Computational Statistics (2020) 35:1319–1344
https://doi.org/10.1007/s00180-019-00937-4

ORIGINAL PAPER

Efficient inference in state-space models through adaptive learning in online Monte Carlo expectation maximization

Donna Henderson[1] · Gerton Lunter[2]

Received: 3 June 2018 / Accepted: 22 November 2019 / Published online: 3 December 2019

Abstract
Expectation maximization (EM) is a technique for estimating maximum-likelihood parameters of a latent variable model given observed data by alternating between taking expectations of sufficient statistics, and maximizing the expected log likelihood. For situations where sufficient statistics are intractable, stochastic approximation EM (SAEM) is often used, which uses Monte Carlo techniques to approximate the expected log likelihood. Two common implementations of SAEM, Batch EM (BEM) and online EM (OEM), are parameterized by a "learning rate", and their efficiency depend strongly on this parameter. We propose an extension to the OEM algorithm, termed Introspective Online Expectation Maximization (IOEM), which removes the need for specifying this parameter by adapting the learning rate to trends in the parameter updates. We show that our algorithm matches the efficiency of the optimal BEM and OEM algorithms in multiple models, and that the efficiency of IOEM can exceed that of BEM/OEM methods with optimal learning rates when the model has many parameters. Finally we use IOEM to fit two models to a financial time series. A Python implementation is available at https://github.com/luntergroup/IOEM.git.

Keywords Stochastic approximation expectation maximization · Sequential Monte Carlo · Latent variable model · Online estimation

1 Introduction

Expectation Maximization (EM) is a general and widely used technique for estimating maximum likelihood parameters of latent variable models (Dempster et al. 1977). It involves iterating two steps: computing the expected log-likelihood marginalizing over the latent variable conditioned on parameters and data (the E step), and optimizing

✉ Gerton Lunter
gerton.lunter@well.ox.ac.uk

[1] Wellcome Centre of Human Genetics, University of Oxford, Oxford OX3 7BN, UK

[2] MRC Weatherall Institute of Molecular Medicine, Unversity of Oxford, Oxford OX3 9DS, UK

parameters to maximize this expected log-likelihood (the M step). In important special cases the E-step is analytically tractable; examples include linear systems with Gaussian noise (Shumway and Stoffer 1982) and finite-state hidden Markov models (Baum 1972). In general however, Monte Carlo techniques such as Stochastic EM (SEM; Celeux and Diebolt 1985; Celeux et al. 1995) and Monte Carlo EM (MCEM; Wei and Tanner 1990) are necessary to approximate the required integral. The stochastic nature of Monte Carlo techniques result in noisy parameter estimates, and to address this, methods such as Stochastic Approximation EM (SAEM; Nowlan 1991; Celeux and Diebolt 1992; Delyon et al. 1999) were developed that make smaller incremental updates parameterized by a learning rate γ or learning schedule $\{\gamma_t\}$.

In this paper we focus on models where the latent variable has a longitudinal structure and follows a Markov model (see e.g. Lopes and Tsay 2010 for examples in financial econometrics). For such models, the required samples from the posterior distribution can be generated using Sequential Monte Carlo (SMC) techniques (see Doucet et al. 2001; Doucet and Johansen 2009 and references therein). In one approach, the Batch EM (BEM) algorithm processes a contiguous chunk of data to generate latent variable samples from the posterior, which are used in the M step to update parameters. An alternative approach is online EM (OEM; Mongillo and Denève 2008; Cappé 2009), in which parameters are continuously updated as data are processed. Analogous to SAEM, OEM algorithms have a parameter γ controlling the learning rate, an idea apparently first introduced in this context by Jordan and Jacobs (1993). Several recent papers have addressed related problems. For instance Yildirim et al. (2013) use a particle filter to implement an online EM algorithm for change point models (see also Fearnhead 2006; Fearnhead and Vasileiou 2009), which uses a pre-specified learning schedule (called "step-size sequence" in their work) to control convergence. Le Corff and Fort (2013) introduced a "block online" EM algorithm for hidden Markov models that combines online and batch ideas, controlling convergence through a block size sequence τ_k.

All these algorithms thus require choosing tuning parameters in the form of a batch size, block sequence, learning rate or a learning schedule. It turns out that this choice can strongly influences the performance of these algorithms. For instance, for BEM, very large batch sizes lead to inaccurate estimates because of slow convergence, whereas very small batch sizes lead to imprecise estimates due to the inherent stochasticity of the model within a small batch of observations. The optimal batch size in BEM or the optimal learning rate in OEM depends on the particularities of the model.

This raises the question of how to choose this tuning parameter. Several authors have proposed adaptive acceleration techniques for EM methods that obviate the need for choosing tuning parameters (Jamshidian and Jennrich 1993; Lange 1995; Varadhan and Roland 2008), but these methods require that the E-step is analytically tractable. In the context of (stochastic) gradient descent optimization (Bottou 2012), several influential adaptive algorithms have recently been proposed (Zeiler 2012; Kingma and Ba 2015; Mandt et al. 2016; Reddi et al. 2018) that have few or no tuning parameters. In principle, these methods can be used to find maximum likelihood parameters, but unless data is processed in batches, applying these methods to state-space models with a sequential structure is not straightforward. In addition, EM approaches enjoy several advantages over gradient descent methods, including automatic guarantees of parameter constraints and increased numerical stability (Xu and Jordan 1996; Cappé 2009; Kantas et al. 2009; Chitralekha et al. 2010).

Here we introduce a novel algorithm, termed Introspective Online EM (IOEM), which removes the need for setting the learning rate by estimating optimal parameter-specific learning rates from the data. This is particularly helpful when inferring parameters in a high dimensional model, since the optimal learning rate may differ between parameters. IOEM can be applied to inference in state-space models with observations Y_t and state variables X_t governed by transition probability function $f(x_{t+1}|x_t,\theta)$ and observation probability function $g(y_t|x_t,\theta)$, for which $f(x_t|x_{t-1},\theta)g(y_t|x_t,\theta)$ belongs to an exponential family with sufficient statistic $s(x_{t-1},x_t,y_t)$. Broadly, IOEM works by estimating both the precision and the accuracy of parameters in an online manner through weighted linear regression, and uses these estimates to tune the learning rate so as to improve both simultaneously.

The outline of this paper is as follows. Section 2 introduces BEM, OEM, and a simplified version of IOEM in the context of a one-parameter autoregressive state-space model. Section 3 introduces the complete IOEM algorithm required for inference in the full 3-parameter autoregressive model. Section 4 discusses simulation results of the algorithms for these two models. In addition we consider a 2-dimensional autoregressive model to show the benefit of the proposed algorithm when inferring many parameters, and we demonstrate desirable performance in the stochastic volatility model, an important case as it is nonlinear and hence relevant to actual applications of SAEM. In Sect. 5 we apply IOEM with the autoregressive and stochastic volatility models to a financial time series, and we end the paper with a brief discussion.

2 EM algorithms for a simplified autoregressive model

Here we review BEM (Dempster et al. 1977), OEM (Cappé 2009) and SMC (Doucet and Johansen 2009), and present the IOEM algorithm in a simplified context. This illustrates the main ideas behind IOEM before presenting the full algorithm in Sect. 3.

We consider a simple noisily-observed autoregressive model with one unknown parameter, equivalent to an ARMA(1,1) model. We observe the sequence of random variables $Y_{1:t} := \{Y_k\}_{k=1,\dots,t}$ that depend on the unobserved sequence $X_{1:t} := \{X_k\}_{k=1,\dots,t}$ as follows:

$$\begin{aligned} X_t &= aX_{t-1} + \sigma_w W_t, \\ Y_t &= X_t + \sigma_v V_t, \end{aligned} \tag{1}$$

where W_t and V_t are i.i.d. standard normal variates, $a = 0.95$ and $\sigma_w^2 = 1$ are known parameters, and σ_v^2 is unknown. Under this model, we have the following transition and emission densities:

$$f(x_t|x_{t-1}) = (2\pi\sigma_w^2)^{-1/2} \exp\left\{ -\frac{(x_t - ax_{t-1})^2}{2\sigma_w^2} \right\},$$

$$g(y_t|x_t) = (2\pi\sigma_v^2)^{-1/2} \exp\left\{ -\frac{(y_t - x_t)^2}{2\sigma_v^2} \right\}.$$

We have chosen σ_v^2 as the unknown parameter as it is the most straightforward to estimate, allowing us to introduce the idea of IOEM while avoiding certain complications that we address in Sect. 3. As f and g are members of the exponential family of distributions, the M step of EM can be done using sufficient statistics, and the E step amounts to calculating their expectation. In this model, the parameter σ_v^2 has the sufficient statistic

$$S_t = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t}\sum_{k=1}^{t}(Y_k - X_k)^2\right]. \tag{2}$$

The estimate of σ_v^2 is obtained by setting $\hat{\sigma}_{v,t}^2 = \hat{S}_t$. More generally, for an unknown parameter θ, $\hat{\theta}_t = \Lambda(\hat{S}_t)$ where Λ is a known function mapping sufficient statistics to parameter estimates.

To estimate S_t, we use sequential Monte Carlo (SMC) to simulate particles $X_{1:t}^{(i)}$ and their associated weights $w(X_{1:t}^{(i)})$, $i = 1, \ldots, N$, so that

$$\sum_{i=1}^{N} w(X_{1:t}^{(i)})\delta_{X_{1:t}^{(i)}} \tag{3}$$

approximates the distribution $p(X_{1:t}|Y_{1:t}, \theta)$. The standard MCEM approximation of $p(X_{1:t}|Y_{1:t}, \hat{\theta})$ would require storage of all observations $Y_{1:t}$ and simulation of $X_{1:t}^{(i)}$ each time $\hat{\theta}$ is updated, and ideally an increasing Monte Carlo sample size as the parameter estimates near convergence. To avoid this, we employ SAEM (Celeux and Diebolt 1992) which effectively averages over previous parameter estimates as an alternative to generating a new Monte Carlo sample every time an estimate is updated, and hence is more suitable to online inference. This method as proposed in Cappé and Moulines (2009) approximates the expectation in (2) recursively.

The outline of the SMC with EM algorithm we consider in this paper is as follows (Doucet and Johansen 2009):

Algorithm 1 Sequential Importance Resampling (bootstrap filter)

```
For time $t \geq 1$:
  1. For $i = 1, \ldots, N$ :
       Sample $X_t^{(i)} \sim \begin{cases} \mu(\cdot|\hat{\theta}_0), & \text{if } t = 1 \\ f(\cdot|X_{t-1}^{(i)}, \hat{\theta}_{t-1}), & \text{if } t \geq 2 \end{cases}$
  2. Compute normalized weights satisfying $w_t(X_{1:t}^{(i)}) \propto w_{t-1}(X_{1:t-1}^{(i)})\cdot g(Y_t|X_t^{(i)}, \hat{\theta}_{t-1})$
  3. Update $\hat{\theta}_{t-1}$ to $\hat{\theta}_t$ using chosen EM method
  4. Resample particles if $ESS < \frac{N}{2}$
```

Here $\mu(\cdot|\hat{\theta}_0)$ is the initial distribution for X_1, ESS is the effective sample size defined as $[\sum_{i=1}^{N} w_t(X_{1:t}^{(i)})^{-2}]^{-1}$, $w_0(\cdot) = 1/N$, and $X_t^{(i)}$ is shorthand for the t^{th} coordinate of $X_{1:t}^{(i)}$. In models with multiple unknown parameters, each parameter is updated in step 3 of the algorithm, however we will refer only to a single parameter θ to keep the notation simple.

Throughout this paper we follow common practice in using the fixed-lag technique in order to reduce the mean square error between S_t and $\hat{S}_t$ (Cappé and Moulines 2005; Cappé et al. 2007). We choose a lag $\Delta > 0$ and at time t, using particles $X_{1:t}^{(i)}$ shaped by data $Y_{1:t}$, we estimate the $t - \Delta^{\text{th}}$ term of the summation in (2). We will use $X_{1:t}^{(i)}(t - \Delta)$ to denote the $t - \Delta^{\text{th}}$ coordinate of the particle $X_{1:t}^{(i)}$, but we will continue to write $X_t^{(i)}$ as a shorthand for $X_{1:t}^{(i)}(t)$. (See Table 1 for an overview of notation used in this paper).

The fixed-lag technique involves making the approximation

$$S_t \approx \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-\Delta}\sum_{j=1}^{t-\Delta} s(Y_j, X_j)\right] \approx \frac{1}{t-\Delta}\sum_{j=1}^{t-\Delta} \mathbb{E}_{X_{1:j+\Delta}|Y_{1:j+\Delta},\hat{\theta}}\left[s(Y_j, X_j)\right]$$

where we assume that S_t can be written as

$$S_t = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta} \sum_{j=1}^{t} s(Y_j, X_j)$$

This allows S_t to be updated in an online manner by computing the component-wise sufficient statistics

$$\begin{aligned}\tilde{s}_t &:= \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[s(Y_{t-\Delta}, X_{1:t}(t-\Delta))\right] \\ &\approx \sum_i w_k(X_{1:t}^{(i)}) s(Y_{t-\Delta}, X_{1:t}^{(i)}(t-\Delta)),\end{aligned}$$

allowing $\hat{S}_t$ to be updated as

$$\hat{S}_t = \gamma_t \cdot \tilde{s}_t + (1-\gamma_t) \cdot \hat{S}_{t-1},$$

with some learning schedule γ_t; in (3) $\gamma_t = 1/(t - \Delta)$. This approach is slightly different from that of Cappé and Moulines (2005); see Sect. 7.1 for a discussion.

Choosing a large value of Δ allows SMC to use many observations to improve the posterior distribution of $X_{t-\Delta}$. However the cost of a large Δ is an increased path degeneracy due to the resampling procedure, which increases the sample variance. The optimal choice for Δ balances the opposing influences of the forgetting rate of the model and the collapsing rate of the resampling process due to the divergence between the proposal distribution and the posterior distribution. For the examples in this paper we chose $\Delta = 20$ as recommended by Cappé and Moulines (2005), which seems to be a reasonable choice for our models.

There are various other techniques to improve on this basic SMC method, including improved resampling schemes (Douc and Cappé 2005; Olsson et al. 2008; Doucet and Johansen 2009; Cappé et al. 2007), and choosing better sampling distributions through lookahead strategies or resample-move procedures (Pitt and Shephard 1999; Lin et al. 2013; Doucet and Johansen 2009), which are not discussed further here. Instead, in the remainder of this paper, we focus on the process of updating the parameter estimates $\hat{\theta}_t$. The remainder of this section describes the options for step 3 of Algorithm 1.

2.1 Batch expectation maximization

Batch Expectation Maximization (BEM) processes the data in batches. Within a batch of size b, the parameter estimate stays constant ($\hat{\theta}_t = \hat{\theta}_{t-1}$) and the update to the sufficient statistic

$$\tilde{s}_t := \sum_i w_t(X_{1:t}^{(i)}) \cdot (Y_{t-\Delta} - X_{1:t}^{(i)}(t-\Delta))^2,$$

is collected at each iteration t. At the end of the mth batch we have $t = mb$, at which time

$$\hat{S}_t^{BEM} := \frac{1}{b} \sum_{k=(m-1)b+1}^{mb} \tilde{s}_k,$$

is our approximation of S, and $\hat{\sigma}_{v,t}^2 := \hat{S}_t^{BEM}$.

The batch size determines the convergence behavior of the estimates. For a fixed computational cost, choosing b too small will result in noise-dominated estimates and low precision, whereas choosing b too large will result in precise but inaccurate estimates due to slow convergence.

2.2 Online expectation maximization

BEM only makes use of the collected evidence at the end of each batch, missing potential early opportunities for improving parameter estimates. OEM addresses this issue by updating the parameter estimate at every iteration. The approximation of S at time t is a running average of $\{\tilde{s}_k\}_{k=\Delta+1,\ldots,t}$, weighted by a pre-specified learning schedule. The choice of learning schedule determines how quickly the algorithm "forgets" the earlier parameter estimates. In OEM at time t,

$$\hat{S}_t^{OEM} = \gamma_t \cdot \tilde{s}_t + (1-\gamma_t) \cdot \hat{S}_{t-1}^{OEM}, \tag{4}$$

where $\{\gamma_t\}_{t=1,2,\ldots}$ is the chosen learning schedule, typically of the form

$$\gamma_t = t^{-c} \tag{5}$$

for a fixed choice of $c \in (0.5, 1]$ (Cappé 2009). Note that when using lag Δ, $\gamma_t = (t - \Delta)^{-c}$ for $t \geq \Delta$. This update rule ensures that at time t, $\hat{S}^{OEM}$ is a weighted sum of $\{\tilde{s}_k\}_{k=\Delta+1,\ldots,t}$ where the term $\tilde{s}_k$ has weight

$$\eta_k^t := \gamma_k(1 - \gamma_{k+1}) \cdots (1 - \gamma_{t-1})(1 - \gamma_t). \tag{6}$$

Algorithm 2 Online Expectation Maximization for a simplified autoregressive model

```
For time $t \geq 1$:
  1. Simulate and calculate weights of new particles as outlined in Algorithm 1
  2. Collect sufficient statistic $\tilde{s}_t = \sum_{i=1}^N w_t(X_{1:t}^{(i)}) \cdot (Y_{t-\Delta} - X_{1:t}^{(i)}(t - \Delta))^2$
  3. Update running average of sufficient statistics $\hat{S}_t^{OEM} = \gamma_t \tilde{s}_t + (1 - \gamma_t)\hat{S}_{t-1}^{OEM}$
  4. Maximize expected likelihood by setting $\hat{\theta}_t := \hat{S}_t^{OEM}$
```

Although this method can outperform BEM as parameters are updated continuously, its performance remains strongly dependent on the parameter c determining the learning schedule γ_t, and a suboptimal choice can reduce performance by orders of magnitude. At one extreme, the estimates will depend strongly only on the most recent data, resulting in noisy parameter estimates and low precision. At the other extreme, the estimates will average out stochastic effects but be severely affected by false initial estimates, resulting in more precise but less accurate estimates. Again, the best choice depends on the model.

A pragmatic approach to the problem of choosing a tuning parameter in OEM takes inspiration from Polyak (1990). In this method, a learning schedule that emphasizes incoming data is used to ensure quick initial convergence, while imprecise estimates are avoided at later iterations by averaging all OEM estimates beyond a threshold t_0.

$$\hat{\theta}_t^{AVG} = \begin{cases} \hat{\theta}_t^{OEM} & \text{for } t < t_0 \\ \frac{1}{t-t_0+1} \sum_{k=t_0}^t \hat{\theta}_k^{OEM} & \text{for } t \geq t_0. \end{cases}$$

Choosing an appropriate threshold t_0 can be more straightforward than choosing c for $\gamma_t = t^{-c}$, but it still requires the user to have an intuition for how the estimates for each parameter will behave. We will refer to this method as AVG, use $c = 0.6$, and set $t_0 = 50{,}000$ which is half the total iterations for our examples.

2.3 Introspective online expectation maximization

We now introduce IOEM to address the issue of having to pre-specify a learning schedule $\{\gamma_t\}_{t=1,\ldots}$. The algorithm is similar to OEM, but instead of pre-specifying γ_t, we estimate the precision and accuracy in the sufficient statistic updates $\{\tilde{s}_k\}_{k=\Delta+1,\ldots,t}$ and use these to determine γ_{t+1}. More precisely, we keep online estimates of a weighted regression on the dependent variables $\{\tilde{s}_k\}_{k=\Delta+1,\ldots,t}$ where $k-t$ serves as the (shifted) explanatory variable:

$$\tilde{s}_k = \beta_0 + \beta_1(k-t) + \epsilon_k \tag{7}$$

where $\epsilon_k \sim N(0,\sigma^2)$, and data point $(k-t, \tilde{s}_k)$ has weight (6) as before. This weighted regression results in intercept and slope estimates $\hat{\beta}_0$, $\hat{\beta}_1$ and variance estimates $\hat{\sigma}_0^2$, $\hat{\sigma}_1^2$, where at convergence $\hat{\beta}_0$ is the sought-after estimate and $\hat{\beta}_1 \simeq 0$. We do not use standard weighted regression, in which weights are inversely proportional to the variance of the observation, as this assumption is not justified here and would lead to biased estimates of $\hat{\sigma}_{0,1}^2$. Instead we assume that observations share an unknown variance, and we use the weights to modulate the influence of each observation to the regression estimates, to reduce the impact of the bias in earlier observations; see Sect. 7.2 for details.

We next use the regression coefficients to estimate the past iteration where the drift term $|\hat{\beta}_1|(k-t)$ is of the same order as the uncertainty $\hat{\sigma}_0$ in the main estimate $\hat{\beta}_0$:

$$t-k = \alpha\frac{\hat{\sigma}_0}{|\hat{\beta}_1| + \hat{\sigma}_1}, \tag{8}$$

where $\hat{\sigma}_1$ ensures that division by zero does not occur, and α tunes the algorithms's sensitivity to model misfit due to underlying parameter changes; we use $\alpha = 1$ unless stated otherwise. We propose a learning rate γ_{t+1}^{reg} that results in a characteristic forgetting time $1/\gamma_{t+1}^{reg}$ matching this distance:

$$\gamma_{t+1}^{reg} = \frac{|\hat{\beta}_1| + \hat{\sigma}_1}{\alpha\hat{\sigma}_0}. \tag{9}$$

This choice ensures that a substantial slope estimate $|\hat{\beta}_1|$ indicating that $\hat{\beta}_0$ has low accuracy puts large weight on the incoming statistic, improving accuracy, whereas a large $\hat{\sigma}_0$ reflecting low precision in estimate $\hat{\beta}_0$ results in a small weight, smoothing out successive estimates and improving precision. We impose restrictions on γ_{t+1} which keep it between the most extreme valid learning schedules for OEM. Taken together, the update step for γ becomes

$$\gamma_{t+1} = \min\left((t+1)^{-c}, \max\left(\gamma_{t+1}^{reg}, \gamma_t/(1+\gamma_t)\right)\right) \tag{10}$$

where $c > 0.5$ is chosen to be very close to 0.5 and guarantees convergence. These restrictions ensure that our algorithm satisfies the assumptions of Theorem 1 of Cappé and Moulines (2009), namely that $0 < \gamma_t < 1$, $\sum_{t=1}^{\infty}\gamma_t = \infty$, and $\sum_{t=1}^{\infty}\gamma_t^2 < \infty$. Hence for any model for which f and g satisfy the assumptions guaranteeing convergence of the standard OEM estimator, the IOEM algorithm is also guaranteed to converge. The precise conditions are detailed in Assumption 1, Assumption 2, and Theorem 1 of Cappé and Moulines (2009).

Algorithm 3 Introspective Online Expectation Maximization for a simplified autoregressive model

```
For time $t \geq 1$:
  1. Simulate and calculate weights of particles using SMC with parameter $\hat{\theta}_{t-1}$
  2. Collect sufficient statistic $\tilde{s}_t = \sum_{i=1}^{N} w_t(X_{1:t}^{(i)}) \cdot (Y_{t-\Delta} - X_{1:t}^{(i)}(t-\Delta))^2$
  3. Maximize expected likelihood by setting $\hat{\theta}_t = \hat{S}_t^{IOEM} := \gamma_t \cdot \tilde{s}_t + (1-\gamma_t) \cdot \hat{S}_{t-1}^{IOEM}$
  4. Perform weighted regression on $\tilde{s}$ to calculate $\gamma_{t+1}$ via (9-10).
```

3 The IOEM algorithm for the full autoregressive model

The adapting learning schedule $\{\gamma_t\}_{t=1,\ldots}$ sets IOEM apart from OEM. However, the way γ_t is calculated in Algorithm 3 only works in the special case that a single sufficient statistic and the single parameter of interest coincide (here, $\hat{\sigma}_{v,t}^2 = \hat{S}_t$). In general, the sufficient statistics $\hat{S}$ are mapped to parameter estimates $\hat{\theta}$ by a function Λ, leading to a more involved setup that we explore here. To this end, we now consider the full noisily-observed autoregressive model AR(1) with master equations as in (1), but now with unknown parameters a, σ_w, and σ_v. We define four sufficient statistics,

$$S_{1,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=1}^{t-1} X_k^2\right],$$

$$S_{2,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=1}^{t-1} X_k \cdot X_{k+1}\right],$$

$$S_{3,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=2}^{t} X_k^2\right],$$

$$S_{4,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t}\sum_{k=1}^{t}(Y_k - X_k)^2\right].$$

Then, in BEM and OEM, we update the parameter estimates to

$$\hat{a}_t = \hat{S}_{2,t}/\hat{S}_{1,t}, \tag{11}$$

$$\hat{\sigma}_{w,t} = (\hat{S}_{3,t} - (\hat{S}_{2,t})^2/\hat{S}_{1,t})^{1/2}, \tag{12}$$

$$\hat{\sigma}_{v,t} = (\hat{S}_{4,t})^{1/2}, \tag{13}$$

where $\hat{S}_t$ is an approximation of S_t.

In most cases, as above, the function Λ mapping $\hat{S}_t$ to $\hat{\theta}_t$ is nonlinear, and requires multiple sufficient statistics as input. To avoid bias, we want all sufficient statistics that inform one parameter estimate to share a learning schedule $\{\gamma_t\}_{t=1,2,\ldots}$. We therefore estimate an adapting learning schedule for each parameter independently, by performing the regression on the level of the parameter estimates (Algorithm 4), rather than on the level of the sufficient statistics. We will calculate $\hat{S}_t$ as in OEM (4) using our adapting learning schedule instead of a user specified learning schedule. Because the adapting learning schedule is specific to each parameter, we will have multiple estimates of certain summary sufficient statistics. In this case $S_{1,t}$ and $S_{2,t}$ are estimated by $\hat{S}^a_{1,t}$ and $\hat{S}^a_{2,t}$ for (11) and by $\hat{S}^{\sigma_w}_{1,t}$ and $\hat{S}^{\sigma_w}_{2,t}$ for (12).

Simply regressing on $\hat{\theta}_{1:t}$ with respect to t would correspond to regression on $\hat{S}_{1:t}$, not $\tilde{s}_{1:t}$. As $\hat{S}$ is a running average, there is a strong correlation between $\hat{S}_{t-1}$ and $\hat{S}_t$ and hence also a strong dependence between $\hat{\theta}_{t-1}$ and $\hat{\theta}_t$. In order to perform the regression on the parameters we must "unsmooth" $\hat{\theta}_{1:t}$ to create pseudo-independent parameter updates $\tilde{\theta}_t$ (see Algorithm 4). This is accomplished by taking linear combinations,

$$\tilde{\theta}_t := \frac{1}{\gamma_t} \cdot \hat{\theta}_t + \left(1 - \frac{1}{\gamma_t}\right) \cdot \hat{\theta}_{t-1},$$

where the coefficients are chosen so as to minimize the covariance between successive updates, justifying the term pseudo-independent. The resulting updates correspond with the unsmoothed sufficient statistics updates $\tilde{s}_t$ used in Sect. 2.3. See Sect. 7.3 for further details on this step.

Algorithm 4 Introspective Online Expectation Maximization (general case)

```
For time $t \geq 1$:
1. Simulate and calculate weights of new particles using SMC with parameter $\hat{\theta}^{IOEM}_{t-1}$
2. Collect sufficient statistics $\tilde{s}_t$
3. Update running average of sufficient statistics
   $\hat{S}_t = \gamma_t \tilde{s}_t + (1 - \gamma_t)\hat{S}_{t-1}$
4. Maximize expected likelihood by setting $\hat{\theta}_t = \Lambda(\hat{S}_t)$
5. Create pseudo-independent parameter updates
   $\tilde{\theta}_t = \frac{1}{\gamma_t} \cdot \hat{\theta}_t + (1 - \frac{1}{\gamma_t}) \cdot \hat{\theta}_{t-1}$
6. Perform weighted regression on $\tilde{\theta}$ to calculate $\gamma_{t+1}$
```

4 Simulations

We performed inference on different models using the BEM, OEM and IOEM algorithms as described above. For BEM we used batch sizes from 100 to 10,000, and for OEM we used learning schedules $\gamma_t = t^{-c}$ with c ranging from 0.6 to 0.9. In all cases the bootstrap filter was run with $N = 100$ particles, and the algorithm was run from $t = 1$ to $t = 100{,}000$. For all parameter choices, 100 independent replicates were generated, and we show the distribution of inferred parameter values across these replicates.

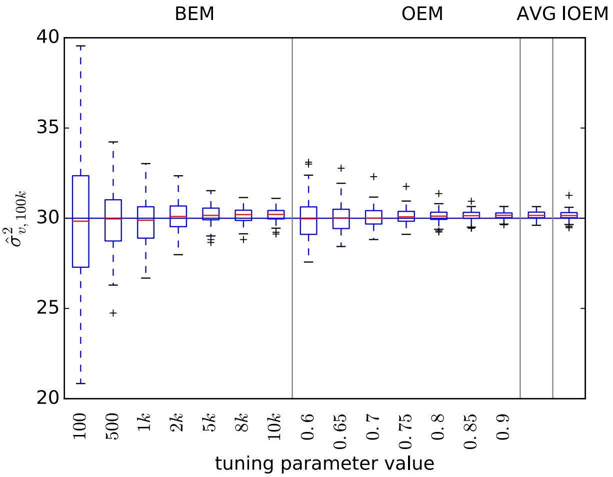

Fig. 1 Comparison of EM methods on simplified AR model with known true parameters $a = .95$, $\sigma_w = 1$, and unknown true $\sigma_v^2 = 30$, and initial parameter estimate $\sigma_{v,0}^2 = 20$. $\hat{\sigma}^2_{v,100k}$ is plotted for 100 replicates, $N = 100$

4.1 Inference with the simplified IOEM algorithm

We first applied the simplified IOEM algorithm (Algorithm 3) to the problem of inferring σ_v^2 in model (1), with all other parameters assumed known, and compared the results with the BEM and OEM algorithms (Fig. 1). The choice of tuning parameter in BEM and OEM makes a significant difference to the precision of the estimate even after 100,000 observations. IOEM was able to recognize that behavior similar to BEM with $b = 10{,}000$ or OEM with $c = 0.9$ was optimal. The accuracy and precision of IOEM are comparable with those of the post-OEM averaging technique (AVG) with parameters $c = 0.6$ and $t_0 = 50{,}000$.

4.2 Inference with the complete IOEM algorithm

We next treated all four parameters of the AR(1) model (1) as unknown, and inferred them using the full IOEM algorithm (Algorithm 4). Estimates for the a parameter under different EM methods are presented in Fig. 2; for the other parameter inferences see Sect. 7.5, Fig. 6.

In the AR(1) model, IOEM outperforms most other EM methods when estimating the a parameter, while AVG for the chosen parameter settings ($c = 0.6$, $t_0 = 50{,}000$) provides slightly more precise estimates at similar accuracy. It is worth noting that in this case, OEM with $c = 0.6$ substantially outperforms OEM with $c = 0.9$, in contrast to the results shown in Fig. 1. This is a result of the bad initial estimates. OEM with $c = 0.6$ forgets the earlier simulations much faster than OEM with $c = 0.9$ and hence is able to move its estimates of a, σ_w, and σ_v much more quickly. Here IOEM recognizes that it should have similar behavior to OEM with $c = 0.6$, whereas in the inference displayed in Fig. 1 IOEM chose behavior similar to OEM with $c = 0.9$. IOEM can indeed adapt to the model.

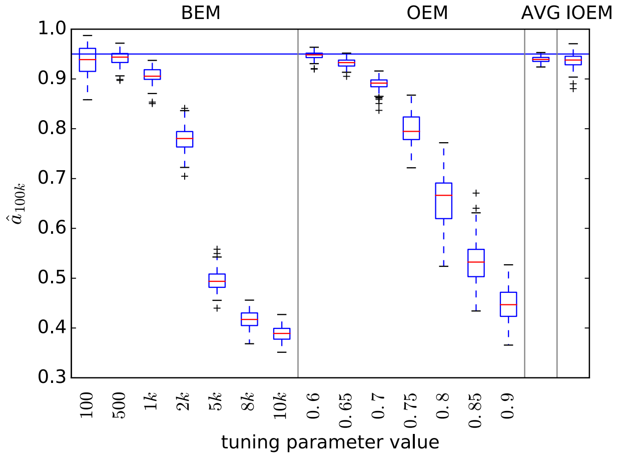

Fig. 2 Comparison of EM methods on full autoregressive model with unknown true parameters $a = 0.95$, $\sigma_w = 1$, $\sigma_v = 5.5$ and initial parameters $a_0 = 0.8$, $\sigma_{w,0} = 3$, $\sigma_{v,0} = 1$. $\hat{a}_t$ at $t = 100{,}000$ is plotted for 100 replicates, $N = 100$

4.3 Inference of multiple parameters

Next we investigated a model with a larger number of parameters and varying accuracy of initial parameter estimates. One of the advantages of the IOEM algorithm over OEM is its ability to adapt to each parameter independently. To highlight this, we applied IOEM to a simple 2-dimensional autoregressive model. For this model we consider the sequences $\{Y^A, Y^B\}_{1:t}$ as observed, while $\{X^A, X^B\}_{1:t}$ are unobserved, where

$$\begin{aligned} X_t^A &= a^A X_{t-1}^A + \sigma_w^A W_t^A, & X_t^B &= a^B X_{t-1}^B + \sigma_w^B W_t^B, \\ Y_t^A &= X_t^A + \sigma_v V_t^A, & Y_t^B &= X_t^B + \sigma_v V_t^B. \end{aligned} \tag{14}$$

Note that Y^A and Y^B are uncoupled, and that their master equation have independent parameters except for a shared parameter σ_v. By giving component A good initial estimates and B bad initial estimates, we can see how the different EM methods cope with a combination of accurate and inaccurate initializations. IOEM is able to identify the set with good initial estimates (a^A, σ_w^A) and quickly start smoothing out noise. To IOEM, the other parameters appear to not have converged (σ_w^B and σ_v because they are at the wrong value, a^B because it will be changing to compensate for σ_w^B and σ_v).

Figure 3 shows the inference of σ_v, which due to its dependence on components A and B, suffers the most from a blanket choice of tuning parameter in BEM or OEM. OEM with $c = 0.6$ and OEM with $c = 0.9$ both suffer in this model as they are both well suited to parameter estimation in one of the components, but not the other. AVG provides precise but biased estimates in this case, because of its reliance on a fast-forgetting initial OEM stage which again is suited to only one of the model components. IOEM on the other hand is able to capture the best of both worlds, striving for precision in component A and initially foregoing precision in favour of accuracy in component B.

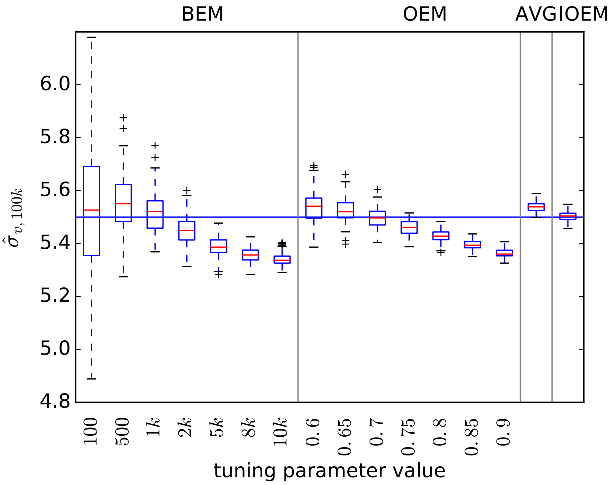

Fig. 3 Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma_w^A = 1, \sigma_v = 5.5, a^B = 0.95, \sigma_w^B = 1$ and initial parameters $a_0^A = 0.95, \sigma_{w,0}^A = 1, \sigma_{v,0} = 3, a_0^B = 0.95$, $\sigma_{w,0}^B = 3$. $\hat{\sigma}_{v,t}$ at $t = 100{,}000$ is plotted for 100 replicates, $N = 100$

The inference of the other parameters and comparisons with a different choice of AVG threshold are shown in Sect. 7.5, Figs. 7, 8, 9, 10.

4.4 Inference of parameters of a stochastic volatility model

The previous sections have demonstrated IOEM is comparable to choosing the optimal tuning parameter in OEM or BEM in certain models. However, the models shown have all been based on the noisily observed autoregressive model, which is a linear Gaussian case where in practice analytic techniques would be preferred over SAEM. We now examine the behaviour of these algorithms when inferring the parameters of a non-linear stochastic volatility model defined by transition and emission densities

$$f(x_t|x_{t-1}) = (2\pi\sigma^2)^{-1/2} \exp\left\{ - \frac{(x_t - \phi x_{t-1})^2}{2\sigma^2} \right\}, \tag{15}$$

$$g(y_t|x_t) = (2\pi\beta^2 e^{x_t})^{-1/2} \exp\left\{ - \frac{1}{2\beta^2 e^{x_t}} y_t^2 \right\}. \tag{16}$$

We define four summary sufficient statistics,

$$S_{1,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1} \sum_{k=1}^{t-1} X_k \cdot X_{k+1} \right],$$

$$S_{2,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1} \sum_{k=1}^{t-1} X_k^2 \right],$$

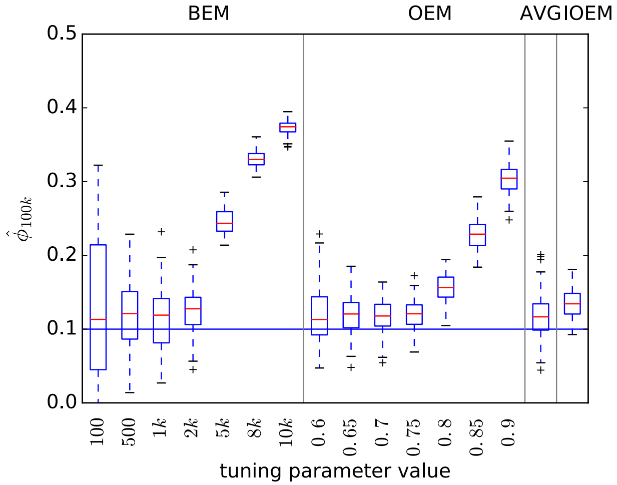

Fig. 4 Estimates of ϕ in stochastic volatility model

$$S_{3,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t-1}\sum_{k=2}^{t} X_k^2\right],$$

$$S_{4,t} = \mathbb{E}_{X_{1:t}|Y_{1:t},\theta}\left[\frac{1}{t}\sum_{k=1}^{t} e^{-X_k}\cdot Y_k^2\right].$$

Then the set of parameters that maximises the likelihood at step t are

$$\hat{\phi}_t = \hat{S}_{1,t}/\hat{S}_{2,t}, \tag{17}$$

$$\hat{\sigma}_t = (\hat{S}_{3,t} - (\hat{S}_{1,t})^2/\hat{S}_{2,t})^{1/2}, \tag{18}$$

$$\hat{\beta}_t = (\hat{S}_{4,t})^{1/2}, \tag{19}$$

Again IOEM results in similar estimates to the optimal BEM/OEM and the online averaging technique with a well-chosen threshold (see Fig. 4 and Sect. 7.5, Fig. 11).

5 Application to financial time series

We next applied our approach to daily log returns for US dollar to UK pound exchange rates, obtained from `oanda.com`. Between 18/05/2010 to 2/3/2016, roughly the period between the 2010 flash crash and the Brexit referendum, rates were fairly stable and might be described by an ARMA(1,1) model equivalent to (1). To assess confidence in estimates, we independently inferred model parameters 24 times from day-on-day log returns measured at every full hour (Fig. 5). We note that these time series are not fully comparable due to intraday seasonalities (Cornett et al. 1995), an effect that may be expected to increase the observed variation between the 24 time

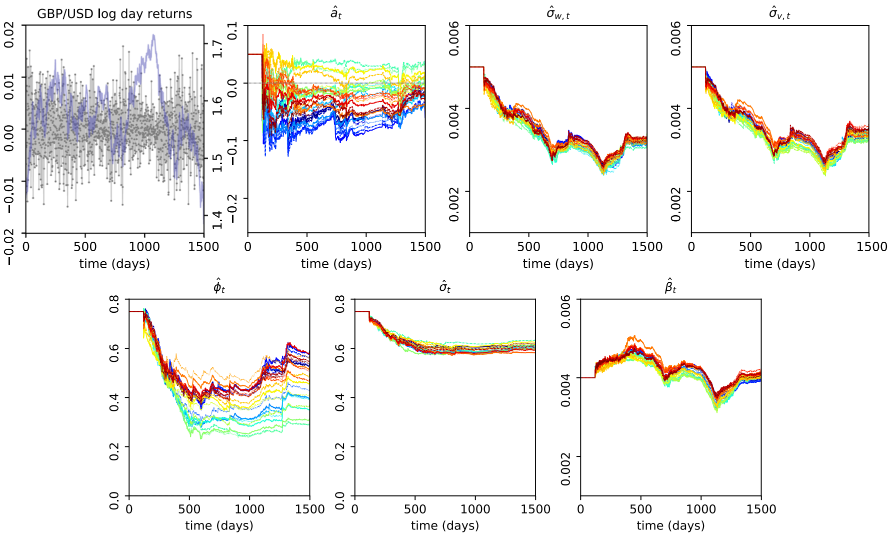

Fig. 5 Running estimates of parameters of model (1) (top) and model (16) (bottom) on time series of daily GBP/USD log returns (top left panel) based on 24 different hourly offsets (colours) using IOEM ($N = 5000$). We used $\alpha = 2$ to reduce the impact of underlying parameter changes; for results with $\alpha = 1$ see Figs. 15 and 16 (color figure online)

series, which would lead to conservative confidence estimates. Results suggest a weak negative correlation of successive daily log returns ($a < 0$), which is supported by a direct fit of an ARMA(1,1) model to the data (Fig. 12). Although the ARMA(1,1) model assumes fixed parameters and in particular constant volatility, running inferences strongly indicate volatility variations (Figs. 5 and 13), suggesting model (16) might be appropriate. Inferred values of ϕ indicate substantial day-to-day inertia in volatility. Running estimates of parameters are fairly constant in time, although those for β show that the model has difficulty tracking the two sudden drops in volatility that occurred in this period, indicating model misfit.

6 Conclusion

Stochastic Approximation EM is a general and effective technique for estimating parameters in the context of SMC. However, convergence can be slow, and improving convergence speed is of particular interest in this setting. We have shown that IOEM produces accurate and precise parameter estimates when applied to continuous state-space models. Across models, and across varying levels of accuracy of the initial estimates, the efficiency of IOEM matches that of BEM/OEM with the optimal choice of tuning parameter. The AVG procedure also shows good behaviour, but like BEM/OEM it has tuning parameters, and when these are chosen suboptimally performance is not as good as IOEM (Figs. 9 and 10). BEM/OEM/AVG all make use of a single learning schedule $\{\gamma_t\}$, and for more complex models a single learning schedule generally cannot achieve optimal convergence rates for all parameters, as we have shown for the 2-dimensional AR example. In addition, AVG works by post-hoc averaging of noisy estimates, and since the inferences depend on the noisy estimates themselves, this implicitly relies on the model being sufficiently linear around the true parameter value. We expect IOEM to be more resilient to strong nonlinearities than AVG, but we have not explored this idea further here.

IOEM finds parameter-specific learning schedules, resulting in better performance than standard methods with a single learning rate parameter are able to achieve. IOEM can be applied with minimal prior knowledge of the model's behavior, and requires no user supervision, while retaining the convergence guarantees of BEM/OEM, therefore providing an efficient, practical approach to parameter estimation in SMC methods. While not the focus of this paper, application to a financial time series suggests that IOEM may be useful in informally assessing model fit; it would be interesting to investigate whether this could be made rigorous.

Acknowledgements This work was supported by Wellcome Trust grants 090532/Z/09/Z and 102423/Z/13/Z, and Medical Research Council Strategic Alliance Funding MC_UU_12025.

7 Appendix

7.1 Fixed-lag technique

Our fixed-lag technique is slightly different than that proposed in the literature (Cappé and Moulines 2005; Olsson et al. 2008). Compared to the existing approach it uses less intermediate storage. Recall that the approximation we aim to evaluate is

$$\hat{S}_t = \sum_i w_t(X_{1:t}^{(i)}) \cdot \sum_{u=1}^{t} s_u(X_{1:t}^{(i)}(u), Y(u)),$$

where the sufficient statistic is written explicitly as a sum over the path traced out by the particle $X_{1:t}^{(i)}$. The drawback is that for $u \ll t$ the paths will have collapsed due to resampling, increasing the variance for those contributions to S. The solution proposed in Cappé and Moulines (2005) is to use instead the approximation

$$\hat{S}_t \approx \sum_i \left(\sum_{u=1}^{t-\Delta} w_{u+\Delta}(X_{1:u+\Delta}^{(i)}) s_u(X_{1:u+\Delta}^{(i)}(u), Y(u)) \right.$$
$$\left. + \; w_t(X_{1:t}^{(i)}) \sum_{u=t-\Delta+1}^{t} s_u(X_{1:t}^{(i)}(u), Y(u)) \right).$$

This requires storing the quantities

$$\{s_u(X^{(i)}_{1:u+\Delta}(u)), Y(u)\}_{u=t-\Delta,\ldots,t}$$

for each sufficient statistic and each particle. This storage can be expensive if large numbers of sufficient statistics are tracked. Instead, at iteration t we use the approximation

$$\hat{S}_t \approx \sum_{u=1}^{t-\Delta} \sum_i w_{u+\Delta}(X^{(i)}_{1:u+\Delta}) s_u(X^{(i)}_{1:u+\Delta}(u), Y(u)).$$

By disregarding terms involving s_u for $u > t - \Delta$ and switching the summation in this way, we can now update $\hat{S}$ at each iteration by adding the contribution of the current particles to a single summary statistic at a distance Δ, without requiring per-particle storage other than each particle's recent history.

7.2 Weighted regression

The term "weighted regression" usually refers to regression where the errors are independent and normally distributed with zero mean and known variance (up to a multiplicative constant), and the data is weighted inversely proportionally to its variance. In our case, the data is assumed to drift, contributing an additional, non-independent term to the error. Weights are used to only focus on recent data where the drift contributes an error of the same order of magnitude as the normally distributed noise, while discounting the impact of data points further away. In this setup we are interested both in estimating the regression coefficients, and the error in these estimates.

Perry Kaufman's adaptive moving average (AMA) (Kaufman 1995) is a similar averaging technique which reacts to the trends and volatility (jointly referred to as the behavior) of the sequence. The difference lies in the measure of the behavior. AMA relies on a user specified window length n. The n most recent data points are used to measure the behavior. This would be equivalent to using equally-weighted linear regression over the last n points. By using weighted regression, the contribution of points to the behavior measures is also influenced by the previously observed behavior. For example, a sharp trend will effectively employ a smaller n value as we have lost interest in the behavior before that trend.

Let X be the $2 \times n$ matrix consisting of a column of 1s and a column with the dependent variable, let y be the vector of observations, let β be the two coefficients, and ϵ the vector of errors, with $\epsilon_k \sim N(0, \sigma^2)$. Finally let w be a vector of weights. We estimate β by minimizing

$$s^2 = (X_w\beta - y_w)^\top (X_w\beta - y_w),$$

where X_w and y_w are defined as

$$X_w := \begin{bmatrix} w_1 & w_1 \cdot (-n+1) \\ \vdots & \vdots \\ w_n & w_n \cdot 0 \end{bmatrix}; \qquad y_w := \begin{bmatrix} w_1 \cdot y_1 \\ \vdots \\ w_n \cdot y_n \end{bmatrix}.$$

Setting the derivative $\partial s^2/\partial\beta = 2(X_w\beta - y_w)^\top X_w$ to zero and solving for β results in weighted regression estimator $\hat{\beta} = (X_w^\top X_w)^{-1}X_w^\top y_w$, or explicitly

$$\hat{\beta}_1 = \frac{\left(\sum w_k^2 x_{2k} y_k\right) - \left(\sum w_k^2 x_{2k}\right)\left(\sum w_k^2 y_k\right)}{\left(\sum w_k^2 x_{2k}^2\right) - \left(\sum w_k^2 x_{2k}\right)^2},$$
$$\hat{\beta}_0 = \frac{\left(\sum w_k^2 x_{2k}^2\right)\left(\sum w_k^2 y_k\right) - \left(\sum w_k^2 x_{2k} y_k\right)\left(\sum w_k^2 x_{2k}\right)}{\left(\sum w_k^2 x_{2k}^2\right) - \left(\sum w_k^2 x_{2k}\right)^2}.$$

From this expression we can see that $\hat{\beta}$ can be updated in an online manner as k increases simply by updating the above summations. The variance in $\hat{\beta}$ can be estimated as follows:

$$\begin{aligned}\operatorname{var}\hat{\beta} &= \operatorname{var}(X_w^\top X_w)^{-1}X_w^\top y_w\\ &= E\left[(X_w^\top X_w)^{-1}X_w^\top \epsilon_w \epsilon_w^\top X_w (X_w^\top X_w)^{-1}\right]\\ &= (X_w^\top X_w)^{-1}X_w^\top \operatorname{diag}(w_k^2\sigma^2)X_w(X_w^\top X_w)^{-1}.\end{aligned}$$

If $w_k^2 = 1$ this simplifies to the usual $\operatorname{var}\hat{\beta} = \sigma^2(X^\top X)^{-1}$. Writing out the expression for $\operatorname{var}\hat{\beta}$ explicitly shows that it is again possible to find online updates for the relevant terms.

7.3 Pseudo-independent parameter updates

In order to perform our regression on the level of the parameters, we need to map from $\tilde{s}^{(t)}$ to $\hat{S}^{(t)}$ and then to $\hat{\theta}^{(t)}$. We do not wish to regress on $\hat{\theta}^{(1:t)}$, as $\hat{\theta}^{(t-1)}$ and $\hat{\theta}^{(t)}$ are highly correlated. Instead we want a sequence defined in the parameter space where the correlations resemble those in $\tilde{s}^{(1:t)}$. We define this sequence as

$$\tilde{\theta}_t := \frac{1}{\gamma_t}\hat{\theta}_t + \left(\frac{\gamma_t - 1}{\gamma_t}\right)\hat{\theta}_{t-1}.$$

Here we show that $\tilde{\theta}_i$ and $\tilde{\theta}_j$ are uncorrelated for all $i \neq j$, under the assumption that $\tilde{s}_i$ and $\tilde{s}_j$ are uncorrelated ($i \neq j$). Define $\{\eta_k^t\}_{k=0,\dots,t}$ to be the sequence that satisfies $\hat{S}_t = \sum_{k=0}^t \eta_k^t \tilde{s}_k$ and $\sum_{k=0}^t \eta_k^t = 1$. Note that $\eta_t^t = \gamma_t$, $\eta_{t-1}^t = \gamma_{t-1}(1-\gamma_t)$, and so on. Now,

$$\begin{aligned}\operatorname{cov}(\tilde{\theta}_i, \tilde{\theta}_j) &= \operatorname{cov}\left(\frac{1}{\gamma_i}\hat{\theta}_i + \frac{\gamma_i - 1}{\gamma_i}\hat{\theta}_{i-1}, \frac{1}{\gamma_j}\hat{\theta}_j + \frac{\gamma_j - 1}{\gamma_j}\hat{\theta}_{j-1}\right)\\ &= \frac{1}{\gamma_i\gamma_j}\operatorname{cov}(\hat{\theta}_i, \hat{\theta}_j)\\ &\quad + \frac{1}{\gamma_j}\left(1 - \frac{1}{\gamma_i}\right)\operatorname{cov}(\hat{\theta}_{i-1}, \hat{\theta}_j)\end{aligned}$$

$$+\frac{1}{\gamma_i}\left(1-\frac{1}{\gamma_j}\right)\operatorname{cov}(\hat{\theta}_i,\hat{\theta}_{j-1})$$
$$+\left(1-\frac{1}{\gamma_i}\right)\left(1-\frac{1}{\gamma_j}\right)\operatorname{cov}(\hat{\theta}_{i-1},\hat{\theta}_{j-1}). \tag{20}$$

Writing $\hat{\theta}_i = f_0 + f_1 \sum_{k=0}^{i} \eta_k^i \tilde{s}_k$ and recalling that

$$\operatorname{cov}(\tilde{s}_i,\tilde{s}_j)=\begin{cases}0, & \text{if } i \neq j\\ \sigma_i^2, & \text{if } i=j,\end{cases}$$

it follows that

$$\begin{aligned}\operatorname{cov}(\hat{\theta}_i,\hat{\theta}_j) &= \operatorname{cov}\left(f_1\sum_{k=0}^{i}\eta_k^i\tilde{s}_k,\ f_1\sum_{k=0}^{j}\eta_k^j\tilde{s}_k\right)\\ &= \sum_{k=0}^{i} f_1^2\eta_k^i\eta_k^j\sigma_i^2,\end{aligned}$$

for $i < j$. Substituting into the four terms of (20) yields

$$\begin{aligned}\operatorname{cov}(\tilde{\theta}_i,\tilde{\theta}_j) = &\frac{1}{\gamma_i\gamma_j}\sum_{k=0}^{i} f_1^2\eta_k^i\eta_k^j\sigma_k^2\\ &+\frac{1}{\gamma_j}\left(\frac{\gamma_i-1}{\gamma_i}\right)\sum_{k=0}^{i-1} f_1^2\eta_k^{i-1}\eta_k^j\sigma_k^2\\ &+\frac{1}{\gamma_i}\left(\frac{\gamma_j-1}{\gamma_j}\right)\sum_{k=0}^{i} f_1^2\eta_k^i\eta_k^{j-1}\sigma_k^2\\ &+\left(\frac{\gamma_i-1}{\gamma_i}\right)\left(\frac{\gamma_j-1}{\gamma_j}\right)\sum_{k=0}^{i-1} f_1^2\eta_k^{i-1}\eta_k^{j-1}\sigma_k^2.\end{aligned}$$

If we define

$$\begin{aligned}a &:= f_1^2\eta_i^i\eta_i^{j-1}\sigma_i^2,\\ b &:= \sum_{k=0}^{i-1} f_1^2\eta_k^{i-1}\eta_k^{j-1}\sigma_k^2,\end{aligned}$$

and note that

$$\eta_k^j = (1-\gamma_j)\eta_k^{j-1} \quad \text{for all } k < j,$$

then

$$\begin{aligned}\operatorname{cov}(\tilde{\theta}_i,\tilde{\theta}_j) = &\frac{1}{\gamma_i\gamma_j}(1-\gamma_j)a + \frac{1}{\gamma_i\gamma_j}(1-\gamma_i)(1-\gamma_j)b\\ &+\frac{1}{\gamma_j}\left(\frac{\gamma_i-1}{\gamma_i}\right)(1-\gamma_j)b\end{aligned}$$

$$
\begin{aligned}
&+ \frac{1}{\gamma_i}\left(\frac{\gamma_j - 1}{\gamma_j}\right) a + \frac{1}{\gamma_i}\left(\frac{\gamma_j - 1}{\gamma_j}\right)(1-\gamma_i)b \\
&+ \left(\frac{\gamma_i - 1}{\gamma_i}\right)\left(\frac{\gamma_j - 1}{\gamma_j}\right) b \\
&= 0.
\end{aligned}
$$

Hence, if $\tilde{s}_i$ and $\tilde{s}_j$ are independent for all $i \neq j$, then $\tilde{\theta}_i$ and $\tilde{\theta}_j$ are uncorrelated $(i \neq j)$, justifying the term "pseudo-independent updates" for $\tilde{\theta}_i$.

7.4 Notation reference

See Table 1.

Table 1 Notation used in this paper

Notation	Meaning	Associated methods
θ	True parameter	All
$\hat{\theta}_t$	Parameter estimate at time t	All
$\tilde{\theta}_t$	Pseudo-independent parameter update	IOEM
$\tilde{s}_t$	Sufficient statistic update at time t	All
$\hat{S}_t$	Summary sufficient statistic from averaging $\tilde{s}$	All
N	Number of particles	All
$\varDelta$	Lag of fixed-lag technique	All
$\hat{\beta}_0$	Regression intercept ML estimate	IOEM
$\hat{\beta}_1$	Regression slope ML estimate	IOEM
$\hat{\sigma}_0^2$	Variance of regression intercept ML estimate	IOEM
$\hat{\sigma}_1^2$	Variance of regression slope ML estimate	IOEM

7.5 Figures

See Figs. 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16.

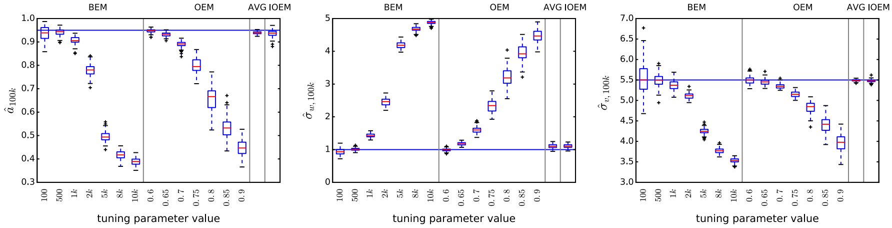

Fig. 6 Comparison of EM methods on full autoregressive model with unknown true parameters $a = 0.95$, $\sigma_w = 1, \sigma_v = 5.5$ and initial parameters $a_0 = 0.8, \sigma_{w,0} = 3, \sigma_{v,0} = 1$. Parameter estimates at $t = 100{,}000$ are plotted for 100 replicates, $N = 100$

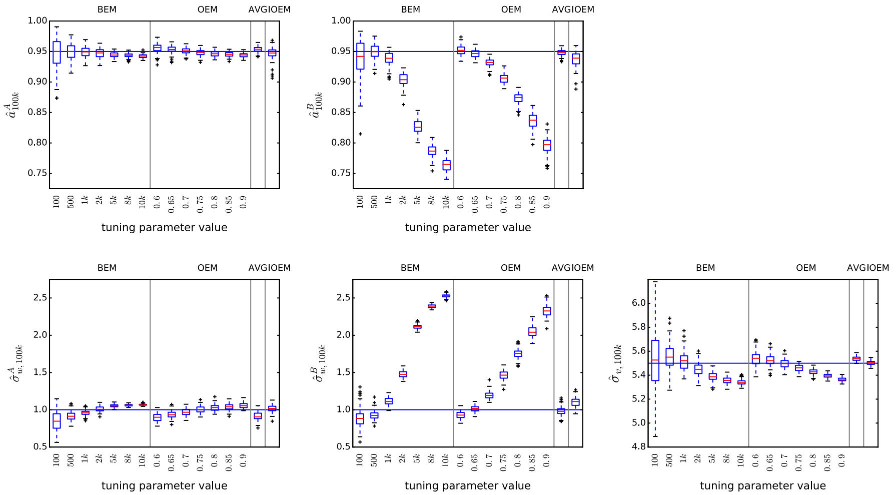

Fig. 7 Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma_w^A = 1, \sigma_v = 5.5, a^B = 0.95, \sigma_w^B = 1$ and initial parameters $a_0^A = 0.95, \sigma_{w,0}^A = 1, \sigma_{v,0} = 3, a_0^B = 0.95$, $\sigma_{w,0}^B = 3$. Parameter estimates at $t = 100{,}000$ are plotted for 100 replicates, $N = 100$

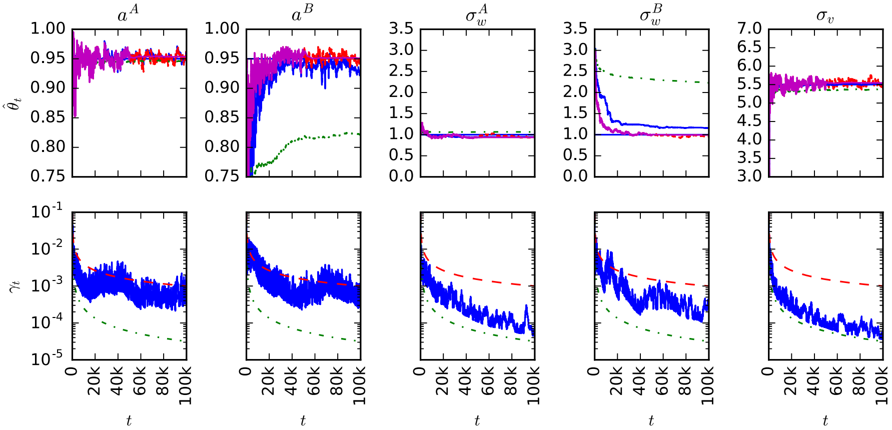

Fig. 8 Parameter-specific convergence in the 2-dimensional autoregressive model over 100,000 observations. Each column displays information for a single parameter. The top row shows the sequence of parameter estimates for three EM methods. The bottom row shows the learning schedule γ_t for the three EM methods. Blue solid line: IOEM; red dashed line: OEM with $c = 0.6$; green dash-dot line: OEM with $c = 0.9$; magenta solid line: averaged OEM technique with a threshold $t_0 = 50{,}000$ (color figure online)

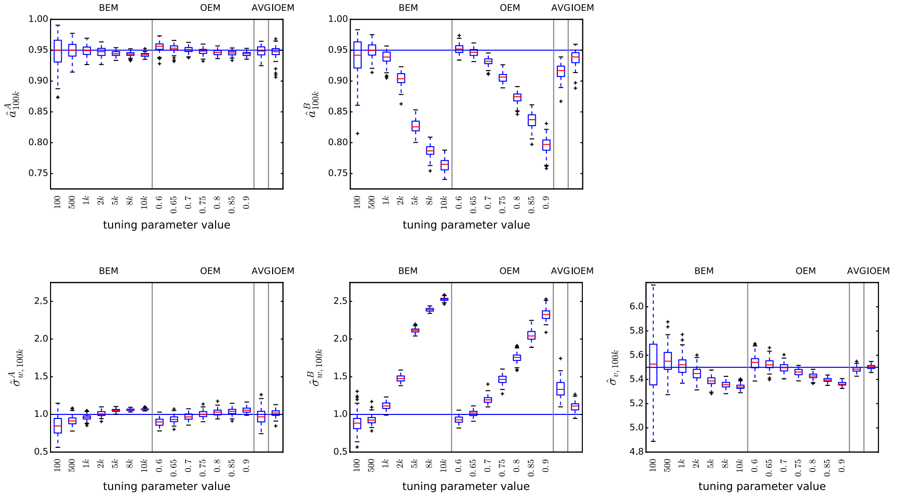

Fig. 9 Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma_w^A = 1, \sigma_v = 5.5, a^B = 0.95, \sigma_w^B = 1$ and initial parameters $a_0^A = 0.95, \sigma_{w,0}^A = 1, \sigma_{v,0} = 3, a_0^B = 0.95$, $\sigma_{w,0}^B = 3$. Parameter estimates at $t = 100{,}000$ are plotted for 100 replicates, $N = 100$

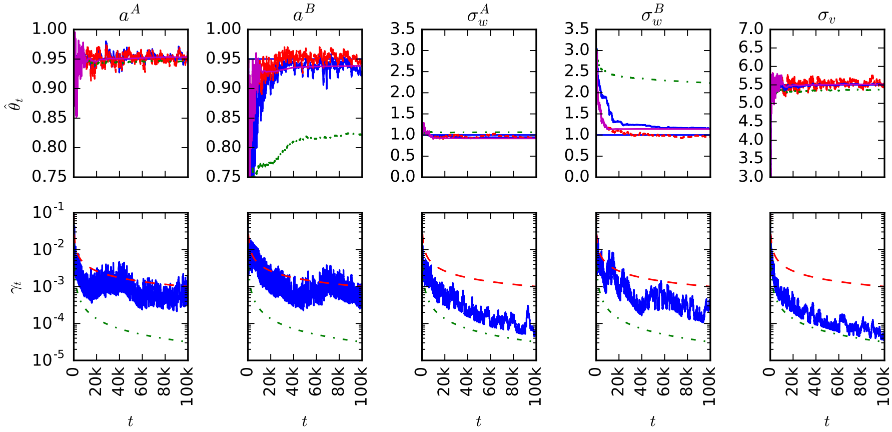

Fig. 10 Parameter-specific convergence in the 2-dimensional autoregressive model over 100,000 observations. Each column displays information for a single parameter. The top row shows the sequence of parameter estimates for four EM methods. The bottom row shows the learning schedule γ_t for the three EM methods. Blue solid line: IOEM; red dashed line: OEM with $c = 0.6$; green dash-dot line: OEM with $c = 0.9$; magenta solid line: averaged OEM technique with a threshold $t_0 = 10{,}000$ (color figure online)

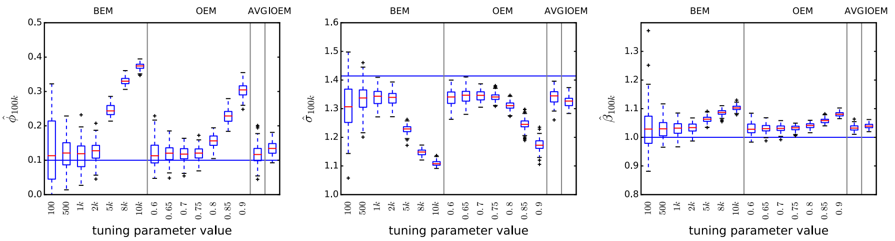

Fig. 11 Comparison of EM methods on stochastic volatility model with unknown true parameters $\phi = 0.1$, $\sigma = \sqrt{2}$, $\beta = 1$ and initial parameters $\phi_0 = 0.5$, $\sigma_0 = 1$, $\beta_0 = \sqrt{2}$. Parameter estimates at $t = 100{,}000$ are plotted for 100 replicates, $N = 100$

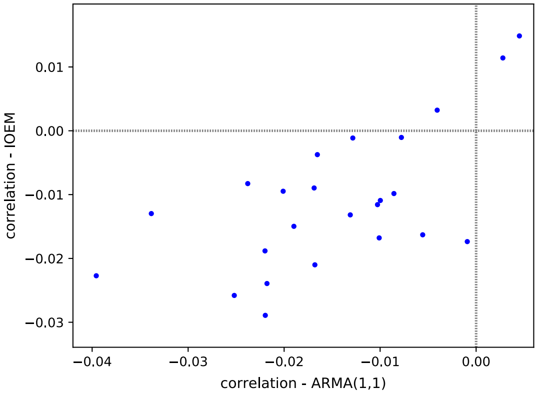

Fig. 12 Implied day-to-day correlation of log returns ρ_{IOEM}, $\rho_{ARMA(1,1)}$ for the 24 daily time series, calculated as $\rho_{IOEM} = a/(1 + (1 - a^2)\sigma_v^2/\sigma_w^2)$ and $\rho_{ARMA(1,1)} = 1 + 2\phi\theta + \theta^2/(\phi + \theta)(1 + \phi\theta)$, where the AR coefficient ϕ, a governs the asymptotic falloff in correlation in the two parameterizations, and θ is the coefficient of the moving average component. Results indicate a non-significant trend for a negative day-to-day correlation of log returns across the period studied. Inferred values of ρ across 24 different hourly offsets are themselves correlated but not identical between the two models, as expected as IOEM emphasizes recent observations over earlier ones, in contrast to the global optimization used for fitting the ARMA(1,1) model

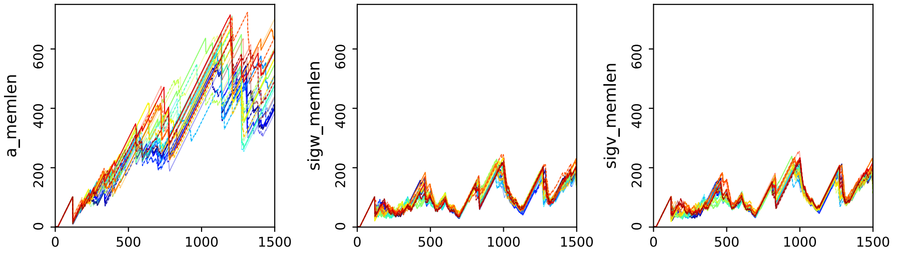

Fig. 13 Memory length (γ_t^{-1}) for the three parameters of model (1) for the 24 daily time series. The data does not appear to support a stationary model, particularly for σ_w and σ_v, as can be seen from the collapse of the memory length γ_t^{-1} particularly around $t = 650$ and $t = 1150$, coinciding with periods of low volatility

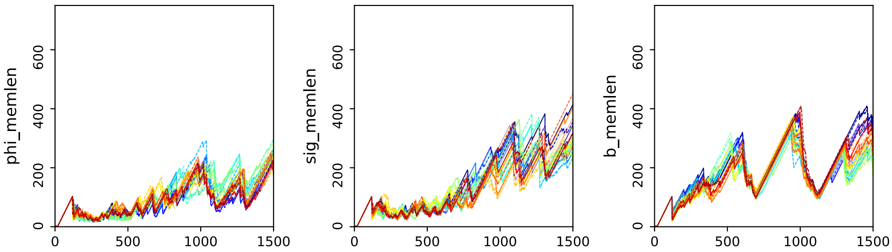

Fig. 14 Memory length (γ_t^{-1}) for the three parameters of model (16) for the 24 daily time series. The inferred parameters (see Fig. 5) appear to support a stationary model, although the collapse in memory length for parameter β due to apparent drift indicates the model's difficulty in tracking sudden changes in volatility around $t = 650$ and $t = 1150$

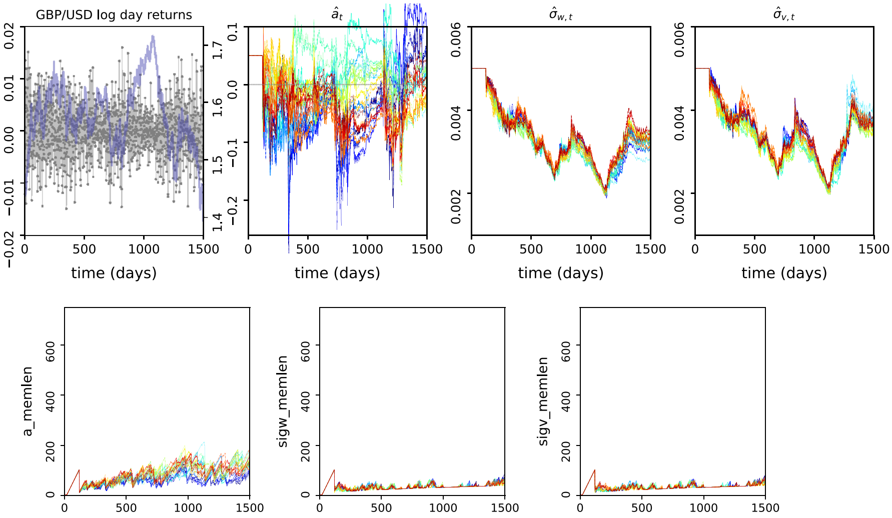

Fig. 15 Running estimates of parameters of model (1) on daily GBP/USD log returns, and memory length, inferred using $\alpha = 1$

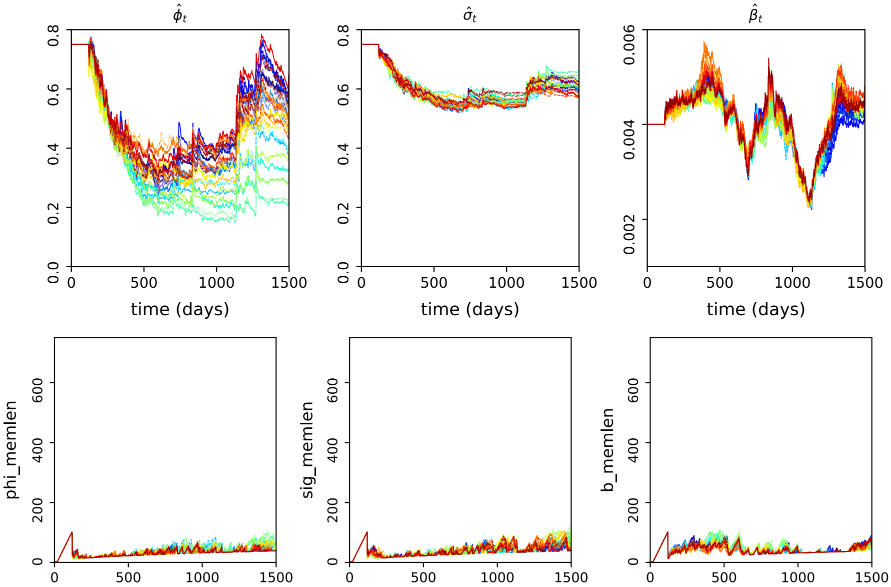

Fig. 16 Running estimates of parameters of model (16) on daily GBP/USD log returns, and memory length, inferred using $\alpha = 1$

References

Baum LE (1972) An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities 3:1–8

Bottou L (2012) Stochastic gradient descent tricks. In: Montavon G, Orr GB, Müller KR (eds) Neural networks: tricks of the trade. Lecture notes in computer science, , vol 7700, pp 421–436

Cappé O (2009) Online sequential Monte Carlo EM algorithm. In: IEEE/SP 15th Workshop on statistical signal processing, 2009. SSP'09, pp 37–40. IEEE

Cappé O, Godsill SJ, Moulines E (2007) An overview of existing methods and recent advances in sequential monte carlo. Proc IEEE 95(5):899–924

Cappé O, Moulines E (2005) On the use of particle filtering for maximum likelihood parameter estimation. In: 13th European signal processing conference, pp 1–4. IEEE

Cappé O, Moulines E (2009) On-line expectation–maximization algorithm for latent data models. J R Stat Soc Ser B 71(3):593–613

Celeux G, Chaveaux D, Diebolt J (1995) On stochastic versions of the EM algorithm. Technical Report 2514, INRIA

Celeux G, Diebolt J (1985) The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. Comput Stat Q 2:73–82

Celeux G, Diebolt J (1992) A stochastic approximation type EM algorithm for the mixture problem. Stoch Stoch Rep 2(41):119–132

Chitralekha SB, Prakash J, Raghavan H, Gopaluni R, Shah SL (2010) A comparison of simultaneous state and parameter estimation schemes for a continuous fermentor reactor. J Process Control 20(8):934–943

Cornett MM, Schwarz TV, Szakmary AC (1995) Seasonalities and intraday return patterns in the foreign currency futures market. J Bank Finance 19:843–869

Delyon B, Lavielle M, Moulines E (1999) Convergence of a stochastic approximation version of the EM algorithm. Ann Stat 27(1):94–128

Dempster AP, Laird NM, Rubin DB (1977) J R Stat Soc Ser B. Maximum likelihood from incomplete data via the EM algorithm, Soc., pp 1–38

Douc R, Cappé O (2005) Comparison of resampling schemes for particle filtering. In: Proceedings of the 4th international symposium on image and signal processing and analysis, pp 64–69. IEEE

Doucet A, de Freitas N, Gordon N (eds) (2001) Sequential Monte Carlo methods in practice. Springer

Doucet A, Johansen AM (2009) A tutorial on particle filtering and smoothing: fifteen years later. Handb Nonlinear Filter 12:656–704

Fearnhead P (2006) Efficient and exact bayesian inference for multiple changepoint problems. Stat Comput 16:203–213

Fearnhead P, Vasileiou D (2009) Bayesian analysis of isochores. J Am Stat Assoc 104:132–141

Jamshidian M, Jennrich RI (1993) Acceleration of the EM algorithm by using Quasi-Newton methods. J Am Stat Assoc 88(421):221–228

Jordan MI, Jacobs RA (1993) Hierarchical mixtures of experts and the EM algorithm. In: Proceedings of the 1993 international joint conference on neural networks, pp 1339–1344

Kantas N, Doucet A, Singh SS, Maciejowski JM (2009) An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In: 15th IFAC symposium on system identification (SYSID), vol 102, p 117

Kaufman P (1995) Smarter trading: improving performance in changing markets. McGraw-Hill, New York

Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: 3rd International conference on learning representations, ICLR 2015

Lange K (1995) A quasi Newton acceleration of the EM algorithm. Stat Sin 5:1–18

Le Corff S, Fort G (2013) Online expectation maximization based algorithms for inference in hidden Markov models. Electron J Stat 7:763–792

Lin M, Chen R, Liu JS et al (2013) Lookahead strategies for sequential Monte Carlo. Stat Sci 28(1):69–94

Lopes HF, Tsay RS (2010) Particle filters and bayesian inference in financial econometrics. J Forecast 30(1):168–209

Mandt S, Hoffman MD, Blei DM (2016) A variational analysis of stochastic gradient algorithms. In: Proceedings of the 33rd international conference on machine learning, vol 48

Mongillo G, Denève S (2008) Online learning with hidden Markov models. Neural Comput 20(7):1706–1716

Nowlan S (1991) Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures, Ph.D. thesis, School of Computer Science. Cargegie Mellon University

Olsson J, Cappé O, Douc R, Moulines E et al (2008) Sequential monte carlo smoothing with application to parameter estimation in nonlinear state space models. Bernoulli 14(1):155–179

Pitt MK, Shephard N (1999) Filtering via simulation: auxiliary particle filters. J Am Stat Assoc 94(446):590–599

Polyak BT (1990) A new method of stochastic approximation type. Avtomatika i telemekhanika 7:98–107

Reddi SJ, Kale S, Kumar S (2018) On the convergence of Adam and beyond. In: Proceedings of ICLR

Shumway RH, Stoffer DS (1982) An approach to time series smoothing and forecasting using the EM algorithm. J Time Ser Anal 3(4):253–264

Varadhan R, Roland C (2008) Simple and globally convergent methods for accelerating the convergence of any EM algorithm. Scand J Stat 35(2):335–353

Wei GCG, Tanner MA (1990) A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. J Am Stat Assoc 85(411):699–704

Xu L, Jordan M (1996) On convergence properties of the EM algorithm for gaussian mixtures. Neural Comput 8(1):129–151

Yildirim S, Singh SS, Doucet A (2013) An online expectation-maximization algorithm for changepoint models. J Comput Graph Stat 22(4):906–926

Zeiler MD (2012) ADADELTA: an adaptive learning rate method. arXiv:1212.5701