Efficient convergence through adaptive learning in sequential Monte Carlo Expectation Maximization

Donna Henderson · Gerton Lunter

27 May 2018

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 according 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. 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

Mathematics Subject Classification (2010) 65C60 · 65B99 · 62F10 · 62L12 · 62M05

Acknowledgements  This work was supported by Wellcome Trust grants 090532/Z/09/Z and 102423/Z/13/Z.
1 Introduction

Expectation Maximization (EM) is a widely used and general technique for estimating maximum likelihood parameters of a latent variable model (Dempster et al. 1977). We will be considering models with a sequential structure. Elegant algorithms are available for special cases of sequential models, such as linear systems with Gaussian noise (Shumway and Stoffer 1982), and finite-state hidden Markov models (Baum 1972). Here we focus on inference in complex models that do not admit analytic solutions, for which sequential Monte Carlo (SMC) methods are widely used to approximate the expectation in the E-step. Generally, the use of Monte Carlo methods in the context of EM is known as stochastic approximation EM (SAEM; Delyon et al. 1999) and this class of methods is favored in practice over gradient-based approaches due to their relative stability and computational efficiency when estimating high dimensional parameters (Chitralekha et al. 2010; Kantas et al. 2009).

Convergence of EM methods can nevertheless be slow for complex models and/or with large data volumes. Several authors have proposed acceleration techniques (Jamshidian and Jennrich 1993; Lange 1995; Varadhan and Roland 2008), but these require that the E-step is analytically tractable. For SAEM standard recursive EM methods are used instead, the two most popular being batch EM (BEM) and online EM (OEM). Both methods require the user to specify a tuning parameter, and in both cases the performance of the algorithm is strongly dependent on the chosen parameter. 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 equivalently the optimal learning rate in OEM, depends on the particularities of the model.

While the relative merits of these and other methods for parameter estimation have been studied in detail (see e.g. Kantas et al. 2009), the problem of choosing optimal learning rates has received relatively little attention. Here we introduce a novel algorithm, termed Introspective Online EM (IOEM), which removes the need for setting the learning rate altogether by estimating the optimal parameter-specific learning rate along with the parameters of interest. This is particularly helpful when inferring parameters in a high dimensional model, since the optimal tuning parameter may differ between parameters. 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. Sect. 2 uses a one-parameter autoregressive state-space model to introduce BEM, OEM, and a simplified version of IOEM. Sect. 3 considers the full 3-parameter autoregressive model, which requires the complete IOEM algorithm. Sect. 4 considers a 2-dimensional autoregressive model to show the benefit of the proposed algorithm when inferring many parameters. Finally, Sect. 5 demonstrates desirable performance in
the stochastic volatility model, an important case as it is nonlinear and hence more similar to applications of SAEM.

2 EM for a Simplified Autoregressive Model

Here we review SMC, BEM, OEM, and present the IOEM algorithm with a simple model. This illustrates the main concepts behind IOEM before delving into details in Sect. 3.

We consider a simple autoregressive model with one unknown parameter. We observe the sequence of random variables \( Y_{1:t} := \{Y_k\}_{k=1}^{t} \) which depends on the unobserved sequence \( X_{1:t} := \{X_k\}_{k=1}^{t} \), as follows:

\[
X_t = a X_{t-1} + \sigma_w W_t, \\
Y_t = X_t + \sigma_v V_t, \tag{1}
\]

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 \( \sigma_v^2 \) 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 \( \sigma_v^2 \) as the unknown parameter as it is the most straightforward to estimate, allowing us to introduce the idea of IOEM without certain complications which 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 so the E step amounts to the expectation of the sufficient statistics. In this model, the parameter \( \sigma_v^2 \) 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 \( \sigma_v^2 \) is obtained by setting \( \hat{\sigma}_v^2 = \hat{S}_t \). More generally, for an unknown parameter \( \theta \), \( \hat{\theta}_t = \Lambda(\hat{S}_t) \) where \( \Lambda \) 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 \( Y_{1:t} \), the
simulation of \(X^{(i)}_t\) 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 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:

**Algorithm 1** Sequential Importance Resampling (bootstrap filter)

For time \(t \geq 1\):

1. For \(i = 1, \ldots, N\):
   
   Sample \(X^{(i)}_t \sim \begin{cases} \mu(\cdot|\hat{\theta}_0), & \text{if } t = 1 \\ f(\cdot|X^{(i)}_{t-1}, \hat{\theta}_{t-1}), & \text{if } t \geq 2 \end{cases}\)

2. Compute normalized weights satisfying
   
   \[w_t(X^{(i)}_1: t) \propto w_{t-1}(X^{(i)}_1: t-1) \cdot g(Y_t|X^{(i)}_t, \hat{\theta}_{t-1})\]

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 \(\sum_{i=1}^N w_t(X^{(i)}_t)^{-2} \), \(w_0(\cdot) = 1/N\), and \(X^{(i)}_t\) is shorthand for the \(i\)th coordinate of \(X^{(i)}_1\). 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 \(\theta\) 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). In particular, we choose a lag \(\Delta > 0\) and then at time \(t\), using particles \(X^{(i)}_{1:t}\) shaped by data \(Y_{1:t}\), estimate the \(t - \Delta\)th term of the summation in (2). We will use \(X^{(i)}_{1:t}(t-\Delta)\) to denote the \(t - \Delta\)th coordinate of the particle \(X^{(i)}_{1:t}\), but we will continue to write \(X^{(i)}_t\) as a shorthand for \(X^{(i)}_{1:t}(t)\). (see Table 1 for an overview of notation used in this paper.)

The fixed-lag technique involves making the approximation

\[
S_t \approx \mathbb{E}_{X_{1:t}|Y_{1:t}, \theta} \left[ \frac{1}{t-\Delta} \sum_{j=1}^{t-\Delta} s(Y_j, X_j) \right] \\
\approx \frac{1}{t-\Delta} \sum_{j=1}^{t-\Delta} \mathbb{E}_{X_{1:j+\Delta}|Y_{1:j+\Delta}, \hat{\theta}} \left[ s(Y_j, X_j) \right], \quad (4)
\]
Adaptive learning in sequential Expectation Maximization

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 componentwise sufficient statistics

$$\tilde{s}_t : = \mathbb{E}_{X_1:t|Y_1:t,\theta} [s(Y_{t-\Delta}, X_{1:t}(t - \Delta))]$$

$$\approx \sum_i w_k(X_{1:t}^{(i)}) s(Y_{t-\Delta}, X_{1:t}^{(i)}(t - \Delta)),$$

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 weight $\gamma_t$; in (4) $\gamma_t = 1/(t - \Delta)$. This approach is slightly different from that of (Cappé and Moulines 2005); see Sect. 7.1 for a discussion.

Choosing a large value of $\Delta$ allows SMC to use many observations to improve the posterior distribution of $X_{t-\Delta}$. However the cost of a large $\Delta$ is a loss in particle independence due to the resampling procedure which increases the sample variance. The optimal choice for $\Delta$ 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 \( m \)th batch we have \( t = mb \), at which time
\[
\hat{S}_{t}^{\text{BEM}} := \frac{1}{b} \sum_{k=(m-1)b+1}^{mb} \tilde{s}_k,
\]
is our approximation of \( S \), and \( \hat{\sigma}^2_{v,t} := \hat{S}_{t}^{\text{BEM}} \).

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 \( \{\hat{s}_k\}_{k=\Delta+1,\ldots,t} \), weighted by a pre-specified weighting sequence. The choice of weighting sequence determines how quickly the algorithm “forgets” the earlier parameter estimates. In OEM at time \( t \),
\[
\hat{S}_{t}^{\text{OEM}} = \gamma_t \cdot \tilde{s}_t + (1 - \gamma_t) \cdot \hat{S}_{t-1}^{\text{OEM}},
\]

where \( \{\gamma_k\}_{k=1,2,\ldots} \) is the chosen weighting sequence, typically of the form \( \gamma_t = t \cdot c \) for a chosen \( c \in (0, 1] \) (Cappé 2009). Note that when using lag \( \Delta \), \( \gamma_t = (t - \Delta)^{-c} \) for \( t \geq \Delta \). This update rule ensures that at time \( t \), \( \hat{S}_{t}^{\text{OEM}} \) is a weighted sum of \( \{\hat{s}_k\}_{k=\Delta+1,\ldots,t} \) where the term \( \tilde{s}_k \) has weight
\[
\eta_k := \gamma_k (1 - \gamma_{k+1}) \cdots (1 - \gamma_{t-1})(1 - \gamma_{t}).
\]

#### 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 \( \tilde{s}_t = \sum_{i=1}^{N} w_t(X_{1:t}^{(i)}, Y_t - \Delta - X_{1:t}^{(i)}(t - \Delta))^2 \)
3. Update running average of sufficient statistics \( \hat{S}_{t}^{\text{OEM}} = \gamma_t \tilde{s}_t + (1 - \gamma_t) \hat{S}_{t-1}^{\text{OEM}} \)
4. Maximize expected likelihood by setting \( \hat{\theta}_t := \hat{S}_{t}^{\text{OEM}} \)

Although this method can outperform BEM, its performance remains strongly dependent on the parameter \( c \) determining the weighting sequence, 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 weight sequence 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 $t_0$.

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

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 weighting sequence $\{\gamma_k\}_{k=1}^{\infty}$. The algorithm is similar to OEM, but instead of pre-specifying $\gamma_t$, we estimate the precision and accuracy in the sufficient statistic updates $\{\tilde{s}_k\}_{k=\Delta+1}^{t}$ and use these to determine the next weight $\gamma_{t+1}$. More precisely, we keep online estimates of a weighted regression on the dependent variables $\{\tilde{s}_k\}_{k=\Delta+1}^{t}$, where the index $k$ serves as the explanatory variable and the data point $(k, \tilde{s}_k)$ has weight (6) as before. This weighted regression results in intercept and slope estimates $\hat{\beta}_0, \hat{\beta}_1$, and estimates of their variance $\hat{\sigma}_0^2, \hat{\sigma}_1^2$. We next use these estimates to define a proposed weight as follows:

$$\gamma^{reg}_{t+1} = \frac{\hat{\sigma}_1 + \hat{\beta}_1}{\hat{\sigma}_0},$$

This definition of $\gamma^{reg}_{t+1}$ ensures that a substantial slope estimate $\hat{\beta}_1$ indicating low accuracy in our previous parameter estimates will put a large weight on the incoming statistic, improving accuracy. A large $\hat{\sigma}_0$ reflecting low precision in the estimates will result in a small weight, so that successive estimates are smoothed out, improving precision.

We do not use standard weighted regression, where the weights are assumed to be inversely proportional to the variance of the observation, as this assumption is not justified here; the standard procedure would lead to biased estimates of $\hat{\sigma}_0^2$ and $\hat{\sigma}_1^2$, and would impact the performance of IOEM. Instead we assume that observations share an unknown variance, and we use the weights to modulate the influence of each observation to the estimates of both $\hat{\beta}_{0,1}$ and $\hat{\sigma}_{0,1}^2$. See Sect. 7.2 for details.
We impose restrictions on $\gamma_t$ which keep it between the most extreme choices for OEM. Taken together, the update step for $\gamma$ becomes

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

(7)

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 Cappe 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 Cappe 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 new particles using SMC with parameter $\hat{\theta}_{t-1}$
2. Collect sufficient statistic
   $$\hat{s}_t = \sum_{i=1}^{N} w_t(X^{(i)}_{1:t}) \cdot (Y_{t-\Delta} - X^{(i)}_{1:t}(t - \Delta))^2$$
3. Maximize expected likelihood by setting
   $$\hat{\theta}_t = \hat{S}_{t}^{IOEM} := \gamma_t \cdot \hat{s}_t + (1 - \gamma_t) \cdot \hat{S}_{t-1}^{IOEM}$$
4. Perform weighted regression on $\hat{s}$ to calculate $\gamma_{t+1}$

The results of using BEM, OEM, and IOEM to perform parameter inference on model (1) with a wide range of tuning parameters $b$ from 100 to 10,000, and $c$ from 0.6 to 0.9, are presented in Figure 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 $t_0 = 50,000$.

The adapting weight sequence $\{\gamma_k\}_{k=1,\ldots}$ sets IOEM apart from OEM. This formulation of IOEM only works in the setting where $\theta$ has a linear relationship with a single sufficient statistic (here $\hat{\sigma}_{t}^2 = \hat{S}_t$) and is meant as an introduction to some of the ideas involved in IOEM. The method outlined in Algorithm 3 will not suffice when the function $\Lambda$ mapping the sufficient statistics to $\theta$ does not have this simple form. We introduce the general IOEM algorithm in Sect. 3 below.
3 EM Simulations in the Full Autoregressive Model

The model of Sect. 2 is special in that the sufficient statistic and the parameter of interest coincide. Generally this is not true, 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 $a, \sigma_w$, and $\sigma_v$. We define four sufficient statistics,

$$ S_{1,t} = E_{X_1 \mid Y_1, \theta} \left[ \frac{1}{t-1} \sum_{k=1}^{t-1} X_k^2 \right], $$

$$ S_{2,t} = E_{X_1 \mid Y_1, \theta} \left[ \frac{1}{t-1} \sum_{k=1}^{t-1} X_k \cdot X_{k+1} \right], $$

$$ S_{3,t} = E_{X_1 \mid Y_1, \theta} \left[ \frac{1}{t-2} \sum_{k=2}^{t} X_k^2 \right], $$

$$ S_{4,t} = E_{X_1 \mid Y_1, \theta} \left[ \frac{1}{t} \sum_{k=1}^{t} (Y_k - X_k)^2 \right]. $$

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

$$ \hat{a}_t = \frac{S_{2,t}}{S_{1,t}}, $$

$$ \hat{\sigma}_{w,t} = \left( \frac{S_{3,t}}{S_{2,t}} - \frac{(S_{2,t})^2}{S_{1,t}} \right)^{1/2}, $$

$$ \hat{\sigma}_{v,t} = \left( \frac{S_{4,t}}{S_{1,t}} \right)^{1/2}. $$
where \( \hat{S}_t \) is an approximation of \( S_t \).

In most cases, as above, the function \( \Lambda \) 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 weight sequence \( \{ \gamma_k \}_{k=1,2,...} \). We therefore estimate an adapting weight sequence 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 (5) using our adapting weight sequence instead of a user specified weighting sequence. Because the adapting weight sequence 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 \( \hat{S}_{1,t} \) and \( \hat{S}_{2,t} \) for (8) and by \( \hat{S}_{1,t}^{\sigma w} \) and \( \hat{S}_{2,t}^{\sigma w} \) for (9).

Simply regressing on \( \hat{\theta}_{1,t} \) with respect to \( t \) would correspond to regression on \( \hat{S}_{1,t} \), not \( \hat{s}_{1,t} \). 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 \( \hat{\theta}_t \) (see Algorithm 4). This is accomplished by taking linear combinations,

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

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 \( \hat{s}_t \) used in Sect. 2.3. See Sect. 7.3 for further details on this step.

**Algorithm 4** Introspective Online Expectation Maximization in the general model

For time \( t \geq 1 \):

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

Estimates for the \( a \) parameter under different EM methods are presented in Fig. 2; for the other parameter inferences see Sect. 7.5, Fig. 5. In the AR(1) model, IOEM outperforms most other EM methods when estimating the \( a \) parameter. It is worth noting that in this case, OEM with \( c = 0.6 \) substantially
Adaptive learning in sequential Expectation Maximization

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, $N = 100$

outperforms OEM with $c = 0.9$. 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 $a$, $\sigma_w$, and $\sigma_v$ much more quickly. Here IOEM recognizes that it should have similar behavior to OEM with $c = 0.6$, whereas in the inference displayed in Figure 1 IOEM chose behavior similar to OEM with $c = 0.9$. IOEM can indeed adapt to the model.

4 EM Simulations in a Two-Dimensional AR Model

Now we investigate a model with a larger number of parameters and varying accuracy of initial parameter estimates. IOEM’s main advantage 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

$$
X^A_t = a^AX^A_{t-1} + \sigma^AW^A_t, \quad X^B_t = a^BX^B_{t-1} + \sigma^BW^B_t, \\
Y^A_t = X^A_t + \sigma_vV^A_t, \quad Y^B_t = X^B_t + \sigma_vV^B_t. \tag{11}
$$

Note that $Y^A$ and $Y^B$ are uncoupled, and that their master equation have independent parameters except for a shared parameter $\sigma_v$. 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.
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, $N = 100$

(a^A, \sigma_w^A) and quickly start smoothing out noise. To IOEM, the other parameters appear to not have converged ($\sigma_{w}^B$ and $\sigma_v$ because they are at the wrong value, $a^B$ because it will be changing to compensate for $\sigma_{w}^B$ and $\sigma_v$).

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. 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.

Figure 3 shows the inference of $\sigma_v$, which due to its dependence on components A and B, suffers the most from a blanket choice of tuning parameter in BEM or OEM. The inference of the other parameters and comparisons with a different choice of AVG threshold are shown in Sect. 7.5, figures 6-9.

5 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
\[ f(x_t|x_{t-1}) = (2\pi \sigma^2)^{-1/2} \exp\left\{ -\frac{(x_t - \phi x_{t-1})^2}{2\sigma^2} \right\}, \]
\[ g(y_t|x_t) = (2\pi \beta^2 e^{x_t})^{-1/2} \exp\left\{ -\frac{1}{2\beta^2 e^{x_t}} y_t^2 \right\}. \]

We define four summary sufficient statistics,
\[ S_{1,t} = \mathbb{E}_{X_1:t, Y_1:t, \theta} \left[ \frac{1}{t-1} \sum_{k=1}^{t-1} X_k \cdot X_{k+1} \right], \]
\[ S_{2,t} = \mathbb{E}_{X_1:t, Y_1:t, \theta} \left[ \frac{1}{t-1} \sum_{k=1}^{t-1} X_k^2 \right], \]
\[ S_{3,t} = \mathbb{E}_{X_1:t, Y_1:t, \theta} \left[ \frac{1}{t-1} \sum_{k=2}^{t} X_k^2 \right], \]
\[ S_{4,t} = \mathbb{E}_{X_1:t, Y_1:t, \theta} \left[ \frac{1}{t} \sum_{k=1}^{t} e^{-X_k \cdot Y_k^2} \right]. \]

Then the set of parameters that maximises the likelihood at step \( t \) are
\[ \hat{\phi}_t = \frac{\hat{S}_{1,t}}{\hat{S}_{2,t}}, \]  
\[ \hat{\sigma}_t = \left( \frac{\hat{S}_{3,t} - (\hat{S}_{1,t})^2}{\hat{S}_{2,t}} \right)^{1/2}, \]  
\[ \hat{\beta}_t = (\hat{S}_{4,t})^{1/2}. \]

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. 10).

### 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. 8-9). In addition, BEM/OEM/AVG all make use of a single learning schedule \( \{\gamma(s)\} \), 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.
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.

References

Leonard E Baum. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. Inequalities, 3:1–8, 1972.

Olivier Cappé. Online sequential monte carlo em algorithm. In Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on, pages 37–40. IEEE, 2009.

Olivier Cappé and Eric Moulines. On the use of particle filtering for maximum likelihood parameter estimation. In Signal Processing Conference, 2005 13th European, pages 1–4. IEEE, 2005.

Olivier Cappé and Eric Moulines. On-line expectation–maximization algorithm for latent data models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 71(3):593–613, 2009.

Olivier Cappé, Simon J Godsill, and Eric Moulines. An overview of existing methods and recent advances in sequential monte carlo. Proceedings of the IEEE, 95(5):899–924, 2007.

Saneej B Chitralekha, J Prakash, H Raghavan, RB Gopaluni, and Sirish L Shah. A comparison of simultaneous state and parameter estimation schemes for a continuous fermentor reactor. Journal of Process Control, 20(8):934–943, 2010.

Bernard Delyon, Marc Lavielle, and Eric Moulines. Convergence of a stochastic approximation version of the em algorithm. The Annals of Statistics, 27(1):94–128, 1999.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the royal statistical society. Series B (methodological), pages 1–38, 1977.
7 Supplemental text

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

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

where the sufficient statistic is written explicitly as a sum over the path traced out by the particle $X^{(i)}_{1:t}$. The drawback is that for $u \ll t$ the paths will have collapsed due to resampling, increasing the variance for those contributions to
The solution proposed in Cappé and Moulines (2005) is to use instead the approximation
\[
\hat{S}_t \approx \sum_i \left( \sum_{u=1}^{t-\Delta} w_{u+\Delta}(X_{1:u+\Delta}^{(i)} s_u(X_{1:u+\Delta}^{(i)}(u), Y(u)) \\
+ w_t(X_{1:t}^{(i)}) \sum_{u=t-\Delta+1}^{t} s_u(X_{1:t}^{(i)}(u), Y(u)) \right).
\]

This requires storing the quantities
\[
\{ s_u(X_{1:u+\Delta}^{(i)}(u), Y(u)) \}_{u=t-\Delta,...,t}
\]
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
\[
\hat{S}_t \approx \sum_{u=1}^{t-\Delta} \sum_i w_{u+\Delta}(X_{1:u+\Delta}^{(i)} s_u(X_{1:u+\Delta}^{(i)}(u), Y(u)).
\]

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 \( \Delta \), 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 \( \beta \) be the two coefficients, and \( \epsilon \) the vector of errors, with \( \epsilon_k \sim N(0, \sigma^2) \). Finally let \( w \) be a vector of weights. We estimate \( \beta \) by minimizing
\[
\sigma^2 = (X_w \beta - y_w)^\top (X_w \beta - y_w),
\]
where \( X_w \) and \( y_w \) are defined as
\[
X_w := \begin{bmatrix}
w_1 \cdot w_1 \cdot (-n + 1) \\
\vdots \\
w_n \cdot w_n \cdot 0
\end{bmatrix} \quad \text{and} \quad y_w := \begin{bmatrix}
w_1 \cdot y_1 \\
\vdots \\
w_n \cdot y_n
\end{bmatrix}.
\]
Setting the derivative \( \partial \sigma^2 / \partial \beta = 2(X_w \beta - y_w)^\top X_w \) to zero and solving for \( \beta \) results in the standard estimator for weighted regression
\[
\hat{\beta} = (X_w^\top X_w)^{-1} X_w^\top y_w,
\]
or more explicitly
\[
\hat{\beta}_1 = \frac{\sum w_k^2 x_{2k} y_k}{\left( \sum w_k^2 x_{2k}^2 \right) - \left( \sum w_k^2 x_{2k} \right)^2},
\]
\[
\hat{\beta}_0 = \frac{\sum w_k^2 x_{2k} \sum w_k^2 y_k - \left( \sum w_k^2 x_{2k} y_k \right) \left( \sum w_k^2 x_{2k} \right)}{\left( \sum w_k^2 x_{2k}^2 \right) - \left( \sum w_k^2 x_{2k} \right)^2}.
\]
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:
\[
\text{var} \hat{\beta} = \text{var}(X_w^\top X_w)^{-1} X_w^\top y_w
= \text{var}(X_w^\top X_w)^{-1} X_w^\top \epsilon_w
= E \left[ (X_w^\top X_w)^{-1} X_w^\top \epsilon_w \epsilon_w^\top X_w (X_w^\top X_w)^{-1} \right]
= (X_w^\top X_w)^{-1} X_w^\top \text{diag}(w_k \sigma^2) X_w (X_w^\top X_w)^{-1}.
\]
If \( w_k^2 = 1 \) this simplifies to the usual \( \text{var} \hat{\beta} = \sigma^2 (X^\top X)^{-1} \). Writing out the expression for \( \text{var} \hat{\beta} \) 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 \( \hat{\theta}^{(t-1)} \), 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
\[
\hat{\theta}_t := \frac{1}{\gamma_t} \hat{\theta}_t + \left( \frac{\gamma_t - 1}{\gamma_t} \right) \hat{\theta}_{t-1}.
\]
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 $\{\eta_k^t\}_{k=0,...,t}$ 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 $\eta_k^t = \gamma_t$, $\eta_{t-1}^t = \gamma_{t-1}(1 - \gamma_t)$, and so on. Now,

$$\text{cov}(\tilde{\theta}_i, \tilde{\theta}_j) = \text{cov}\left(\frac{1}{\gamma_t} \hat{\theta}_i + \frac{\gamma_i - 1}{\gamma_t} \hat{\theta}_{i-1} + \frac{\gamma_j - 1}{\gamma_j} \hat{\theta}_{j-1}\right)$$

$$= \frac{1}{\gamma_t \gamma_j} \text{cov}(\hat{\theta}_i, \hat{\theta}_j)$$
$$+ \frac{1}{\gamma_j} \left(1 - \frac{1}{\gamma_i}\right) \text{cov}(\hat{\theta}_{i-1}, \hat{\theta}_j)$$
$$+ \frac{1}{\gamma_i} \left(1 - \frac{1}{\gamma_j}\right) \text{cov}(\hat{\theta}_i, \hat{\theta}_{j-1})$$
$$+ \left(1 - \frac{1}{\gamma_i}\right) \left(1 - \frac{1}{\gamma_j}\right) \text{cov}(\hat{\theta}_{i-1}, \hat{\theta}_{j-1}). \quad (15)$$

Writing $\hat{\theta}_i = f_0 + f_1 \sum_{k=0}^i \eta_k^i \tilde{s}_k$ and recalling that $\text{cov}(\tilde{s}_i, \tilde{s}_j) = \begin{cases} 0, & \text{if } i \neq j \\ \sigma_i^2, & \text{if } i = j, \end{cases}$ it follows that

$$\text{cov}(\hat{\theta}_i, \hat{\theta}_j) = \text{cov}\left(f_1 \sum_{k=0}^i \eta_k^i \tilde{s}_k, f_1 \sum_{k=0}^j \eta_k^j \tilde{s}_k\right)$$
$$= \sum_{k=0}^i f_1^2 \eta_k^i \eta_k^j \sigma_k^2,$$

for $i < j$. Substituting into the four terms of (15) yields

$$\text{cov}(\hat{\theta}_i, \hat{\theta}_j) = \frac{1}{\gamma_i \gamma_j} \sum_{k=0}^i f_1^2 \eta_k^i \eta_k^j \sigma_k^2$$
$$+ \frac{1}{\gamma_j} \left(\frac{\gamma_i - 1}{\gamma_i}\right) \sum_{k=0}^{i-1} f_1^2 \eta_k^{i-1} \eta_k^j \sigma_k^2$$
$$+ \frac{1}{\gamma_i} \left(\frac{\gamma_j - 1}{\gamma_j}\right) \sum_{k=0}^{j-1} f_1^2 \eta_k^i \eta_k^{j-1} \sigma_k^2$$
$$+ \left(\frac{\gamma_i - 1}{\gamma_i}\right) \left(\frac{\gamma_j - 1}{\gamma_j}\right) \sum_{k=0}^{i-1} f_1^2 \eta_k^{i-1} \eta_k^{j-1} \sigma_k^2.$$

If we define

$$a := f_1^2 \eta_0^i \eta_0^{i-1} \sigma_i^2,$$
\[ b := \sum_{k=0}^{i-1} f_k^2 \eta_k^{i-1} \eta_k^{i-1} \sigma_k^2, \]

and note that
\[ \eta_k^j = (1 - \gamma_j) \eta_k^{j-1} \text{ for all } k < j, \]

then
\[
\text{cov}(\tilde{\theta}_i, \tilde{\theta}_j) = \frac{1}{\gamma_i \gamma_j} (1 - \gamma_j)a + \frac{1}{\gamma_i \gamma_j} (1 - \gamma_i)(1 - \gamma_j)b \\
+ \frac{1}{\gamma_j} \left( \frac{\gamma_i - 1}{\gamma_i} \right) (1 - \gamma_j)b \\
+ \frac{1}{\gamma_i} \left( \frac{\gamma_i - 1}{\gamma_j} \right) a + \frac{1}{\gamma_i} \left( \frac{\gamma_i - 1}{\gamma_i} \right) (1 - \gamma_i)b \\
+ \left( \frac{\gamma_i - 1}{\gamma_i} \right) \left( \frac{\gamma_j - 1}{\gamma_j} \right) b \\
= 0.
\]

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

| notation | meaning | associated methods |
|----------|---------|--------------------|
| \( \theta \) | true parameter | all |
| \( \hat{\theta}_t \) | parameter estimate at time \( t \) | all |
| \( \tilde{\theta}_t \) | pseudo-independent parameter update | IOEM |
| \( \bar{s}_t \) | sufficient statistic update at time \( t \) | all |
| \( \hat{S}_t \) | summary sufficient statistic from averaging \( \bar{s} \) | all |
| \( N \) | number of particles | all |
| \( \Delta \) | 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 |

Table 1: Notation used in this paper

### 7.5 Supplementary figures
Fig. 5: 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. 6: Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma^A_w = 1$, $\sigma_v = 5.5$, $a^B = 0.95$, $\sigma^B_w = 1$ and initial parameters $a^A_0 = 0.95$, $\sigma^A_{w,0} = 1$, $\sigma_{v,0} = 3$, $a^B_0 = 0.95$, $\sigma^B_{w,0} = 3$. Parameter estimates at $t = 100,000$ are plotted for 100 replicates, $N = 100$. 
Fig. 7: 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 sequence of weights $\gamma_t$ 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 $t_0 = 50,000$.

Fig. 8: Comparison of EM methods on 2-dimensional autoregressive model with true parameters $a^A = 0.95$, $\sigma^A_{w,0} = 1$, $\sigma_v = 5.5$, $a^B = 0.95$, $\sigma^B_{w,0} = 1$ and initial parameters $a^A_{01} = 0.95$, $\sigma^A_{w,0} = 1$, $\sigma_v = 3$, $a^B_{01} = 0.95$, $\sigma^B_{w,0} = 3$. Parameter estimates at $t = 100,000$ are plotted for 100 replicates, $N = 100$. 
Fig. 9: 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 sequence of weights $\gamma_t$ 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 $t_0 = 10,000$.

Fig. 10: 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, $N = 100$.