence, the performance of both methods in terms of efficiency and cost is about the same.

Moving finally on to IF 1, we have considered various versions of the later. First, the baseline version (plainly denoted IF 1 in Table 1) does not perform well. It is obviously inferior to IF 2. Our first modification consisted in replacing the original adjustment of the parameters according to eq. (8) by a simple averaging over the time dimension: $\theta_m = \frac{1}{T} \sum_t \theta_{t,m-1}$. The reason is that we conjectured that the numerical derivatives in eq. (8) can lead to instabilities. Indeed, when using eq. (8) not as single equation for each element of the vector of parameters (i.e. estimating $V_{t,m}$ as the full variance-covariance matrix over all parameters, and not as a diagonal matrix), we often observed explosive instabilities. As it turns out, "Iterated Filtering 1 with averaging" indeed performs better, with results only slightly behind those of IF 2. The outcome of another subtle change is shown in the remainder of Table 1. Another conjecture for the initial poor performance of IF 1 could be that with uniform draws of initial parameter estimates from $[1, 2]$ for m_0 and $[0, 5]$ for σ a large deviation of particularly the initial value of the scale factor

from its ‘true’ value might be hard to overcome. This problem is more severe for IF 1 as it starts with a specific *initial value* which is subsequently transformed into a swarm via stochastic perturbations, while IF 2 starts immediately with an entire swarm. One could argue that the large variance of the initialization of the disturbances should provide a strong randomization force but still the Normal perturbations could have as their mean a value potentially far from the true one. We, thus, also experimented with an initialization using the empirical standard deviation of the data which is a plausible a priori estimate. Results for IF 1 in its original form indeed improve tremendously and are getting close to those of the SML estimates with $B = 2,000$. Indeed, practically the same precision was obtained for this version with $B = 1,000$ and $B = 2,000$. Using $M = 100$ would lead to an average cost of about 90 seconds for this version with $B = 1,000$ so that it would appear more cost-efficient than SML. For “IF 1 with averaging”, in contrast, no improvements could be obtained with fixing the initial value of σ in this way.

In conclusion, we find that Iterated Filtering 2 attains about the same profile of accuracy and computational cost as SML with Nelder-Mead optimization. IF 1 can be somewhat more efficient, but its superiority hinges on specific details of its implementation.

4.2 Univariate continuous-time MSM

The results of the previous section are confirmed to a large part by our experiments with the continuous-time univariate MSM. For variation, we have chosen a higher multifractal parameter, $m_0 = 1.4$, in this case. Since no analytical solution for the likelihood function is available for the continuous-time model, Table 2 starts with SML. While, of course, the results are not directly comparable to those of Table 1 because of the different nature of the process, the details confirm previous insights (cf., e.g., Lux, 2008): A higher multifractal parameter can be estimated with more precision, but impacts negatively on the accuracy of the estimates of the scale factor, σ , due to the higher degree of heterogeneity of the variance dynamics.

In the evaluation of the online algorithms, we have dispensed with the largest particle size $B = 50,000$, because of the immense computational burden that comes with it. For SOSM and the bootstrap filter, we see that their accuracy is somewhat worse than that of SML. As with the discrete model, accuracy might approach that of SML when moving to even higher numbers of particles, but this would then again be achieved at a much higher cost than with SML. Quite in contrast to our results in the previous sub-section, the online Filter 1 performs better even than SML already with $B = 2,000$ particles and it improves further when increasing B .

Moving on to the iterated filters, there is a certain difference to the results for the discrete version in that IF 2 does not really improve upon the bootstrap filter. Indeed, IF 2 with $B = 2,000$ yields practically the same accuracy like the bootstrap filter with $B = 20,000$ while the later is definitely cheaper in computation time. The IF 1 algorithm shows the following tendencies: Its original version gets close to SML for parameter m_0 , but displays much higher variability for σ . It seems natural that this deficiency can be remedied when choosing the initial value for the second parameter in a data-driven way rather than randomly from a wide interval. The resulting estimates outperform SML already for $B = 1,000$ and improve slightly when increasing the number of particles to $B = 2,000$. IF 1 with averaging over the sample, in contrast, remains inferior to this version and of about the same quality as IF 2. The main surprise is, however, that even

the best performing version of IF 1 leads to a clear deterioration of results compared to its online counterpart which in the present case provides the unambiguously best and also the cheapest estimates.²

4.3 Bivariate continuous-time MSM

Since the bivariate model proposed in sec. 2 is the first continuous-time version of a multivariate multifractal model in extant literature, this is also the first time that estimation of such a model is explored. Similarly as with its univariate counterpart, analytical maximum likelihood is infeasible because of the lack of a closed form solution to the likelihood function. Hence, we can only work with numerical approximations of the likelihood and the same algorithms as before are compared as alternative avenues to determine the maximum of the likelihood function.

Table 3 provides the results. The underlying parameters are $k_1 = 5$, $k = 10$, $m_1 = 1.4$, $m_2 = 1.6$, $\sigma_1 = \sigma_2 = 1$ and $\rho = 0.4$. Our main interest is to see whether the increase of the parameter space from two parameters in the univariate case to five for the bivariate model changes the performance of the various algorithms, and whether their relative ranking in terms of precision and computation time remains the same or not under this new model.

First, starting with simulated ML with the Nelder-Mead optimization algorithm, we find that (i) the multifractal parameters m_1 and m_2 , are estimated with the same or even slightly higher precision, (ii) the scale parameters σ_1 and σ_2 , are estimated with somewhat lower precision, (iii) the correlation coefficient ρ for which no counterpart in the univariate case exists, seems to have a satisfactory signal-to-noise ratio, (iv) overall, results improve when moving from $B = 1,000$ to $B = 2,000$ but only by a small margin, (v) in comparison to the univariate case, the computational demands with the same lengths of the time series and number of particles are increasing *less* than proportionally in reaction to the higher number of parameters. Overall, the traditional SML approach appears, therefore, to provide a promising avenue also in the multivariate case.

Next, we inspect the results for the three online estimators with the number of particles $B = 2,000$, $B = 10,000$ and $B = 20,000$. For two of these algorithms, the SOSM and the bootstrap filter, overall results are definitely worse than those of the SML estimates and in relation to the results for the univariate models shown before, they all fall more clearly behind the SML estimates. An exception is the online Filter 1. This filter even with $B = 2,000$ already gets close to the performance of SML with $B = 1,000$. For $B = 10,000$ and beyond, online IF 1 outperforms SML for four out of five parameters which is reminiscent of the findings for the univariate continuous-time model. It is also interesting to note that for the online algorithms and their iterated versions, computation time is almost independent of the number of parameters. This is so because all the major computational components remain the same as in the univariate case. At $B = 20,000$, for instance the computation time of the online algorithms remains uniformly below that of SML with $B = 1,000$ particles.

For the iterated filters, we report parameter estimates after $M = 200$ and $M = 300$ rounds. The comparison of these numbers shows that convergence has taken place in all cases at such a number of repetitions of the filtering algorithms. For IF 1, we initiate the particle swarm again by random draws of the initial values of m_1 and m_2 from the admissible interval $[1, 2]$, and set σ_1 , σ_2 and ρ equal to the sample standard deviation and

² Because of this unexpected outcome, results have been checked multiple times but have always confirmed the pattern of Table 2.

correlation between both series. Since initialization with random draws for the parameters of the covariance matrix always turned out to be inferior so far, we do not pursue this alternative anymore. For IF 2, the initial swarm is chosen from uniform distributions: $m_1, m_2 \in [1, 2]$, $\sigma_1, \sigma_2 \in [0, 5]$ and $\rho \in [-1, 1]$.

Computation time (taking into account that only 300 iterations have been performed) increases again only very slightly against the univariate case. Overall, with convergence being obtained after $M = 200$ iterations as the latest, the computational demands of SML and the iterated filter are roughly comparable. Overall, however, we find that iteration does hardly improve the performance in comparison to the online execution of the same filters.

For the iterated bootstrap filter (IF 2) and IF 1 with averaging (IF1av), the efficiency of the estimates clearly falls behind that of SML. It is noteworthy that IF 2 has particular problems in estimating σ_1 and σ_2 . The only serious competitor is the original version of IF 1 (with data-dependent choices of starting values for σ_1 , σ_2 and ρ) which for some parameters improves upon SML. However, in contrast to its online counterpart, we also see gains and losses in precision for different parameters. Overall, iterating the filtering operations does not yield a clear improvement, and, thus, we end up with a similar conclusion as for the univariate case: online Filter 1 provides the best and cheapest way for identification of the parameters.

4.4 Trivariate continuous-time MSM

We have seen in the preceding sections that more or less similar precision and efficiency can be attained with the Nelder-Mead or simplex optimization algorithm and iterated filtering, at least when the latter approach is designed appropriately using relevant information on good starting values. Since with higher numbers of parameters, the computational burden of the simplex method increases faster than that of online or iterated filters, it appears interesting to go one step further and explore the efficiency of the various methods when estimating a model with still more parameters. For that purpose, we consider the trivariate version of the continuous-time MSM following the same principles of construction like the previous bivariate version. We then have 9 parameters to estimate: the three multifractal coefficients m_1 , m_2 and m_3 , and the variance and covariances of the three Brownian motions. We denote the parameters of the variance-covariance matrix as σ_1 , σ_2 , σ_3 , ρ_{12} , ρ_{23} and ρ_{13} , with obvious meanings.

Table 4 shows results for various estimators of a trivariate MSM with the following parametrization: $k_1 = 8$, $k = 15$, $m_1 = 1.2$, $m_2 = 1.4$, $m_3 = 1.6$, $\sigma_1 = \sigma_2 = \sigma_3 = 1$, $\rho_{12} = 0.4$, $\rho_{23} = 0.7$ and $\rho_{13} = 0.2$. We have skipped here the SOSM filter because of its unsatisfactory performance in the previous applications and only report the final outcome of the iterated filters after $M = 300$ iterations. As we can see in Table 4, the computational demands of SML are increasing by a factor 3 when moving from the five parameters of the bivariate case to the nine parameters of the trivariate case. In contrast, the computation time of the various filters increases by less than 20 percent (of which part is due to the higher number of multifractal multipliers k). However, none of the online and iterated filters reaches the performance of SML. The online bootstrap filter shows certain improvements of accuracy when increasing the number of particles, but less so under iterated application (IF 2). It performs particularly poorly in identifying the variances of the Brownian motion. While this weakness has already transpired with the bivariate model, it appears even more pronounced for the trivariate model. Noting

that the expectation and standard deviation of the initial swarm of particles are 2.5 and 1.443, respectively, it seems that both filter 2 and IF 2 are less and less able to move away from the broad initial distribution when the parameter space increases (note that their univariate applications did not suffer from this deficiency).

The online Filter 1 starts out better in terms of variances and much worse in terms of multifractal parameters than Filter 2. However, it appears almost insensitive to the number of particles. It improves, in contrast, under iterated application, but only if estimates are averaged over the samples. The latter version is the only one that seems a serious competitor to SML with some parameters of lower accuracy and some others with higher precision across our 100 Monte Carlo replications.

4.5 Models with more Markov states

We proceed by investigating the performance of sequential Monte Carlo estimators when increasing the number of states of the Markov chain further.

Table 5 provides results for the discrete-time and continuous time univariate MSM with up to $k = 15$ hierarchical levels and for the continuous-time bivariate model with $k = 15$ levels. Note that particularly in the later case, the estimated model has as many as $4^{15} \approx 10^9$ states. We keep the other parameters constant at their values in the previous Monte Carlo simulations, and we confine ourselves to the estimation algorithm that had displayed the most consistent behavior in the previous subsections, the Nelder-Mead or simplex approach (denoted by SML). All the examples shown in Table 5 exhibit the following characteristics: (i) the estimates of the multifractal parameters m_0, m_1 and m_2 show hardly any change compared to the previous cases with smaller k , i.e., the precision of their estimation remains practically insensitive to the increase in the number of Markov states, (ii) for the scale parameter σ , RMSEs are all increasing with k which, however, is expected as it becomes increasingly difficult for the estimators to distinguish between very long-lived volatility components and a constant factor (cf. Lux, 2008), (iii) except for the variances in the bivariate case, all parameters can be estimated with very little biases (as indicated by the proximity of FSSEs and RMSEs), (iv) the correlation parameter ρ in the bivariate case appears also to be very little affected by the higher number of states, (v) already with $B = 1,000$ the SML estimators in the discrete case are again quite close to those obtained from exact maximum likelihood, and overall, there is hardly any clear gain from increasing the number of particles from $B = 1,000$ to $B = 2,000$, (vi) the computation time increases only slightly for SML with increasing k , while we see that at $k = 12$ we approach the boundary of feasible full maximum likelihood estimation.

Overall, the results of Table 5 indicate that the sheer number of Markov states does not necessarily constitute an obstacle to efficient estimation via sequential Monte Carlo even though the number of particles in our application remains very small in comparison to the number of states. This convenient lack of sensitivity is certainly owed to the fact that the multifractal models allow for a large number of states without a proliferation of the number of parameters.

5 Applications

In the following, we apply sequential Monte Carlo estimation to a selection of financial returns of different asset classes. We estimate the discrete-time and continuous time MSM and compare their in-sample and out-of-sample fit vis-à-vis each other and against

standard GARCH and FIGARCH models. While we use the Nelder–Mead algorithm for this analysis, we subsequently will also shed some light on how different algorithms affect the precision of in-sample and out-of-sample results in selected cases. Applications of multivariate MSM models are left for future research.

In order to get a broad perspective of the performance of the competing models, we consider in the following: three stock market indices, the German DAX, the S&P 500 and the Japanese Nikkei, all extracted as daily data from Datastream over the in-sample period 01/01/1980 to 12/31/2004. The same in-sample period is used for the price of gold as an important precious metal that is considered to be at least as volatile as stock indices. As another class of financial data, we include the exchange rates of some major currencies: the US dollar against the euro with in-sample data spanning the period 01/01/1999 to 12/31/2009, the Swiss franc - euro rate from 7/15/2003 through 12/31/2009 and the Japanese yen - dollar rate from 01/01/1986 to 21/31/2004. For the out-of-sample analysis we use all data after the respective in-sample period through to the end of the month 02/2015.³

Table 6 presents the in-sample parameter estimates of discrete-time and continuous-time MSM models with $k = 10$ hierarchical levels for these seven time series. We can observe that generally the parameter estimates and the likelihood values of both models are very close to each other. The crucial parameter m_0 hovers within a relatively narrow range between 1.2 and 1.4 so that the estimated models are all relatively similar (as it is typically found for GARCH models as well).

Table 7 shows formal test results for better fit of any of the two models (denoting the discrete model by MF and the continuous-time, Poisson multifractal model by PMF) against each other and against the seminal GARCH and FIGARCH models.⁴ Since in all pairwise comparisons, the two competitors are non-nested, we apply the model selection test of Vuong for non-nested alternatives (Vuong, 1989). In the comparisons with GARCH and FIGARCH, we also document the outcome of the adjusted Vuong test taking into account the larger number of parameters of the GARCH and FIGARCH models.

Table 7 conveys the following information: except for the exchange rates. USD/EUR and CHF/EUR, the two multifractal models are always diagnosed as superior to GARCH and FIGARCH at standard levels of significance. For those two exchange rates, the results of the baseline Vuong test are not conclusive in either direction, but when taking into account the number of parameters to be estimated with the different competitors, MF and PMF are again diagnosed as superior. We might not expect the relatively small likelihood differences of MF and PMF to be significantly different under a formal test. However, the Vuong test indicates a significantly different fit in four out of seven cases at least at the 5 percent level (at the one percent level for three). All those cases indicate a significantly better fit of the discrete-time model. To see in how far the out-of-sample performance confirms these results, we compute mean squared errors (MSE) and mean absolute errors (MAE) of the forecasts from the four alternative models over various horizons. Table 8 exhibits these relative to the MSE/MAE of a naïve forecast using historical volatility. Also shown is the outcome of tests for superior predictive ability of Hansen (2005). This test compares the relative performance of a particular model

³ The same data sets have been used by Ghonghadze and Lux (2016) and Chen and Lux (2017) to scrutinize the performance of an alternative class of asset-pricing models. A comparison with their results shows that the MSM models perform throughout better than the behavioral models that are the focus of these studies.

⁴ Estimated in their (1,1) specifications. Results are in agreement with typical parameter estimates of GARCH(1,1) and FIGARCH(1,1) models in the literature.

against a group of competitors. In the present setting, the competing models are the remaining three time series models as well as historical volatility. The tested null is that the pertinent model is not outperformed by any of its competitors.

The out-of-sample results are generally favorable for the MSM class of models:

- for the DAX, the continuous-time PMF dominates all other models and is unambiguously classified as the superior model among the four candidates under the MAE criterion. Under the MSE criterion, GARCH, FIGARCH and MF are all not significantly outperformed by other models for horizons up to 40 days. However, when moving from shorter to longer horizons, the relative performance of MF vis-à-vis these competition improves more and more, and for the longest horizons of fifty days it outperforms all alternative models. PMF, in contrast, is always dominated by the alternatives under the MSE criterion.
- for the S&P 500 a similar pattern applies: GARCH and FIGARCH dominate the MSM family under MSE for short horizons, but MF becomes a stronger competition for longer forecast horizons. Under MAE, the continuous-time PMF beats all alternatives at all levels of significance for all time horizons.
- for NIKKEI, we also find GARCH and FIGARCH again performing better than MF and PMF for horizons up to 10 days, and thereafter superiority of MF alone under MSE. Under the MAE criterion, GARCH and FIGARCH perform worse than historical volatility throughout, while PMF is the only model to provide a gain in forecast precision at least up to 20 period forecast horizons,
- the EUR/USD exchange rate is a stronghold of the GARCH model. It outperforms all other models under MAE. Under MSE it also provides the most accurate forecasts, but for horizons above one day MF and PMF are close followers and can not be significantly outperformed.
- for the YEN/USD, we find PMF to perform very strongly under both the MSE and MAE criterion. While the null hypothesis that the pertinent model cannot be outperformed is not rejected for other models at certain forecast horizons as well, the results are most uniform for PMF with acceptance probabilities of the null hypothesis (the model cannot be outperformed) close to one in almost all cases.
- for the CHF/EUR, under MSE all models hardly perform better than historical volatility with the relative best being MF, while under MAE PMF dominates all competitors, and at all horizons. Note that while it appears generally difficult to forecast volatility of the CHF/EUR rate, the multifractal models both provide a slight improvement against historical volatility under both MSE and MAE while GARCH and FIGARCH generate forecasts that are uniformly inferior to historical volatility.
- finally, for the price of gold, we find no clear tendency under the MSE criterion, and a clear dominance of FIGARCH under MAE. Under MSE, none of our four models can be outperformed at any horizon beyond one day, but MF and PMF exhibit the smallest errors at longer horizons.

Overall, the results of the out-of-sample forecast comparisons differ from the in-sample comparisons in various ways: First, the in-sample dominance of multifractal models in

terms of goodness-of-fit is only partially reflected in superior out-of-sample performance. Second, a clear dominance of PMF is found for the MAE criterion in four out of seven cases, over all forecasting horizons. In all these cases (among those all three stock markets considered in this study) PMF not only dominates over GARCH and FIGARCH, but also over its discrete-time counterpart. This is in contrast to the in-sample results in which MF has been found to have higher goodness-of-fit than PMF. Under the MSE criterion, results are mostly inconclusive or in favor of GARCH/FIGARCH at short forecasting horizons, but become more favorable for MF or PMW at longer horizons. Given that GARCH and FIGARCH have particular parameters geared towards short-term correlations that are absent in the multifractal formalism such dependency on the forecast horizon appears plausible (although it does mostly not apply under the MAE criterion).

One may note that while the estimation of the GARCH and FIGARCH models is a deterministic process, the use of the sequential Monte Carlo approach makes the outcome of the estimation of the MF and PMF models stochastic. With different random seeds of the particles, a different outcome would be obtained. To assess the influence of noise in the stochastic optimization algorithm, the estimation and forecast generation over one period has been replicated twenty times for SML and the various filters and iterated filters of the previous sections. Figs. 3 and 4 display the results for the case of the DAX. Fig. 3 shows the dispersion of the resulting estimates for m_0 and σ over the six algorithms together with the unique estimates from exact ML indicated by dashed lines. It transpires that SML and IF 1 generate estimates closest to the ML parameters and with the smallest degree of dispersion. Particularly poor appear the estimates from Filter 2 and from SOSM (provided one considers the ML estimates as optimal ones).

How does this translate into out-of-sample forecasts? Fig. 4 shows MSEs and AMEs of all 120 estimations relative to those of historical volatility. Interestingly, results are hovering around those obtained with full maximum likelihood. The large deviation of the parameter estimates under the SOSM algorithm from the remaining ones also translates into large deviations of its MSEs/MAEs out-of-sample. To rank the alternatives seems hard to do as their performance under MSE and MAE seems to be negatively correlated. The correlation between both measures is indeed -0.953 for the pooled 120 estimations. In contrast, a high positive correlation of $+0.653$ applies to the estimates of m_0 and σ depicted in Fig. 3. The stochastic approximation of the likelihood and the stochastic evolutionary algorithms applied here all have a tendency to “compensate” for deviations of one parameter by deviations of the other in the same direction (compared to the ML estimates). As demonstrated particularly by the results of the SOSM algorithm, a certain trade-off exists between a good fit in terms of MSE or MAE which indicates that different parameter values would be optimal under different loss criteria. This suggests that further gains of precision could be obtained with appropriate extensions of the baseline multifractal models.

Similar exercises for the other six time series provide results quite close to those for the DAX as displayed in Figs. 3 and 4. In particular, SML and IF 1 almost always show the smallest variation in parameter estimates and the closest proximity to estimates obtained by full maximum likelihood. These two methods also get closest to exact ML with their implied mean squared and mean absolute errors of the out-of-sample forecasts. Pertinent results for the other six financial time series are available upon request.

6 Conclusion

The present paper has addressed a number of issues. First, our main interest was in the performance of different optimization algorithms in locating the parameters maximizing the likelihood function of a class of highly nonlinear, non-Gaussian state-space models. As it turned out, by and large the time-honored Nelder-Mead or simplex method proved itself still to be the most efficient and robust approach vis-à-vis a selection of recently proposed online and iterated filters. For problems with not too many parameters, the iterated filtering algorithm proposed by Ionides et al. (2006) appeared very close both in precision and computational demands, while the somewhat more straightforward, so-called iterated Filter 2 of Ionides et al. (2010) appeared inferior to the former alternatives. Non-iterated, online filters often turned out to be less efficient or too computation intense when trying to compensate for their one-sweep nature by increasing the number of particles (with the noteworthy exception of Filter 1 in some cases). When increasing the number of parameters (as in trivariate MSM with nine parameters), all other methods seemed to remain behind Nelder-Mead. One might argue that with a higher-dimensional parameter space, one should allow for a longer relaxation time of the iterated filters. However, experiments with a slower decay of the disturbances to the parameters did not yield any immediate improvement in efficiency. Nevertheless, variations of the decay pattern and other subtle changes to the algorithms might still be worthwhile to explore in future research.

In terms of the underlying class of state-space models, the present paper has also extended the zoo of MSM models to multivariate continuous-time specifications, and it has provided the first generally applicable avenue towards estimation of continuous-time (Poisson) multifractal models. This is important since continuous-time models are preferred in applications in financial engineering, and continuous-time multivariate multifractal models might also prove useful in other areas of application of the multifractal apparatus. As our empirical application has demonstrated, the Poisson MSM has even turned out more successful in out-of-sample forecasting of volatility than its discrete-time counterpart, and in the majority of cases it outperformed all alternatives that we considered in this paper under the mean absolute error criterion over all forecast horizons. This successful application of the univariate continuous-time MSM should motivate empirical applications of the multivariate specifications as well. We believe this should be an important object for future research.

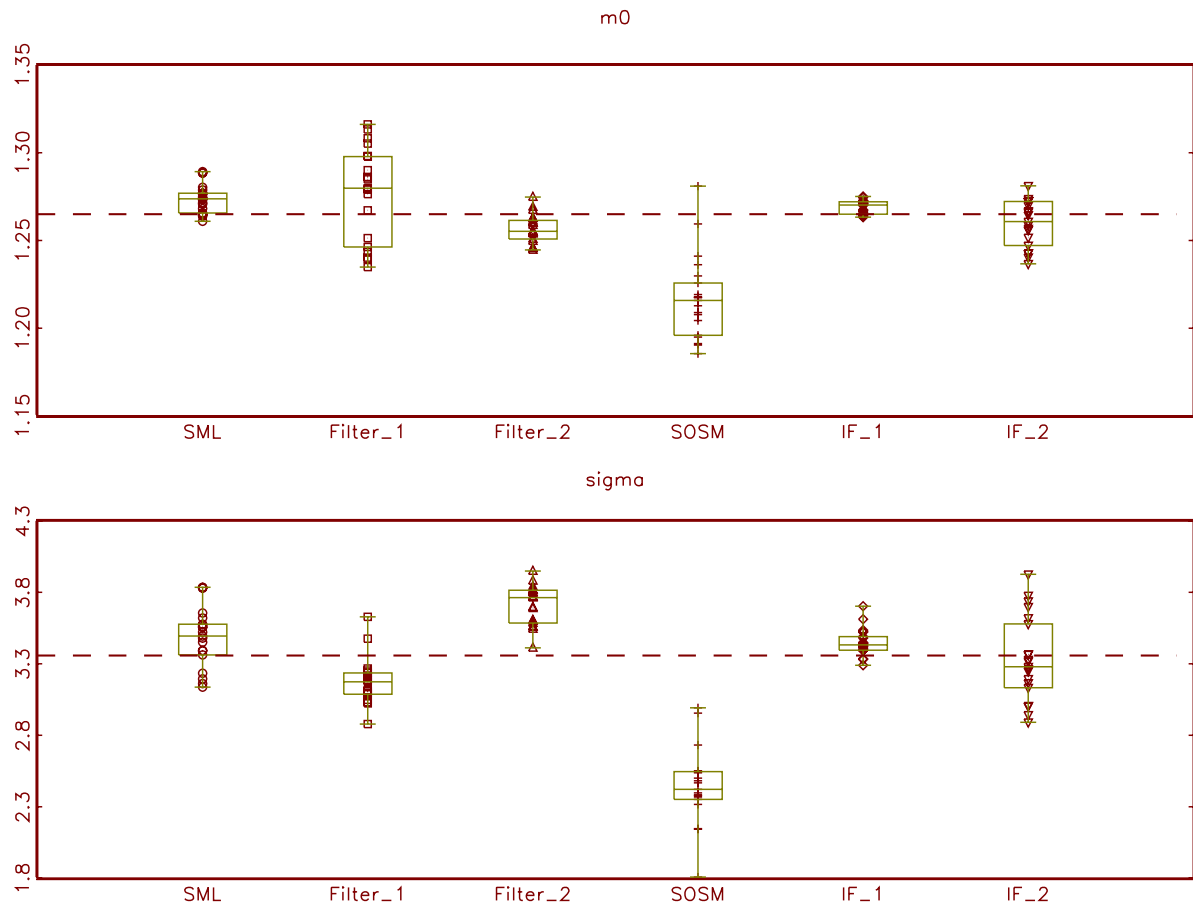


Figure 3: Boxplot of parameter estimates of the discrete-time MF model with different estimation algorithms, each repeated twenty times with different starting values. The underlying time series is returns of the German DAX. The broken lines indicate the parameter estimates obtained by exact maximum likelihood.

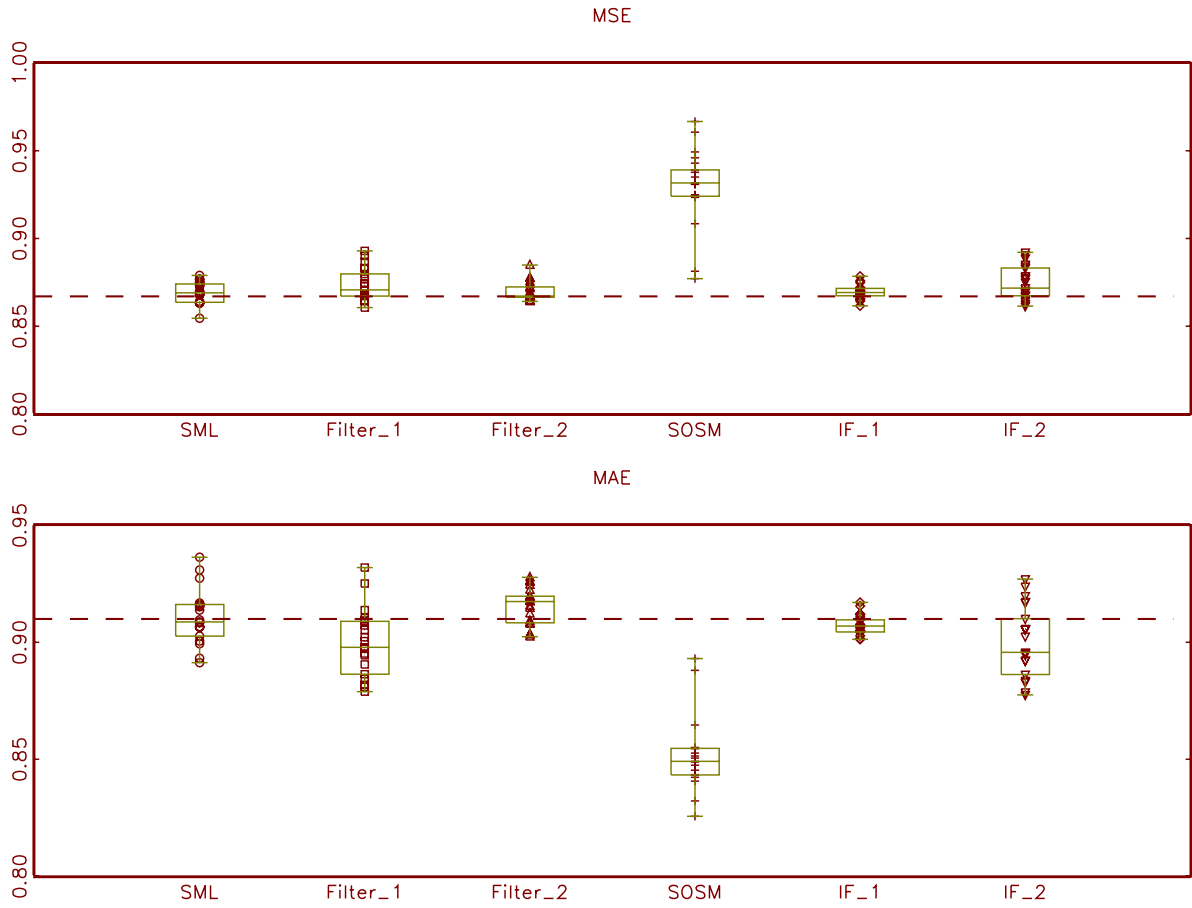


Figure 4: Boxplot of mean squared errors (MSE) and mean absolute errors (MAE) of out-of-sample volatility forecasts on the base of the 120 estimations of the parameters of the discrete MF model for the German DAX illustrated in Fig. 3. The broken lines indicate the MSE and MAE of volatility forecasts on the base of parameter estimates obtained by exact maximum likelihood.

Table 1: Estimation of discrete-time univariate MSM Model

	ML			SML, $B = 1,000$			SML, $B = 2,000$			m_0	σ	t [sec]
	m_0	σ	t [sec]	m_0	σ	t [sec]	m_0	σ	t [sec]			
Mean	1.192	0.982	7.590	1.195	1.009	176.960	1.197	1.005	597.330			
FSSE	0.032	0.067	0.753	0.036	0.086	6.434	0.033	0.076	23.948			
RMSE	0.033	0.069		0.036	0.086		0.033	0.076				
	SOSM, $B = 2000$			$B = 10,000$			$B = 20,000$			$B = 50,000$		
Mean	1.210	1.020	2.690	1.207	1.010	57.370	1.201	0.998	224.390	1.203	1.000	1,388.230
FSSE	0.064	0.176	0.465	0.041	0.095	0.917	0.037	0.081	3.402	0.033	0.079	19.833
RMSE	0.064	0.176		0.041	0.095		0.037	0.080		0.033	0.079	
	Filter 1			$B = 10,000$			$B = 20,000$			$B = 50,000$		
Mean	1.385	1.084	2.950	1.371	1.068	61.730	1.368	1.063	241.260	1.519	0.981	1434.290
FSSE	0.180	0.127	0.219	0.167	0.108	1.262	0.164	0.097	1.643	0.293	0.070	16.570
RMSE	0.258	0.152		0.238	0.127		0.234	0.115		0.432	0.072	
	Filter 2 (Bootstrap Filter)			$B = 10,000$			$B = 20,000$			$B = 50,000$		
Mean	1.181	1.028	2.930	1.178	1.012	61.600	1.178	1.021	241.150	1.177	1.020	1434.410
FSSE	0.050	0.130	0.293	0.047	0.130	0.569	0.045	0.126	1.445	0.045	0.127	17.640
RMSE	0.053	0.132		0.052	0.130		0.050	0.127		0.051	0.128	

Note: The table shows the results of 100 Monte Carlo runs of various estimators, for different numbers of particles B , using the same random numbers for the underlying simulated time series. The underlying parameters of the MSM model are $k = 8$, $m_0 = 1.2$, and $\sigma = 1$. Results are shown as the means, finite sample standard errors (FSSE) and root-mean squared errors (RMSE) of parameter estimates over the 100 Monte Carlo runs. t [sec] is the computation time in seconds per Monte Carlo replication.

Table 1: continued

	m_0 [100]	m_0 [200]	m_0 [300]	m_0 [400]	σ [100]	σ [200]	σ [300]	σ [400]	t [sec]
Iterated Filter 1									
$B = 1,000$									
MSE	1.296	1.295	1.295	1.295	1.642	1.633	1.633	1.633	348.170
FSSE	0.106	0.106	0.106	0.106	0.909	0.904	0.904	0.904	3.485
RMSE	0.143	0.141	0.141	0.141	1.109	1.100	1.100	1.100	
$B = 2,000$									
MSE	1.294	1.293	1.293	1.293	1.619	1.609	1.609	1.609	1163.980
FSSE	0.102	0.102	0.102	0.102	0.893	0.886	0.886	0.886	11.590
RMSE	0.139	0.138	0.138	0.138	1.082	1.072	1.071	1.071	
Iterated Filter 1, starting condition for σ fixed									
$B = 1,000$									
Mean	1.209	1.208	1.208	1.208	1.006	1.002	1.003	1.003	347.690
FSSE	0.034	0.033	0.033	0.033	0.079	0.078	0.078	0.078	2.116
RMSE	0.035	0.034	0.034	0.034	0.079	0.077	0.077	0.077	
$B = 2,000$									
Mean	1.213	1.212	1.212	1.212	1.005	1.003	1.003	1.003	1166.240
FSSE	0.033	0.033	0.033	0.033	0.076	0.076	0.076	0.076	5.364
RMSE	0.035	0.035	0.035	0.035	0.076	0.075	0.076	0.076	
Iterated Filter 1av									
$B = 1,000$									
Mean	1.194	1.197	1.197	1.197	0.990	0.993	0.992	0.992	349.140
FSSE	0.041	0.040	0.040	0.040	0.107	0.097	0.096	0.096	3.188
RMSE	0.041	0.040	0.040	0.040	0.107	0.097	0.096	0.096	
$B = 2,000$									
Mean	1.192	1.192	1.192	1.192	0.991	0.989	0.989	0.989	1166.310
FSSE	0.038	0.037	0.037	0.037	0.083	0.087	0.088	0.088	10.585
RMSE	0.038	0.037	0.037	0.037	0.083	0.087	0.088	0.088	
Iterated Filter 1av, starting condition for σ fixed									
$B = 1000$									
Mean	1.193	1.195	1.195	1.195	0.987	0.985	0.986	0.985	347.460
FSSE	0.042	0.039	0.039	0.039	0.087	0.089	0.089	0.089	2.072
RMSE	0.042	0.039	0.039	0.039	0.088	0.089	0.090	0.090	
$B = 2,000$									
Mean	1.191	1.193	1.193	1.193	0.996	0.993	0.992	0.992	1166.030
FSSE	0.039	0.038	0.038	0.038	0.091	0.089	0.089	0.089	6.031
RMSE	0.040	0.039	0.039	0.039	0.091	0.089	0.089	0.089	
Iterated Filter 2									
$B = 1,000$									
Mean	1.193	1.195	1.194	1.194	0.991	0.997	0.997	0.997	344.110
FSSE	0.041	0.042	0.042	0.042	0.098	0.090	0.091	0.090	3.744
RMSE	0.041	0.042	0.042	0.042	0.098	0.090	0.090	0.090	
$B = 2,000$									
Mean	1.193	1.196	1.195	1.195	0.993	0.987	0.987	0.987	1157.680
FSSE	0.037	0.039	0.039	0.039	0.073	0.080	0.081	0.081	10.459
RMSE	0.037	0.039	0.039	0.039	0.073	0.081	0.082	0.082	

Note: The table shows the results of 100 Monte Carlo runs of various estimators, for different numbers of particles B , using the same random numbers for the underlying simulated time series. The underlying parameters of the MF model are $k = 8$, $m_0 = 1.2$, and $\sigma = 1$. Results are shown as the means, finite sample standard errors (FSSE) and root-mean squared errors (RMSE) of parameter estimates over the 100 Monte Carlo runs. t [sec] is the computation time in seconds per Monte Carlo replication.

Table 2: Estimation of continuous-time univariate MSM Model

	SML, $B = 1,000$			SML, $B = 2,000$			m_0	σ	t [sec]
	m_0	σ	t [sec]	m_0	σ	t [sec]			
Mean	1.390	1.091	197.540	1.391	1.081	643.440			
FSSE	0.033	0.178	8.389	0.031	0.160	22.734			
RMSE	0.034	0.199		0.032	0.179				
	SOSM, $B = 2000$			$B = 10,000$			$B = 20,000$		
Mean	1.387	1.102	3.090	1.386	1.091	62.570	1.388	1.082	241.930
FSSE	0.069	0.304	0.288	0.040	0.229	0.807	0.031	0.195	3.391
RMSE	0.070	0.319		0.042	0.245		0.034	0.211	
	Filter 1 $B = 2000$			$B = 10,000$			$B = 20,000$		
Mean	1.410	1.048	3.240	1.408	1.043	63.220	1.408	1.042	243.480
FSSE	0.022	0.145	0.429	0.017	0.130	0.836	0.017	0.124	3.043
RMSE	0.024	0.152		0.019	0.136		0.018	0.131	
	Filter 2 $B = 2000$			$B = 10,000$			$B = 20,000$		
Mean	1.395	1.127	3.160	1.390	1.103	62.670	1.390	1.090	242.200
FSSE	0.046	0.258	0.368	0.040	0.190	0.911	0.035	0.180	3.094
RMSE	0.046	0.287		0.041	0.216		0.036	0.200	

Table 2: continued

	Iterated Filter 1 B = 1,000					B = 2,000				
	m_0 [300]	m_0 [400]	σ [300]	σ [400]	t [sec]	m_0 [300]	m_0 [400]	σ [300]	σ [400]	t [sec]
Mean	1.424	1.424	1.358	1.358	313.230	1.424	1.424	1.376	1.376	978.180
FSSE	0.028	0.028	0.194	0.194	3.646	0.029	0.029	0.192	0.192	10.671
RMSE	0.036	0.036	0.406	0.406		0.037	0.037	0.422	0.422	
	Iterated Filter 1, starting condition for σ fixed B = 1,000									
Mean	1.386	1.386	1.055	1.054	315.100	1.384	1.384	1.041	1.041	982.400
FSSE	0.029	0.029	0.157	0.157	7.283	0.027	0.027	0.138	0.138	11.183
RMSE	0.032	0.032	0.166	0.166		0.031	0.031	0.143	0.143	
	Iterated Filter 1av B = 1,000									
Mean	1.383	1.383	1.034	1.034	313.320	1.381	1.381	1.024	1.025	978.620
FSSE	0.040	0.040	0.179	0.180	3.556	0.037	0.037	0.162	0.162	12.969
RMSE	0.044	0.044	0.182	0.183		0.041	0.041	0.163	0.163	
	Iterated Filter 1av, starting condition for σ fixed B = 1000									
Mean	1.385	1.385	1.040	1.040	314.380	1.383	1.383	1.032	1.030	981.940
FSSE	0.039	0.039	0.187	0.186	3.687	0.035	0.035	0.168	0.168	11.999
RMSE	0.042	0.042	0.191	0.190		0.038	0.038	0.170	0.169	
	Iterated Filter 2 B = 1,000									
Mean	1.390	1.390	1.066	1.061	298.800	1.382	1.383	1.048	1.049	947.480
FSSE	0.041	0.041	0.215	0.215	1.758	0.037	0.037	0.203	0.205	7.700
RMSE	0.042	0.042	0.224	0.223		0.041	0.041	0.207	0.209	

Note: The table shows the results of 100 Monte Carlo runs of various estimators, for different numbers of particles B , using the same random numbers for the underlying simulated time series. The underlying parameters of the MF model are $k = 8$, $m_0 = 1.4$, and $\sigma = 1$. Results are shown as the means, finite sample standard errors (FSSE) and root-mean squared errors (RMSE) of parameter estimates over the 100 Monte Carlo runs. t [sec] is the computation time in seconds per Monte Carlo replication.

Table 3: Estimation of continuous-time bivariate MSM Model

SML, $B = 1,000$		$B = 2,000$					$B = 10,000$					$B = 20,000$						
	m_1	m_2	σ_1	σ_2	ρ	t [sec]	m_1	m_2	σ_1	σ_2	ρ	t [sec]	m_1	m_2	σ_1	σ_2	ρ	t [sec]
Mean	1.398	1.595	1.015	1.038	0.373	345.610	1.398	1.595	1.007	1.041	0.376	1010.570	1.423	1.602	1.830	2.152	0.398	252.490
FSSE	0.028	0.029	0.267	0.449	0.037	16.166	0.024	0.027	0.258	0.453	0.038	43.464	0.103	0.069	0.815	1.115	0.102	3.503
RMSE	0.028	0.029	0.266	0.448	0.046		0.024	0.027	0.257	0.453	0.044		0.105	0.069	1.160	1.599	0.101	
Self-organized State Space, $B = 2000$		$B = 10,000$					$B = 20,000$					$B = 20,000$						
Mean	1.488	1.602	2.115	2.168	0.470	3.880	1.434	1.602	1.864	2.057	0.417	66.880	1.415	1.605	1.000	1.010	0.353	253.780
FSSE	0.117	0.093	1.080	1.369	0.146	0.327	0.100	0.081	0.961	1.273	0.137	1.113	0.017	0.018	0.221	0.382	0.030	3.171
RMSE	0.146	0.092	1.548	1.794	0.162		0.105	0.081	1.289	1.650	0.137		0.023	0.018	0.219	0.380	0.055	
Filter 1, $B = 2000$		$B = 10,000$					$B = 20,000$					$B = 20,000$						
Mean	1.419	1.609	0.984	0.979	0.350	4.140	1.415	1.605	0.992	0.995	0.351	67.460	1.415	1.605	1.000	1.010	0.353	253.780
FSSE	0.025	0.021	0.237	0.387	0.039	0.349	0.018	0.017	0.236	0.392	0.031	0.809	0.017	0.018	0.221	0.382	0.030	3.171
RMSE	0.031	0.023	0.236	0.386	0.063		0.023	0.018	0.234	0.390	0.058		0.023	0.018	0.219	0.380	0.055	
Filter 2, $B = 2000$		$B = 10,000$					$B = 20,000$					$B = 20,000$						
Mean	1.443	1.614	2.059	2.487	0.401	4.030	1.424	1.613	1.730	2.167	0.402	67.860	1.415	1.611	1.646	2.206	0.404	254.100
FSSE	0.068	0.059	1.123	1.189	0.060	0.223	0.056	0.043	0.904	1.082	0.049	1.146	0.044	0.041	0.709	1.150	0.048	3.740
RMSE	0.080	0.061	1.540	1.900	0.060		0.061	0.045	1.159	1.588	0.049		0.046	0.042	0.956	1.663	0.048	

Table 3: continued

Iterated Filter 1											
$B = 1,000$											
	m_1 [200]	m_1 [300]	m_2 [200]	m_2 [300]	σ_1 [200]	σ_1 [300]	σ_2 [200]	σ_2 [300]	ρ [200]	ρ [300]	t [sec]
MSE	1.430	1.430	1.608	1.608	0.983	0.983	0.957	0.957	0.328	0.328	452.500
FSSE	0.005	0.005	0.008	0.008	0.228	0.228	0.379	0.379	0.037	0.037	5.668
RMSE	0.030	0.030	0.011	0.011	0.227	0.227	0.379	0.379	0.081	0.081	
$B = 2,000$											
MSE	1.430	1.430	1.609	1.609	0.980	0.980	0.952	0.952	0.326	0.326	1261.090
FSSE	0.005	0.005	0.006	0.006	0.230	0.230	0.380	0.380	0.037	0.037	14.099
RMSE	0.030	0.030	0.011	0.011	0.229	0.229	0.382	0.382	0.083	0.083	
Iterated Filter 2											
$B = 1,000$											
Mean	1.415	1.415	1.620	1.620	1.495	1.492	2.110	2.109	0.390	0.390	424.030
FSSE	0.051	0.051	0.041	0.042	0.655	0.654	1.233	1.234	0.054	0.054	4.721
RMSE	0.053	0.053	0.046	0.046	0.818	0.816	1.655	1.654	0.055	0.055	
$B = 2,000$											
Mean	1.415	1.415	1.615	1.616	1.477	1.478	2.111	2.110	0.386	0.387	1201.800
FSSE	0.041	0.040	0.042	0.042	0.629	0.630	1.415	1.417	0.046	0.046	13.581
RMSE	0.043	0.043	0.044	0.045	0.787	0.788	1.794	1.794	0.048	0.047	
Iterated Filter 1av											
$B = 1,000$											
MSE	1.402	1.402	1.611	1.611	1.106	1.106	1.307	1.307	0.375	0.374	452.200
FSSE	0.043	0.043	0.043	0.043	0.309	0.310	0.599	0.599	0.053	0.053	5.580
RMSE	0.043	0.043	0.044	0.044	0.325	0.326	0.670	0.671	0.058	0.059	
$B = 2,000$											
MSE	1.401	1.401	1.610	1.610	1.118	1.118	1.313	1.312	0.381	0.381	1261.900
FSSE	0.035	0.035	0.034	0.034	0.334	0.334	0.642	0.641	0.037	0.037	15.405
RMSE	0.035	0.035	0.035	0.035	0.353	0.353	0.711	0.710	0.042	0.042	

Note: The table shows the results of 100 Monte Carlo runs of various estimators, for different numbers of particles B , using the same random numbers for the underlying simulated time series. The underlying parameters of the MF model are $k = 10$, $k_1 = 4$, $m_1 = 1.4$, $m_2 = 1.6$, $\sigma_1 = \sigma_2 = 1$ and $\rho = 0.4$. Results are shown as the means, finite sample standard errors (FSSE) and root-mean squared errors (RMSE) of parameter estimates over the 100 Monte Carlo runs. t [sec] is the computation time in seconds per Monte Carlo replication.

Table 4: Estimation of continuous-time trivariate MSM Model

SML										
$B = 1,000$										
	m_1	m_2	m_3	σ_1	σ_2	σ_3	ρ_{12}	ρ_{23}	ρ_{13}	t [sec]
Mean	1.196	1.393	1.583	0.982	0.914	1.034	0.354	0.621	0.120	1232.710
FSSE	0.034	0.035	0.031	0.255	0.479	0.870	0.035	0.039	0.042	36.622
RMSE	0.034	0.036	0.035	0.255	0.484	0.866	0.058	0.088	0.090	
$B = 2,000$										
Mean	1.199	1.396	1.587	0.996	0.966	1.082	0.351	0.633	0.125	3206.770
FSSE	0.033	0.036	0.029	0.264	0.524	1.008	0.034	0.035	0.039	115.443
RMSE	0.033	0.036	0.031	0.263	0.522	1.006	0.059	0.075	0.084	
Filter 1										
$B = 2,000$										
	m_1	m_2	m_3	σ_1	σ_2	σ_3	ρ_{12}	ρ_{23}	ρ_{13}	t [sec]
Mean	1.306	1.463	1.575	2.054	2.532	2.580	0.365	0.630	0.171	5.350
FSSE	0.112	0.096	0.100	1.015	1.253	1.276	0.082	0.072	0.097	0.479
RMSE	0.154	0.114	0.102	1.460	1.975	2.027	0.089	0.101	0.101	
$B = 10,000$										
Mean	1.263	1.452	1.605	1.601	2.445	2.750	0.368	0.656	0.221	74.490
FSSE	0.080	0.078	0.058	0.752	1.254	1.270	0.054	0.043	0.067	1.168
RMSE	0.101	0.093	0.058	0.960	1.909	2.158	0.063	0.062	0.070	
$B = 20,000$										
Mean	1.257	1.443	1.596	1.722	2.573	3.086	0.372	0.661	0.227	266.740
FSSE	0.071	0.063	0.051	0.726	1.068	1.271	0.053	0.039	0.053	3.575
RMSE	0.091	0.076	0.051	1.021	1.898	2.439	0.060	0.055	0.060	
Filter 2										
$B = 2,000$										
	m_1	m_2	m_3	σ_1	σ_2	σ_3	ρ_{12}	ρ_{23}	ρ_{13}	t [sec]
Mean	1.386	1.554	1.305	1.028	1.008	0.643	0.356	0.543	0.126	5.320
FSSE	0.039	0.031	0.061	0.290	0.509	0.780	0.051	0.052	0.058	0.469
RMSE	0.190	0.157	0.301	0.290	0.506	0.855	0.067	0.165	0.094	
$B = 10,000$										
Mean	1.359	1.534	1.347	1.050	1.011	0.684	0.370	0.567	0.145	74.910
FSSE	0.033	0.031	0.057	0.307	0.475	0.780	0.032	0.038	0.040	1.264
RMSE	0.163	0.137	0.260	0.309	0.473	0.839	0.044	0.138	0.068	
$B = 20,000$										
Mean	1.354	1.529	1.360	1.030	1.005	0.696	0.366	0.573	0.143	268.860
FSSE	0.030	0.030	0.051	0.295	0.496	0.788	0.032	0.032	0.036	3.493
RMSE	0.157	0.132	0.245	0.295	0.494	0.841	0.046	0.131	0.068	

Table 4: continued

Iterated Filter 1										
$B = 1,000$										
	m_1	m_2	m_3	σ_1	σ_2	σ_3	ρ_{12}	ρ_{23}	ρ_{13}	t [sec]
MSE	1.394	1.555	1.337	1.120	1.083	0.909	0.352	0.519	0.133	612.870
FSSE	0.010	0.011	0.039	0.199	0.395	0.699	0.026	0.035	0.027	4.433
RMSE	0.194	0.155	0.266	0.232	0.402	0.702	0.054	0.184	0.072	
$B = 2,000$										
MSE	1.396	1.560	1.328	1.101	1.064	0.937	0.353	0.515	0.135	1590.380
FSSE	0.009	0.009	0.038	0.206	0.399	0.688	0.026	0.035	0.029	12.762
RMSE	0.197	0.160	0.274	0.229	0.402	0.688	0.053	0.188	0.071	
Iterated Filter 1av										
$B = 1,000$										
	m_1	m_2	m_3	σ_1	σ_2	σ_3	ρ_{12}	ρ_{23}	ρ_{13}	t [sec]
MSE	1.194	1.422	1.603	1.066	1.330	1.566	0.398	0.642	0.183	615.980
FSSE	0.061	0.048	0.053	0.327	0.761	0.985	0.060	0.041	0.065	4.399
RMSE	0.061	0.053	0.053	0.332	0.825	1.132	0.060	0.071	0.067	
$B = 2,000$										
MSE	1.190	1.410	1.605	1.057	1.215	1.554	0.393	0.649	0.183	1598.520
FSSE	0.050	0.043	0.035	0.285	0.657	0.925	0.046	0.036	0.047	11.343
RMSE	0.051	0.044	0.035	0.290	0.689	1.074	0.047	0.063	0.049	
Iterated Filter 2										
$B = 1,000$										
	m_1	m_2	m_3	σ_1	σ_2	σ_3	ρ_{12}	ρ_{23}	ρ_{13}	t [sec]
Mean	1.222	1.434	1.620	1.491	2.162	3.316	0.389	0.655	0.194	612.990
FSSE	0.076	0.063	0.061	0.817	1.417	1.955	0.074	0.052	0.084	7.983
RMSE	0.079	0.072	0.064	0.949	1.827	3.025	0.074	0.068	0.084	
$B = 2,000$										
Mean	1.208	1.422	1.613	1.417	2.194	3.450	0.386	0.657	0.189	1589.720
FSSE	0.059	0.053	0.048	0.674	1.552	1.859	0.050	0.042	0.069	20.078
RMSE	0.060	0.057	0.049	0.790	1.952	3.070	0.051	0.060	0.069	

Note: The table shows the results of 100 Monte Carlo runs of various estimators, for different numbers of particles B , using the same random numbers for the underlying simulated time series. The underlying parameters of the MF model are $k = 15$, $k_1 = 8$, $m_1 = 1.2$, $m_2 = 1.4$, $m_3 = 1.6$, $\sigma_1 = \sigma_2 = 1$ and $\rho_{12} = 0.4$, $\rho_{23} = 0.7$, $\rho_{13} = 0.2$. Results are shown as the means, finite sample standard errors (FSSE) and root-mean squared errors (RMSE) of parameter estimates over the 100 Monte Carlo runs. t [sec] is the computation time in seconds per Monte Carlo replication. Results of iterated filters are reported after $M = 300$ iterations.

Table 5: Markov chains with higher number of states, estimation via ML and SML

	Discrete-time MSM								
	$k = 10$								
	ML			SML, $B = 1,000$			$B = 2,000$		
	m_0	s	t [sec]	m_0	s	t [sec]	m_0	s	t [sec]
Mean	1.193	0.975	478.310	1.203	1.036	177.320	1.199	1.022	589.550
FSSE	0.029	0.123	28.999	0.032	0.157	5.664	0.034	0.141	23.305
RMSE	0.030	0.125		0.032	0.160		0.033	0.142	
	$k = 12$								
Mean	1.193	0.946	11063.200	1.202	1.006	185.700	1.203	1.004	614.510
FSSE	0.029	0.154	1143.582	0.030	0.195	6.320	0.030	0.202	26.304
RMSE	0.029	0.159		0.030	0.194		0.030	0.201	
	Continuous-time MSM								
	$k = 10$								
	SML, $B = 1,000$			SML, $B = 2,000$					
	m_0	s	t [sec]	m_0	s	t [sec]			
Mean	1.395	1.137	197.350	1.394	1.159	648.600			
FSSE	0.028	0.283	6.484	0.030	0.343	20.909			
RMSE	0.029	0.313		0.30	0.376				
	$k = 15$								
Mean	1.400	1.107	218.750	1.403	1.162	681.910			
FSSE	0.035	0.651	8.553	0.033	0.590	32.523			
RMSE	0.035	0.656		0.033	0.609				

Table 5: continued

	Bivariate continuous-time MSM, $k = 15, k_1 = 8$											
	SML, $B = 1000$						$B = 2000$					
	m_1	m_2	s_1	s_2	ρ	t [sec]	m_1	m_2	s_1	s_2	ρ	t [sec]
Mean	1.408	1.598	0.862	0.794	0.359	384.740	1.406	1.589	0.875	0.782	0.367	1075.990
FSSE	0.030	0.031	0.527	0.811	0.046	16.685	0.028	0.025	0.526	0.775	0.040	44.667
RMSE	0.031	0.031	0.543	0.833	0.062		0.029	0.027	0.538	0.802	0.051	

Note: The table shows the results of 100 Monte Carlo runs of various models with higher-order Markov chains of the volatility dynamics. For the extremely time-consuming computations of the exact maximum likelihood estimates with $k = 12$ only 20 replications have been performed. Results are shown as the means, finite sample standard errors (FSSE) and root-mean squared errors (RMSE) of parameter estimates over the 100 (20) Monte Carlo runs. tsec is the computation time in seconds per Monte Carlo replication.

Table 6: Estimation results for discrete-time and continuous-time multifractal models

	discrete-time MSM			continuous-time MSM		
	logL	m_0	σ	logL	m_0	σ
DAX	-15782.519	1.28 (1.27 1.29)	3.63 (3.61 3.63)	-15778.148	1.267 (1.26 1.28)	3.45 (3.44 3.46)
S&P	-14466.052	1.30 (1.28 1.31)	2.59 (2.54 2.61)	-14477.741	1.29 (1.25 1.31)	2.66 (2.54 2.68)
Nikkei	-15365.904	1.32 (1.31 1.33)	3.19 (3.18 3.20)	-15372.313	1.30 (1.29 1.31)	2.79 (2.78 2.80)
USD/EUR	-5224.508	1.20 (1.18 1.21)	1.68 (1.67 1.70)	-5228.835	1.20 (1.18 1.20)	1.56 (1.55 1.58)
YEN/USD	-9466.130	1.34 (1.33 1.35)	2.12 (2.11 2.14)	-9488.598	1.30 (1.28 1.30)	1.73 (1.72 1.74)
CHF/EUR	-1550.838	1.27 (1.26 1.28)	0.85 (0.84 0.86)	-1551.004	1.24 (1.23 1.25)	0.72 (0.71 0.74)
Gold	-13942.552	1.41 (1.40 1.42)	3.24 (3.23 3.25)	-13980.734	1.42 (1.41 1.43)	3.50 (3.49 3.51)

Note: The table presents the maximized likelihood values and the estimates of the parameters m_0 and σ for both the discrete-time and continuous-time univariate multifractal model for a selection of seven representative financial time series. In-sample estimation horizons are as described in the main test. In order to compute confidence intervals for the point estimator, we have used the profile likelihood method as the lack of continuity of the particle filter approximation prevents us from using derivative-based methods. The profile likelihood is a function of one parameter, say θ_1 , with the remaining parameters, say θ_2 , being obtained by maximization under the constraint that θ_1 is fixed. A 95 percent confidence interval is, then, defined by upper and lower values $\bar{\theta}_{1,\pm}$ so that $|l(\hat{\theta}_1) - l(\bar{\theta}_{1,\pm})| = 1.92$, with $l(\cdot)$ the log likelihood function, and $\hat{\theta}_1$ the globally optimal value of the pertinent parameter. Because of the computational demands of the maximization step, we have computed the profile likelihood for integer steps of length 0.01 around the global optimum. As it can be seen in the Table, the so obtained confidence intervals are very narrow throughout, and often already the immediate neighboring values of $\hat{\theta}_1$ would give a likelihood difference outside the 95 percent interval.

Table 7: Model selection tests

	MF vs GARCH	adj. (0.000)	PMF vs GARCH	adj. (0.000)	MF vs FIGARCH	adj. (0.000)	PMF vs FIGARCH	adj. (0.000)	MF vs PMF
DAX	2.031 (0.021)	6.423 (0.000)	2.059 (0.020)	6.451 (0.000)	1.887 (0.030)	10.670 (0.000)	1.929 (0.027)	10.712 (0.000)	-0.737 (0.769)
S&P500	3.147 (0.001)	7.538 (0.000)	3.030 (0.001)	7.421 (0.000)	3.463 (0.000)	12.246 (0.000)	3.345 (0.000)	12.128 (0.000)	2.817 (0.002)
Nikkei	3.453 (0.000)	7.845 (0.000)	3.419 (0.000)	7.811 (0.000)	3.610 (0.000)	12.392 (0.000)	3.580 (0.000)	12.363 (0.000)	1.274 (0.101)
USD/EUR	1.081 (0.140)	5.061 (0.000)	0.673 (0.250)	4.654 (0.000)	0.095 (0.462)	8.056 (0.000)	-0.337 (0.632)	7.624 (0.000)	1.719 (0.043)
YEN/USD	6.206 (0.000)	10.460 (0.000)	6.429 (0.000)	10.683 (0.000)	6.417 (0.000)	14.925 (0.000)	6.685 (0.000)	15.193 (0.000)	3.274 (0.000)
CHF/EUR	1.028 (0.152)	4.743 (0.000)	1.136 (0.128)	4.851 (0.000)	0.974 (0.165)	8.404 (0.000)	1.087 (0.138)	8.517 (0.000)	0.017 (0.493)
Gold	7.761 (0.000)	12.152 (0.000)	7.284 (0.000)	11.676 (0.000)	7.638 (0.000)	16.421 (0.000)	7.164 (0.000)	15.947 (0.000)	6.875 (0.000)

Note: The table shows the results of the in-sample model selection test for non-nested alternatives of Vuong (1989). The probabilities in parentheses are the probabilities of the rejection of the null hypothesis that the first named model provides a better fit to the data than the second one. The adjusted version of the Vuong test (labelled "adj.") takes into account different numbers of parameters of the competing models. Both versions of the test are symmetric so that high values of the probabilities could be seen as evidence in favor of dominance of the second model.

Table 8: Out-of-sample forecasting comparisons

horizon	DAX: MSE					DAX: MAE				
	GARCH	FIGARCH	MF	PMF	HV	GARCH	FIGARCH	MF	PMF	HV
1	0.844 (0.522)	0.841 (0.920)	0.857 (0.278)	0.871 (0.022)	1.000 (0.000)	0.929 (0.083)	0.930 (0.037)	0.918 (0.000)	0.907 (0.955)	1.000 (0.000)
5	0.862 (0.818)	0.867 (0.583)	0.887 (0.207)	0.900 (0.020)	1.000 (0.000)	0.938 (0.013)	0.946 (0.000)	0.926 (0.000)	0.911 (1.000)	1.000 (0.000)
10	0.883 (0.765)	0.887 (0.670)	0.911 (0.188)	0.923 (0.022)	1.000 (0.000)	0.961 (0.005)	0.974 (0.000)	0.943 (0.000)	0.926 (1.000)	1.000 (0.000)
20	0.945 (0.640)	0.948 (0.690)	0.944 (0.542)	0.953 (0.007)	1.000 (0.000)	0.991 (0.000)	1.005 (0.000)	0.963 (0.000)	0.944 (1.000)	1.000 (0.000)
30	0.963 (0.660)	0.970 (0.540)	0.965 (0.845)	0.971 (0.058)	1.000 (0.010)	1.013 (0.000)	1.033 (0.000)	0.980 (0.000)	0.958 (1.000)	1.000 (0.000)
40	0.992 (0.263)	0.995 (0.115)	0.979 (0.995)	0.983 (0.068)	1.000 (0.007)	1.032 (0.000)	1.052 (0.000)	0.993 (0.000)	0.969 (1.000)	1.000 (0.000)
50	1.007 (0.020)	1.010 (0.000)	0.985 (0.950)	0.988 (0.050)	1.000 (0.015)	1.045 (0.000)	1.066 (0.000)	1.000 (0.000)	0.974 (1.000)	1.000 (0.000)

Table 8: continued

S&P: MSE		S&P: MAE								
horizon	GARCH	FIGARCH	MF	PMF	HV	GARCH	FIGARCH	MF	PMF	HV
1	0.759 (0.815)	0.769 (0.233)	0.850 (0.033)	0.848 (0.020)	1.000 (0.000)	0.944 (0.022)	0.947 (0.000)	0.908 (0.000)	0.904 (1.000)	1.000 (0.000)
5	0.767 (0.672)	0.772 (0.328)	0.877 (0.010)	0.880 (0.018)	1.000 (0.000)	0.946 (0.025)	0.948 (0.000)	0.909 (0.000)	0.903 (1.000)	1.000 (0.000)
10	0.816 (0.785)	0.827 (0.255)	0.907 (0.025)	0.912 (0.043)	1.000 (0.000)	0.973 (0.018)	0.982 (0.000)	0.927 (0.000)	0.922 (1.000)	1.000 (0.000)
20	0.882 (0.718)	0.891 (0.680)	0.939 (0.060)	0.943 (0.013)	1.000 (0.000)	1.003 (0.013)	1.011 (0.000)	0.945 (0.003)	0.940 (1.000)	1.000 (0.000)
30	0.939 (0.505)	0.935 (0.965)	0.955 (0.175)	0.961 (0.000)	1.000 (0.000)	1.032 (0.000)	1.033 (0.000)	0.954 (0.010)	0.951 (1.000)	1.000 (0.000)
40	0.984 (0.142)	0.965 (0.870)	0.964 (0.568)	0.971 (0.003)	1.000 (0.000)	1.055 (0.000)	1.052 (0.000)	0.963 (0.000)	0.959 (1.000)	1.000 (0.000)
50	1.002 (0.075)	0.980 (0.443)	0.972 (0.895)	0.980 (0.000)	1.000 (0.000)	1.070 (0.000)	1.066 (0.000)	0.970 (0.000)	0.965 (1.000)	1.000 (0.000)

Table 8: continued

Nikkei: horizon	Nikkei: MSE					Nikkei: MAE				
	GARCH	FIGARCH	MF	PMF	HV	GARCH	FIGARCH	MF	PMF	HV
1	0.787 (0.595)	0.786 (0.927)	0.863 (0.090)	0.882 (0.037)	1.000 (0.007)	1.033 (0.068)	1.021 (0.165)	1.015 (0.000)	0.991 (0.985)	1.000 (0.335)
5	0.874 (0.230)	0.852 (1.000)	0.911 (0.115)	0.926 (0.025)	1.000 (0.000)	1.061 (0.003)	1.028 (0.025)	1.018 (0.000)	0.987 (0.907)	1.000 (0.168)
10	0.938 (0.092)	0.892 (1.000)	0.936 (0.170)	0.948 (0.025)	1.000 (0.000)	1.101 (0.000)	1.048 (0.007)	1.033 (0.000)	0.994 (0.777)	1.000 (0.313)
20	1.101 (0.013)	0.987 (0.063)	0.962 (0.953)	0.969 (0.020)	1.000 (0.000)	1.191 (0.000)	1.097 (0.000)	1.042 (0.000)	0.996 (0.693)	1.000 (0.307)
30	1.152 (0.003)	1.007 (0.007)	0.972 (1.000)	0.980 (0.013)	1.000 (0.000)	1.252 (0.000)	1.116 (0.000)	1.050 (0.000)	1.000 (0.550)	1.000 (0.450)
40	1.195 (0.005)	1.019 (0.005)	0.978 (1.000)	0.984 (0.007)	1.000 (0.000)	1.294 (0.000)	1.126 (0.000)	1.051 (0.000)	1.000 (0.502)	1.000 (0.497)
50	1.213 (0.003)	1.022 (0.000)	0.982 (1.000)	0.987 (0.013)	1.000 (0.000)	1.351 (0.000)	1.135 (0.000)	1.054 (0.000)	1.000 (0.530)	1.000 (0.470)

Table 8: continued

horizon	USD/EUR: MSE					USD/EUR: MAE				
	GARCH	FIGARCH	MF	PMF	HV	GARCH	FIGARCH	MF	PMF	HV
1	0.931 (0.995)	0.934 (0.365)	0.941 (0.050)	0.940 (0.070)	1.000 (0.000)	0.886 (1.000)	0.895 (0.000)	0.923 (0.000)	0.912 (0.000)	1.000 (0.000)
5	0.933 (0.838)	0.936 (0.212)	0.935 (0.365)	0.933 (0.748)	1.000 (0.000)	0.889 (1.000)	0.896 (0.000)	0.931 (0.000)	0.915 (0.000)	1.000 (0.000)
10	0.937 (0.890)	0.942 (0.140)	0.940 (0.362)	0.938 (0.677)	1.000 (0.000)	0.890 (1.000)	0.899 (0.000)	0.938 (0.000)	0.921 (0.000)	1.000 (0.000)
20	0.941 (0.990)	0.946 (0.133)	0.946 (0.258)	0.944 (0.445)	1.000 (0.000)	0.895 (1.000)	0.906 (0.000)	0.954 (0.000)	0.931 (0.000)	1.000 (0.000)
30	0.946 (0.980)	0.954 (0.070)	0.951 (0.323)	0.948 (0.515)	1.000 (0.000)	0.903 (1.000)	0.919 (0.000)	0.970 (0.000)	0.941 (0.000)	1.000 (0.000)
40	0.955 (0.672)	0.966 (0.015)	0.960 (0.230)	0.956 (0.698)	1.000 (0.000)	0.908 (1.000)	0.929 (0.000)	0.980 (0.000)	0.948 (0.000)	1.000 (0.000)
50	0.955 (0.775)	0.968 (0.015)	0.964 (0.107)	0.957 (0.563)	1.000 (0.000)	0.915 (1.000)	0.941 (0.000)	0.990 (0.000)	0.954 (0.000)	1.000 (0.000)

Table 8: continued

YEN/USD: MSE		YEN/USD: MAE								
horizon	GARCH	FIGARCH	MF	PMF	HV	GARCH	FIGARCH	MF	PMF	HV
1	0.920 (0.777)	0.928 (0.210)	0.922 (0.465)	0.920 (0.950)	1.000 (0.025)	0.899 (0.005)	0.892 (0.030)	0.918 (0.000)	0.885 (0.970)	1.000 (0.000)
5	0.932 (0.775)	0.941 (0.175)	0.935 (0.435)	0.932 (0.897)	1.000 (0.000)	0.910 (0.000)	0.904 (0.020)	0.936 (0.000)	0.895 (1.000)	1.000 (0.000)
10	0.936 (0.912)	0.947 (0.158)	0.942 (0.253)	0.940 (0.593)	1.000 (0.003)	0.923 (0.000)	0.915 (0.025)	0.956 (0.000)	0.907 (0.975)	1.000 (0.000)
20	0.963 (0.448)	0.969 (0.105)	0.966 (0.070)	0.960 (1.000)	1.000 (0.003)	0.940 (0.000)	0.926 (0.200)	0.982 (0.000)	0.923 (0.800)	1.000 (0.000)
30	0.977 (0.280)	0.979 (0.133)	0.982 (0.013)	0.973 (0.960)	1.000 (0.000)	0.953 (0.000)	0.934 (0.367)	1.004 (0.000)	0.933 (0.632)	1.000 (0.000)
40	0.985 (0.058)	0.985 (0.085)	0.992 (0.000)	0.978 (0.985)	1.000 (0.000)	0.965 (0.000)	0.941 (0.583)	1.020 (0.000)	0.942 (0.417)	1.000 (0.000)
50	0.989 (0.065)	0.988 (0.122)	0.999 (0.000)	0.983 (0.963)	1.000 (0.000)	0.972 (0.000)	0.945 (0.930)	1.034 (0.000)	0.951 (0.070)	1.000 (0.000)

Table 8: continued

horizon	CHF/EUR: MSE				CHF/EUR: MAE					
	GARCH	FIGARCH	MF	PMF	HV	GARCH	FIGARCH	MF	PMF	HV
1	0.997 (0.943)	1.001 (0.385)	0.996 (0.745)	0.997 (0.058)	1.000 (0.072)	1.233 (0.005)	1.266 (0.005)	0.935 (0.000)	0.924 (1.000)	1.000 (0.000)
5	1.044 (0.022)	1.059 (0.030)	0.997 (0.945)	0.999 (0.037)	1.000 (0.085)	1.319 (0.000)	1.381 (0.003)	0.941 (0.003)	0.932 (1.000)	1.000 (0.000)
10	1.042 (0.025)	1.054 (0.043)	0.998 (0.998)	0.999 (0.030)	1.000 (0.083)	1.289 (0.003)	1.363 (0.000)	0.946 (0.000)	0.936 (1.000)	1.000 (0.000)
20	1.031 (0.050)	1.041 (0.052)	0.998 (0.990)	0.999 (0.030)	1.000 (0.145)	1.227 (0.003)	1.335 (0.000)	0.957 (0.000)	0.945 (1.000)	1.000 (0.000)
30	1.002 (0.160)	1.004 (0.055)	0.999 (0.995)	1.000 (0.035)	1.000 (0.155)	1.067 (0.005)	1.181 (0.000)	0.964 (0.000)	0.951 (1.000)	1.000 (0.000)
40	1.001 (0.532)	1.003 (0.083)	0.999 (0.995)	1.000 (0.033)	1.000 (0.200)	1.059 (0.000)	1.196 (0.000)	0.970 (0.000)	0.956 (1.000)	1.000 (0.000)
50	1.002 (0.065)	1.005 (0.033)	0.999 (0.887)	1.000 (0.028)	1.000 (0.235)	1.058 (0.000)	1.214 (0.000)	0.977 (0.000)	0.961 (1.000)	1.000 (0.000)

Table 8: continued

horizon	Gold: MSE				Gold: MAE				
	GARCH	FIGARCH	MF	PMF	HV	HV	MF	PMF	HV
1	0.939 (0.820)	0.947 (0.180)	0.969 (0.007)	0.976 (0.000)	1.000 (0.005)	1.000 (0.000)	1.081 (0.000)	1.089 (0.000)	1.000 (0.068)
5	0.940 (0.718)	0.940 (0.620)	0.941 (0.700)	0.940 (0.780)	1.000 (0.000)	0.995 (0.000)	1.071 (0.000)	1.075 (0.000)	1.000 (0.000)
10	0.947 (0.640)	0.947 (0.698)	0.953 (0.347)	0.953 (0.443)	1.000 (0.000)	1.002 (0.000)	1.089 (0.000)	1.094 (0.000)	1.000 (0.000)
20	0.958 (0.583)	0.957 (0.725)	0.963 (0.170)	0.959 (0.760)	1.000 (0.003)	1.022 (0.000)	1.116 (0.000)	1.118 (0.000)	1.000 (0.000)
30	0.974 (0.357)	0.970 (0.510)	0.974 (0.033)	0.967 (0.843)	1.000 (0.003)	1.037 (0.000)	1.128 (0.000)	1.128 (0.000)	1.000 (0.000)
40	0.982 (0.340)	0.978 (0.515)	0.978 (0.225)	0.975 (0.932)	1.000 (0.018)	1.057 (0.000)	1.142 (0.000)	1.143 (0.000)	1.000 (0.000)
50	0.996 (0.100)	0.987 (0.527)	0.988 (0.007)	0.981 (0.968)	1.000 (0.075)	1.068 (0.000)	1.150 (0.000)	1.148 (0.000)	1.000 (0.000)

Note: The table reports mean square errors (MSE) and mean absolute errors (MAE) of the estimated models over the out-of-sample periods specified in the main test. MSEs and MAEs have been normalized by dividing by the pertinent numbers obtained for time-invariant forecasts based on historical volatility during the in-sample period (denoted as HV). In the brackets, we show the results of Hansen's test for superior predictive ability. The null hypothesis tested is that the pertinent benchmark model cannot be outperformed by any of the four other competing models (including HV).

References

- Bhadra, A., E. L. Ionides, K. Laneri, M. Pascual, M. Bouma and R. C. Dhiman (2011), Malaria in northwest india: Data analysis via partially observed stochastic differential equation models driven by Lévy noise., *Journal of the American Statistical Association* 106 494, 440–451.
- Bretó, C. (2014), On idiosyncratic stochasticity of financial leverage effects, *Statistics and Probability Letters* 91, 20–26.
- Calvet, L. and A. Fisher (2001), Forecasting multifractal volatility, *Journal of Econometrics* 105 1, 27–58.
- Calvet, L. and A. Fisher (2004), How to forecast long-run volatility: Regime switching and the estimation of multifractal processes, *Journal of Financial Econometrics* 2 1, 49–83.
- Calvet, L., A. Fisher and S. Thompson (2006), Volatility comovement: A multi-frequency approach, *Journal of Econometrics* 31, 179–215.
- Calvet, L., A. Fisher and L. Wu (2017), Staying on top of the curve: A cascade model of the term structure dynamics, *Journal of Financial and Quantitative Analysis* , in press.
- Chen, Z., editor (2015), *Advanced State-Space Methods for Neural and Clinical Data*, Cambridge University Press.
- Chen, Z. and T. Lux (2017), Estimation of sentiment effects in financial markets: A simulated method of moments approach, *Computational Economics* , in press.
- Chong, E. and S. Zak (2011), *An Introduction to Optimization*, John Wiley, New York, 3rd edition.
- Fernández-Villaverde, J. and J. F. Rubio-Ramirez (2007), Estimating macroeconomic models: A likelihood approach, *Review of Economic Studies* 74 4, 1059–1087.
- Forbes, K. and R. Rigobon (2002), No contagion, only interdependence: Measuring stock market comovements, *Journal of Finance* 57 5, 2223–2261.
- Ghoshadze, J. and T. Lux (2016), Bringing an elementary agent-based model to the data: Estimation via GMM and an application to forecasting of asset price volatility, *Journal of Empirical Finance* 37, 1–19.
- Gillespie, D. T. (1977), Exact stochastic simulation of coupled chemical reactions, *Journal of Physical Chemistry* 81 25, 2340–2361.
- Gordon, N., D. Salmond and A. Smith (1993), Novel approach to nonlinear / non-gaussian Bayesian state estimation, *IEE Proceedings F - Radar and Signal Processing* 140 2, 107–113.
- Grewal and Andrews (2008), *Kalman Filtering Theory and Practice Using MATLAB*, Wiley & Sons.

- Hansen, P. R. (2005), A test for superior predictive ability, *Journal of Economics & Business Statistics* 23 4, 365–380.
- He, D., E. L. Ionides and A. A. King (2010), Plug-and-play inference for disease dynamics: measles in large and small populations as a case study, *Journal of the Royal Society Interface* 7 43, 271–283.
- Ionides, E. L., A. Bhadra, Y. Atchadé and A. King (2011), Iterated filtering, *The Annals of Statistics* 39 3, 1776–1802.
- Ionides, E. L., C. Bretó and A. A. King (2006), Inference for nonlinear dynamical systems, *Proceedings of the National Academy of Sciences* 103 49, 18438–18443.
- Ionides, E. L., D. Nguyen, Y. Atchadé, S. Stoev and A. A. King (2015), Inference for dynamic and latent variable models via iterated, perturbed Bayes maps, *Proceedings of the National Academy of Sciences* 112 3, 719–724.
- Kalman, R. (1960), A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* 82 1, 35–45.
- Kantas, N., A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin (2015), On particle methods for parameter estimation in state-space models, *Statistical Science* 30 3, 328–351.
- King, A. A., E. L. Ionides, M. Pascual, and M. J. Bouma (2008), Inapparent infections and cholera dynamics, *Nature* 454 7206, 877–880.
- King, A. A., D. Nguyen and E. L. Ionides (2016), Statistical inference for partially observed Markov processes via the R package pomp, *Journal of Statistical Software* 69 12.
- Kitagawa, G. (1996), Monte Carlo filter and smoother for non-gaussian nonlinear state space models, *Journal of Computational and Graphical Statistics* 5, 1–25.
- Kitagawa, G. (1998), A self-organizing state-space model, *Journal of the American Statistical Association* 93 443, 1203–1215.
- Leövey, A. (2015), *Multifractal Models: Estimation, Forecasting and Option Pricing*, Ph.D. thesis, Christian-Albrechts-Universität zu Kiel.
- Liu, J. and M. West (2001), Combined parameter and state estimation in simulation-based filtering, in A. Doucet, J. de Freitas and N. Gordon, editors, *Sequential Monte Carlo in Practice*, Springer, New York.
- Liu, R. and T. Lux (2015), Non-homogeneous volatility correlations in the bivariate multifractal model, *European Journal of Finance* 21, 971–991.
- Liu, R. and T. Lux (2017), Generalized method of moment estimation of multivariate multifractal models, *Economic Modelling* 67, 136–148.
- Lux, T. (2008), The Markov-switching multifractal model of asset returns: Gmm estimation and linear forecasting of volatility, *Journal of Business & Economics Statistics* 26, 194–210.

- Lux, T. (2017), Estimation of agent-based models using sequential Monte Carlo methods, *Journal of Economic Dynamics and Control*, in press.
- Lux, T., L. Morales-Arias and C. Sattarhoff (2014), Forecasting daily variations of stock index returns with a multifractal model of realized volatility, *Journal of Forecasting* 33, 532–541.
- Lux, T. and M. Segnon (2018), Multifractal models in finance: Their origin, properties, and applications, in S.-H. Chen, M. Kaboudan and Y.-R. Du, editors, *The Oxford Handbook on Computational Economics and Finance*, Oxford University Press, pages 204–248.
- Malik, S. and K. P. Pitt (2011), Particle filters for continuous likelihood evaluation and maximisation, *Journal of Econometrics* 165 2, 190–209.
- Mandelbrot, B. (1974), Intermittent turbulence in self-similar cascades: divergence of high moments and dimension of the carrier, *Journal of Fluid Mechanics* 62 2, 331–358.
- Mandelbrot, B., L. Calvet and A. Fisher (1997), A multifractal model of asset returns, Cowles Commission Working Paper, Yale University.
- Rios, M. and H. Lopes (2013), The extended Liu and West filter: Parameter learning in Markov-switching volatility models, in Y. Zeng and S. Wu, editors, *State-space Models: Applications in Economics and Finance*, Springer, New York.
- Vuong, Q. H. (1989), Likelihood ratio tests for model selection and non-nested hypotheses, *Econometrica* 57 2, 307–333.
- Zeng, Y. and S. Wu (2013), *State-space Models: Applications in Economics and Finance*, Springer, Berlin.