Enhanced Expectation-Maximization Algorithm for Pure Variance Structural Time Series Models

Javier López-de-Lacalle

Draft version: November 2014

Abstract

Despite some practical advantages of the EM algorithm, its use in the context of structural time series models has been limited due to the observed slow convergence. We propose an enhancement of the algorithm by incorporating information from derivative terms that are null in the original design. Simulation experiments show a notable improvement in the convergence of the algorithm, while keeping parameter estimates practically identical to those obtained with the original algorithm.

Keywords: EM algorithm, Kalman filter, maximum likelihood, structural time series model, unobserved components.

1. Introduction

Structural time series models consist of building-blocks that capture different patterns underlying the dynamics of the data. Several procedures have been developed to obtain maximum likelihood estimates of the parameters in a structural time series model. The focus of this paper is the Expectation-Maximization (EM) algorithm. We study the convergence of the EM algorithm in pure variance structural time series models. Dempster et al. (1977) presented a formal description of the EM algorithm. After this seminal paper the technique was applied to a large variety of situations. Earlier developments of the EM algorithm in the context we are concerned with here were given in Shumway and Stoffer (1982) and Watson and Engle (1983). Koopman and Shephard (1992) and Koopman (1993) provide further calculus about the score vector and the simulation smoother that are involved in the implementation of the EM algorithm. The algorithm is briefly covered in the textbooks Harvey (1989, §4.2.4), Brockwell and Davis (1996, §8.7) and Durbin and Koopman (2001, §7.3.4).

The EM algorithm is characterized by a number of virtues: the likelihood increases at every iteration of the procedure; it often lead to neat expressions for the updating equation; in certain contexts in time series analysis, it has been found to be robust to poorly chosen starting values of the parameters (Hamilton, 1990); self-consistency, which in addition to other properties implies that the same result is obtained under a variety of changing circumstances (Efron, 1982); it ensures that the result satisfies certain constraints such as non-negative variances. Somewhat surprisingly, Watson and Engle (1983) and Harvey and Peters (1990) found that the convergence of the EM algorithm in the context of structural time series models is slow. In particular, as the algorithm approaches the local optimum, the rate of convergence becomes slower and slower. Shumway and Stoffer (1982) noticed this fact as well, pointing it as a disadvantage with respect to other methods. In consequence, the EM algorithm is not widely used. As witnessed in the special issue of the *Journal of Statistical Software* (Commandeur et al., 2011), the most common approach to fit a structural time series model is the optimization of the likelihood function by means of a quasi-Newton method (Byrd et al., 1995).

The EM algorithm has been found to converge slowly in other contexts as well. Some ideas have been proposed in the literature to alleviate this issue. For example, Harvey and Peters (1990) and Jamshidian and Jennrich (1994) proposed choosing a step size at each iteration of the algorithm by means of a line search procedure. More involved ideas are the Aitken's acceleration method (Laird et al., 1987), the conjugate-gradient approach presented in Jamshidian and Jennrich (1993) and the quasi-Newton acceleration proposed in Lange (1995). These approaches mix different methods and depart from the essence of the EM algorithm and the scope of this work.

In this paper, we propose an enhancement where derivative terms that are zero in the original design of the EM algorithm are evaluated at some iterations of the procedure. We show that the additional information obtained from these derivatives improves the convergence of the algorithm, both at points further and closer to the local optimum.

The remaining of the paper is organized as follows. The basic structural model is introduced in Section 2. The traditional and the modified EM algorithms are described in Section 3. Some computational issues are discussed in Section 4. The results of

simulation experiments are summarized in Section 5. An application to two real time series is shown in Section 6. Section 7 concludes. Mathematical derivations are given in the appendices. The algorithms discussed in the paper were implemented in the R language and environment (R Core Team, 2014). A software package[1] and the scripts that replicate the results shown in this paper are available upon request from the author.

2. The basic structural model

The basic structural model (BSM) is a pure variance structural model commonly used in applications. It is a relatively broad model and practitioners often select restricted versions of the model. This model plays a central role in the approach advocated in Harvey (1989) for time series analysis. A detailed view of the features and theoretical properties of this model can be found, for instance, in Harvey (1989, Chapter 2), Brockwell and Davis (1996, Chapter 8) and Durbin and Koopman (2001, §3.2). The model is defined as follows:

$$\begin{aligned}
\textit{observed series:} \quad & y_t = \mu_t + \gamma_t + \epsilon_t, && \epsilon_t \sim \text{NID}(0,\, \sigma^2_\epsilon);\\
\textit{latent level:} \quad & \mu_t = \mu_{t-1} + \beta_{t-1} + \xi_t, && \xi_t \sim \text{NID}(0,\, \sigma^2_\xi);\\
\textit{latent drift:} \quad & \beta_t = \beta_{t-1} + \zeta_t, && \zeta_t \sim \text{NID}(0,\, \sigma^2_\zeta);\\
\textit{latent seasonal:} \quad & \gamma_t = \textstyle\sum_{j=1}^{s-1} -\gamma_{t-j} + \omega_t, && \omega_t \sim \text{NID}(0,\, \sigma^2_\omega),
\end{aligned}$$

for $t = s, \ldots, n$, where s is the periodicity of the data.

The BSM encompasses models that are common in applications: the local level model, that consists of a random walk with a deterministic drift β_0 plus a noise component ϵ_t; the local trend model, where the drift follows a random walk. Setting $\sigma^2_\omega = 0$ yields a model with deterministic seasonality. Setting also $\gamma_1 = \ldots = \gamma_{s-1} = 0$ removes the seasonal component and gives the local trend model. Adding the restriction $\sigma^2_\zeta = 0$ yields the local level model.

[1] The original implementation employs some tools for parallelization that make the installation of the package not straighforward on some platforms. A version of the package more suitable for public distribution is available on the CRAN repository, http://cran.r-project.org/package=stsm.

The state space form of the BSM is given by the following representation:

$$\begin{array}{lll} y_t = Z\alpha_t + \epsilon_t, & \epsilon_t \sim \text{NID}(0, \sigma_\epsilon^2), & \\ \alpha_t = T\alpha_{t-1} + R\eta_t, & \eta_t \sim \text{NID}(0, Q), & \text{with } Q = \begin{bmatrix} \sigma_\xi^2 & 0 & 0 \\ 0 & \sigma_\zeta^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix}, \\ \alpha_0 \sim \text{N}(a_0, P_0), & & \end{array}$$

for $t = 1, \ldots, n$. For $s = 4$, the matrices of this representation are defined as follows:

$$\begin{aligned} y_t &= \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} \alpha_t + \epsilon_t, \\ \alpha_t \equiv \begin{bmatrix} \mu_t \\ \beta_t \\ \gamma_t \\ \gamma_{t-1} \\ \gamma_{t-2} \end{bmatrix} &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mu_{t-1} \\ \beta_{t-1} \\ \gamma_{t-1} \\ \gamma_{t-2} \\ \gamma_{t-3} \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \xi_t \\ \zeta_t \\ \omega_t \end{bmatrix}. \end{aligned}$$

We assume that a_0 and P_0 are known or chosen beforehand. The EM algorithm provides maximum likelihood estimates of the parameters of the model, $\psi = \{\sigma_\epsilon^2, \sigma_\xi^2, \sigma_\zeta^2, \sigma_\omega^2\}$. It will be convenient to denote this vector as $\{\sigma_\epsilon^2, \sigma_{\eta_j}^2\}$ for $j = 1, 2, 3$. Given a set of values for the parameters of the model, the Kalman filter and smoother are run to extract an estimate of the latent components (level, trend and seasonal).

3. Original and enhanced EM algorithm

3.1. Original EM algorithm

The EM algorithm is an iterative procedure that computes maximum likelihood estimates of the vector of parameters ψ. It consists of two steps: 1) expectation step, where the expectation of the density $p(\alpha, y; \psi)$ is evaluated; 2) maximization step, where the expectation is maximized with respect to the vector of parameters.

The joint log-likelihood function of the observed data and the unobserved state vector is given by:

$$\begin{aligned} \log p(\alpha, y; \psi) &= \text{constant} - \frac{n}{2} \log \sigma_\epsilon^2 - \frac{n-1}{2} \log |Q| - \frac{1}{2\sigma_\epsilon^2} \sum_{t=1}^{n} (y_t - Z\alpha_t)^2 \\ &- \frac{1}{2} \sum_{t=2}^{n} \left((\alpha_t - T\alpha_{t-1})' (RQR')^{-1} (\alpha_y - T\alpha_{t-1}) \right). \end{aligned} \tag{1}$$

Since the above log-likelihood function depends on unobserved variables, it is evaluated with respect to the conditional probability density function of the unobserved states,

given the observations. Then, the expected log-likelihood can be written as follows (Shumway and Stoffer, 1982; Koopman and Shephard, 1992):

$$\begin{aligned} E[\log\, p(\alpha, y; \psi)] &= \text{constant} - \frac{n}{2}\log\sigma_\epsilon^2 - \frac{n-1}{2}\log|Q| - \frac{1}{2\sigma_\epsilon^2}\sum_{t=1}^{n}\left(\hat{\epsilon}_t^2 + \text{Var}(\epsilon_t|y)\right) \\ &- \frac{1}{2}\sum_{t=2}^{n}\text{trace}\left[\left(\hat{\eta}_{t-1}\hat{\eta}_{t-1}' + \text{Var}(\eta_{t-1}|y)\right)Q^{-1}\right]. \end{aligned} \tag{2}$$

The elements involved in this expression are obtained by means of the Kalman filter and smoother and the disturbance smoother, given the parameters obtained at the last iteration of the algorithm. For details see also Durbin and Koopman (2001) and Appendix A in this paper. The first order derivatives of equation (2) with respect to the parameters of the model are given by Shumway and Stoffer (1982), Watson and Engle (1983) and Durbin and Koopman (2001):

$$\frac{\partial E[\log\, p(\alpha, y; \psi)]}{\partial \sigma_\epsilon^2} = -\frac{n}{2\sigma_\epsilon^2} + \frac{1}{2\sigma_\epsilon^4}\sum_{t=1}^{n}\left(\hat{\epsilon}_t^2 + \text{Var}(\epsilon_t|y)\right), \tag{3}$$

$$\frac{\partial E[\log\, p(\alpha, y; \psi)]}{\partial \sigma_\eta^2} = \text{diag}\left[-\frac{n-1}{2}Q^{-1} - \frac{(Q'Q)^{-1}}{2}\sum_{t=2}^{n}\left(\hat{\eta}_{t-1}\hat{\eta}_{t-1}' + \text{Var}(\eta_{t-1}|y)\right)\right] \tag{4}$$

The equation (4) holds when Q is a diagonal matrix, as it is the case in the BSM. This equation returns a vector containing the derivatives with respect to the variance parameters in the state vector, $\sigma_{\eta_j}^2$ for $j = 1, 2, 3$, i.e., $\{\sigma_\xi^2, \sigma_\zeta^2, \sigma_\omega^2\}$. Notice that derivation is required only with respect to σ_ϵ^2 and the remaining parameters in Q, that is, $\hat{\epsilon}$, $\text{Var}(\epsilon_t|y)$, $\hat{\eta}$ and $\text{Var}(\eta_{t-1}|y)$ are considered fixed since the expectation is taken conditional on the parameters from the previous iteration. Equating the derivatives (3)-(4) to zero and solving for the parameters of the model yields the following expressions (Shumway and Stoffer, 1982; Koopman, 1993):

$$\begin{aligned} \sigma_\epsilon^2 &= \frac{1}{n}\sum_{t=1}^{n}\left(\hat{\epsilon}_t^2 + \text{Var}(\epsilon_t|y)\right); & \sigma_\xi^2 &= \frac{1}{n-1}\sum_{t=2}^{n}\left(\hat{\eta}_{1,t-1}^2 + \text{Var}(\eta_{1,t-1}|y)\right); \\ \sigma_\zeta^2 &= \frac{1}{n-1}\sum_{t=2}^{n}\left(\hat{\eta}_{2,t-1}^2 + \text{Var}(\eta_{2,t-1}|y)\right); & \sigma_\omega^2 &= \frac{1}{n-1}\sum_{t=2}^{n}\left(\hat{\eta}_{3,t-1}^2 + \text{Var}(\eta_{3,t-1}|y)\right). \end{aligned} \tag{5}$$

These are the updating equations employed in the EM algorithm as described in the references cited above and summarized in Algorithm 1 below.

3.2. Enhanced EM algorithm

We have seen that the strategy of the EM algorithm leads to neat expressions for the updating equations. Here, we notice that as the elements $\hat{\epsilon}$, $\hat{\eta}$, $\text{Var}(\epsilon|y)$ and $\text{Var}(\eta|y)$

Algorithm 1 EM algorithm for pure variance structural time series models

Choose an arbitrary set of values for the parameters $\psi = \{\sigma^2_\epsilon, \sigma^2_\xi, \sigma^2_\zeta, \sigma^2_\omega\}$.

(1) Run the Kalman filter and smoother and the disturbance smoother.

(2) Update the parameters of the model according to equation (5).

Repeat steps (1) and (2) until a predetermined degree of convergence.

are fixed in the expectation step, the following derivatives are zero in the calculations that lead to equations (3)-(4):

$$\begin{array}{llll} \dfrac{\partial \hat{\epsilon}_t}{\partial \sigma^2_\epsilon} = 0; & \dfrac{\partial \hat{\epsilon}_t}{\partial \sigma^2_\eta} = \vec{0}; & \dfrac{\partial \mathrm{Var}(\epsilon_t|y)}{\partial \sigma^2_\epsilon} = 0; & \dfrac{\partial \mathrm{Var}(\epsilon_t|y)}{\partial \sigma^2_\eta} = \vec{0}; \\ \dfrac{\partial \hat{\eta}_t}{\partial \sigma^2_\epsilon} = \vec{0}; & \dfrac{\partial \hat{\eta}_t}{\partial \sigma^2_\eta} = [0]; & \dfrac{\partial \mathrm{Var}(\eta_t|y)}{\partial \sigma^2_\epsilon} = \vec{0}; & \dfrac{\partial \mathrm{Var}(\eta_t|y)}{\partial \sigma^2_\eta} = [0]. \end{array} \tag{6}$$

Looking at the equations of the Kalman filter and smoother summarized in Appendix A, we can see that $\hat{\epsilon}$, $\hat{\eta}$, $\mathrm{Var}(\epsilon|y)$ and $\mathrm{Var}(\eta|y)$ depend on the parameters of the model $\psi = \{\sigma^2_\epsilon, \sigma^2_\xi, \sigma^2_\zeta, \sigma^2_\omega\}$. We propose departing from the original updating equations at some iterations of the algorithm. In particular, instead of keeping the elements $\hat{\epsilon}$, $\hat{\eta}$, $\mathrm{Var}(\epsilon|y)$ and $\mathrm{Var}(\eta|y)$ fixed to the values of the last iteration, we update them when the gradient is evaluated in the maximization step. It can be checked that the gradient is then given by the following equations:

$$\begin{aligned} \frac{\partial E[\log\, p(\alpha, y; \psi)]}{\partial \sigma^2_\epsilon} &= \frac{1}{2\sigma^4_\epsilon}\sum_{t=1}^{n} \hat{\epsilon}^2_t - \frac{1}{\sigma^2_\epsilon}\sum_{t=1}^{n} \hat{\epsilon}_t \frac{\partial \hat{\epsilon}_t}{\partial \sigma^2_\epsilon} + \frac{1}{2\sigma^4_\epsilon}\sum_{t=1}^{n} \mathrm{Var}(\epsilon_t|y) \\ &- \frac{1}{2\sigma^2_\epsilon}\frac{\partial \mathrm{Var}(\epsilon_t|y)}{\partial \sigma^2_\epsilon} - \sum_{t=2}^{n} \hat{\eta}_t \frac{\partial r_t}{\sigma^2_\epsilon} - \frac{1}{2\sigma^2_\eta}\frac{\partial \mathrm{Var}(\epsilon_t|y)}{\partial \sigma^2_\epsilon}, \end{aligned} \tag{7}$$

$$\begin{aligned} \frac{\partial E[\log\, p(\alpha, y; \psi)]}{\partial \sigma^2_{\eta_j}} &= -\frac{n-1}{2\sigma^4_{\eta_j}} - \frac{1}{\sigma^2_\epsilon}\sum_{t=1}^{n} \hat{\epsilon}_t \frac{\partial \hat{\epsilon}_t}{\partial \sigma^2_{\eta_j}} - \frac{1}{2\sigma^2_\epsilon}\sum_{t=1}^{n} \frac{\partial \hat{\epsilon}_t}{\partial \sigma^2_{\eta_j}} + \frac{1}{2\sigma^4_{\eta_j}}\sum_{t=2}^{n} \hat{\eta}^2_{j,t} \\ &- \frac{1}{\sigma^2_{\eta_j}}\sum_{t=2}^{n} \hat{\eta}_{j,t}\left(r_t + \sigma^2_{\eta_j}\frac{\partial r_t}{\partial \sigma^2_{\eta_j}}\right) + \frac{1}{2\sigma^4_{\eta_j}}\sum_{t=2}^{t} \mathrm{Var}(\eta_{j,t-1}|y) \\ &- \frac{1}{2\sigma^2_{\eta_j}}\frac{\partial \mathrm{Var}(\eta_{j,t-1}|y)}{\partial \sigma^2_{\eta_j}}, \quad \text{for } j = 1, \ldots, 3\,. \end{aligned} \tag{8}$$

The calculus that complements these equations is given in Appendix B. The equations (7)-(8) are more involved than the derivatives obtained in equations (3)-(4). In fact,

it is not possible to obtain closed-form expressions for the roots of these derivatives and they must be obtained by means of numerical methods. We use a root finding algorithm of the class of the so-called bracketing algorithms. Among this class of algorithms, we use Brent's method (Brent, 1973). This method is robust to starting values that are far from the root as well as relatively fast. An interval where the root of the gradient is expected to lie must be specified before running the root finding algorithm. As a general rule, the lower bound of the variance parameters can be set equal to zero while the upper bound can be set equal to the variance of the data. The bracketing algorithm reduces the initial interval until the root is bounded within a tolerance chosen beforehand. The modified maximization step is summarized in Algorithm 2.

Algorithm 2 Unconditional maximization step

For σ^2 in ψ

- $\star$ Solve the equations (3)-(4) for σ^2:
  - start a root finding procedure;
  - within this procedure, update $\hat{\epsilon}$, $\hat{\eta}$, $\text{Var}(\epsilon|y)$ and $\text{Var}(\eta|y)$ for each value of σ^2 tried by the root finding procedure.
- $\star$ Update the parameters:
  - if the root finding procedure succeeded:
    - set σ^2 equal to the root found;
  - else:
    - set σ^2 equal to the corresponding value obtained from equation (5).

We will consider a modified algorithm that applies the step described in Algorithm 2 at all the iterations of the algorithm. We will denote this modified algorithm as EM-mod. We will also consider a version that alternates the original and the modified step as follows: the modified maximization step is run every ten iterations of the algorithm starting from the third one, while the original step is run in the remaining iterations. We will refer to this version as the combined EM algorithm, EM-comb for short.

4. Further computational issues

We use a *naive* version of the Kalman filter, i.e., a direct implementation of the recursions that can be found in many textbooks, for instance Harvey (1989) Chapter 3, Pollock (1999) Chapter 9, Durbin and Koopman (2001) Chapter 4. This approach may potentially cause numerical problems. In particular, the covariance matrix of the state vector at period t, P_t, may lose the properties of symmetry and non-negative definiteness. As a safeguard against potential numerical instabilities, a square root filter can be used to compute the matrix P_t. For a review on this issue see, for instance, Tusell (2011) and references therein. In the experiments carried out in this paper, the direct implementation of the Kalman recursions was not troublesome.

Harvey and Peters (1990) extended the EM algorithm in the context of structural models by means of a line search used to choose at each iteration an optimal step size. They found it helpful to improve the convergence of the EM algorithm. In a different context, Jamshidian and Jennrich (1994) obtained small gains in the rate of convergence of the EM algorithm when the step size was chosen by means of optimization methods. Here, we stick to implement the basis of the EM algorithm in order to avoid that ancillary elements distort the interpretation of the simulation exercises shown below. Nevertheless, it may be worth exploring the effect of a line search procedure on the proposed algorithm.

Since we are working with models defined by time invariant matrices, we may expect that the Kalman filter will converge to a steady state (Harvey, 1989, §3.3). The Kalman smoother and the disturbance smoother may also arrive to a steady state. At each iteration of the filter we check whether the steady state has been reached. If the change in the variance of prediction error, f_t, is lower than the tolerance 0.001 over 5 iterations of the filter, we then consider that the filter has converged. Considerable computational savings are therefore obtained in the remaining iterations of the filter as well as in the Kalman smoother and the disturbance smoother.

It is worth noting that the updating step is independent for each parameter. The root finding procedure that is run to obtain a new value of a parameter, say σ_ξ^2, does not require any output from other root finding procedures that are run to update the remaining parameters. Therefore, assuming we are working in a multiple-core processor –which are nowadays common in standard computers– the updating step of the

algorithm can be computed in parallel. That is, instead of updating each parameter serially, the computations are run concurrently by all the available cores. We implemented and tried this approach in a dual-core processor using the Cilk Plus extension of the C language (Intel Corporation, 2010).

5. Simulation results

We perform simulation experiments that consist of 1,000 time series of length 120. The series are generated from the local level model, local trend model, local level with a seasonal component and from the basic structural model with no disturbance term in the observation equation, $\sigma_\epsilon^2 = 0$. Maximum likelihood parameter estimates are obtained by means of the algorithms introduced in Section 3. Following the common practice, the initial state vector, a_0, is defined as a vector of zeros except for the first element, which is set equal to the first observation of the data. The uncertainty on this initial vector is reflected through its covariance matrix, P_0, which is defined as a diagonal matrix that takes on a large value, 10^6 times the variance of the data. Starting parameter values are set equal to 1. The stopping rule is defined by a tolerance equal to 0.01. The maximum number of EM iterations allowed is 250.

Table 1 reports average parameter estimates obtained for each procedure and model. The true parameter values used to generate the data are given in the first row, labelled DGP. The rows labelled EM-orig , EM-mod and EM-comb report the results for the algorithms described in Section 3, which are respectively: the traditional EM algorithm, the modified EM algorithm (that uses the modified updating equations at every iteration) and the combined algorithm (that interchanges the original and the modified equations every ten iterations). On average, parameter estimates are close to the true values regardless of the version of the EM algorithm. Although not reported, these results were very similar to the local optimum found by maximizing the likelihood function using a quasi-Newton method. When differences were noticeable, the results based on the EM algorithm were often closer to the true vector of parameters, giving evidence of the robustness of this method.

Differences in the estimates between the original and the combined versions of the EM algorithm were not expected. More importantly, Table 1 reveals that even when the updating equations proposed in Algorithm 2 are used in all iterations of the

Table 1: Average parameter estimates

	Local level model			Local trend model		
	σ^2_ϵ	σ^2_ξ		σ^2_ϵ	σ^2_ξ	σ^2_ζ
DGP	1600.00	100.00		100.00	30.00	1.00
EM-orig	1608.15	99.47		100.42	29.79	1.01
EM-mod	1609.07	99.19		99.94	30.85	0.96
EM-comb	1608.96	99.24		100.06	30.59	0.98
	Local level plus seasonal			Basic structural model		
	σ^2_ϵ	σ^2_ξ	σ^2_ω	σ^2_ξ	σ^2_ζ	σ^2_ω
DGP	300.00	10.00	100.00	25.00	5.00	100.00
EM-orig	298.92	10.11	100.65	25.05	4.88	101.09
EM-mod	300.07	9.94	100.43	25.14	4.88	101.12
EM-comb	299.95	9.95	100.48	25.12	4.88	101.12

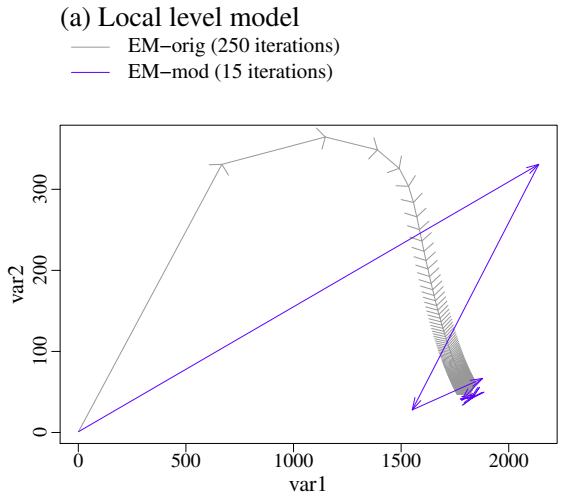
algorithm, practically the same results as the original algorithm is obtained. Therefore, according to these simulations the proposed updating equations do not distort the original rationale of the EM algorithm.

Table 2 reports the number of EM steps required for convergence. Figure 1 illustrates these results displaying the paths followed by the original and the modified algorithms from the starting point to the local optimum. These paths are related to two of the simulated series that were representative of the whole exercise. As others have noticed, we can see that the traditional design of the EM algorithm, EM-orig, converges slowly especially as it approaches the local optimum. The modified procedures converge in far less number of iterations in all models considered in these exercises. For example, EM-mod converged on average in 26 iterations for the local level plus seasonal model. The original procedure required many more iterations, 177 on average, for the same model. A considerable reduction in the number of iterations is also observed for the other models. The number of cases in which the maximum number of iterations is reached is indicated in brackets in Table 2. In these examples, the performance of the modified procedures is notably better than the traditional implementation. For example, in the local level model, EM-orig did not converge after 250 iterations in 101 out of the 1,000 simulated series, while EM-mod failed to converge only in 9 cases.

In most of the applications of the structural models considered here, the time

Figure 1: Sample paths to the local optimum

The figures show sample paths to the local optimum for two simulated series, respectively for the local level and the basic structural model with no disturbance term in the observation equation, $\sigma_\epsilon^2 = 0$. Each coordinate gives the value that each parameter (labelled here as 'var1', 'var2', 'var3') takes on at each step of the algorithm. Each arrow represents an EM iteration. Gray arrows –with narrower tip's angle– depict the path followed traditional EM algorithm. Blue arrows –with broader tip's angle– depict the path followed by the modified algorithm EM-mod.



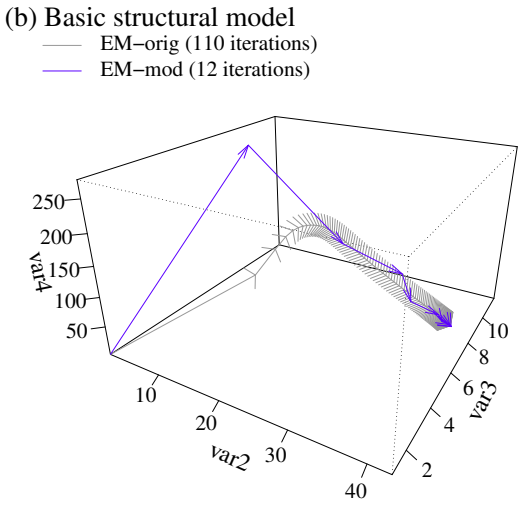


Table 2: Number of iterations until convergence

	Local level model				Local trend model			
	Min.	Median	Mean	Max.(#)	Min.	Median	Mean	Max.(#)
EM-orig	48	134	147	250 (101)	49	240	212	250 (441)
EM-mod	6	12	16	250 (9)	11	34	52	250 (41)
EM-comb	8	28	33	250 (8)	9	58	63	250 (14)
	Local level plus seasonal				Basic structural model			
	Min.	Median	Mean	Max.(#)	Min.	Median	Mean	Max.(#)
EM-orig	79	168	177	250 (195)	11	150	151	250 (56)
EM-mod	10	19	26	250 (19)	6	21	25	250 (2)
EM-comb	38	58	64	250 (18)	9	43	47	250 (1)

Continuation of results in Table 1. Min., minimum; Max., maximum.

required for computations is not usually an issue. For example, running the algorithm for one of the simulated series took on average less than one second. Nevertheless, since the proposed procedure is more cumbersome, for completeness, we show in Table 3 the time employed by each procedure. We include the timings of the implementation mentioned in Section 4, where the updating step of the modified algorithm is computed in parallel. Parameter estimates and convergence of the parallel implementation were the same as those reported for EM-mod in Tables 1 and 2. As expected, the modified version is slower than the traditional implementation. For example, EM-orig employed on average 0.19 seconds to return parameter estimates in the local trend model, while EM-mod employed 0.48 seconds on average. Parallelization reduces the computational time of EM-mod by around a 30%. The parallel implementation of EM-mod was run in a dual-core processor; further gains can be expected in a processor with three cores, that is, with as many cores as parameters to be estimated. The local level model is a special case. For this model, EM-mod is slightly faster than EM-orig and parallelization does not result in a noticeable improvement. The reason is that computations are less demanding in this model since the Kalman filter does not involve operations with matrices but with scalars.

As regards the combined procedure, in these examples it achieved a sensible trade-off between the rate of convergence and the computational time. It is slightly faster

(except in the local level model) although as mentioned before, timing is not an issue in this kind of applications. The point is that the converge in terms of number of iterations is improved with the proposed version of the algorithm. In addition, as reported in Table 2, the number of cases where the combined procedure did not converge after 250 iterations was notably lower than for the original procedure and slightly lower than for EM-mod.

Table 3: Elapsed time until convergence (in seconds)

	Local level model				Local trend model			
	Min.	Median	Mean	Max.	Min.	Median	Mean	Max.
EM-orig	0.03	0.07	0.08	0.15	0.04	0.21	0.19	0.26
EM-mod	< 0.01	0.01	0.01	0.21	0.15	0.43	0.48	1.30
EM-mod parallel	< 0.01	0.01	0.01	0.04	0.10	0.29	0.32	0.91
EM-comb	< 0.01	0.02	0.02	0.15	0.02	0.12	0.13	0.49
	Local level plus seasonal				Basic structural model			
	Min.	Median	Mean	Max.	Min.	Median	Mean	Max.
EM-orig	0.08	0.17	0.18	0.30	0.01	0.15	0.15	0.30
EM-mod	0.18	0.33	0.36	1.63	0.07	0.34	0.35	1.24
EM-mod parallel	0.13	0.23	0.25	1.15	0.07	0.24	0.26	0.93
EM-comb	0.08	0.13	0.14	0.56	0.02	0.10	0.11	0.55

Continuation of results in Table 1. Min., minimum; Max., maximum.

6. Application to real data

In this section, we illustrate the use of the traditional and the modified EM algorithms with two real applications. We fit a structural model to the annual series of flow measurements in river Nile (1871-1970) and to the quarterly series of gas consumption in UK (1960-1986). The latter series is transformed as 100 times the logarithms of the original data. The results are shown in Table 4. As a benchmark, the estimates obtained through the optimization of the likelihood function by means of a quasi-Newton algorithm are also reported. This procedure is labelled as 'qN-optim'.

The local level model is fitted to the Nile time series. The results are almost identical for all the procedures given a tolerance for convergence equal to 0.01. The number of iterations required for convergence are in line with the simulation results

discussed above. The convergence rate is the slowest in the EM-orig procedure (329 iterations); EM-mod converges in 27 iterations and the combined procedure is in-between (88 iterations).

The basic structural model is fitted to the series of gas consumption in UK. The results are similar for all the procedures. The main difference is the allocation of the variance in the trend component. EM-orig returns a variance for the level component larger than the other procedures, $\sigma_\xi^2 = 0.77$. The result obtained with the quasi-Newton optimization method implies a more relevant role of the slope, $\sigma_\zeta^2 = 0.92$, relatively to the level, $\sigma_\xi^2 = 0.00$. In EM-orig and EM-mod the ratio $\sigma_\xi^2/\sigma_\zeta^2$ is larger than unity, instead. In the combined procedure, EM-comb , this ratio is close to unity. Although not displayed, the resulting fitted components (trend and seasonal) were very similar and major differences were not graphically observed. As in the previous application, the number of iterations required for convergence are in line with the behaviour observed in the simulations, except that the combined procedure converges in the same number of iterations as EM-mod.

Table 4: Estimated parameters for Nile and UK gas consumption time series

	Nile annual flow			UK gas quarterly consumption				
	σ_ϵ^2	σ_ξ^2	#	σ_ϵ^2	σ_ξ^2	σ_ζ^2	σ_ω^2	#
qN-optim	15098.58	1469.15	–	19.50	0.00	0.92	37.84	–
EM-orig	15098.21	1469.38	329	16.18	0.77	0.06	34.23	165
EM-mod	15098.53	1469.17	27	17.61	0.23	0.07	33.42	39
EM-comb	15098.33	1469.30	88	18.05	0.06	0.08	33.20	39

'qN-optim' optimizes the likelihood function by means of a quasi-Newton algorithm. # stands for the number of EM steps until convergence (not applicable for 'qN-optim'). In the combined procedure EM-mod is run every ten iterations starting from the third one and EM-orig is run in the other iterations.

7. Conclusion

We have proposed a modification of the EM algorithm in the context of structural time series models where information from derivative terms that are fixed to zero in the original algorithm is included at some iterations of the procedure. We derived full expressions of the derivative terms and modified the algorithm accordingly.

Simulation results validate the proposed modification of the EM algorithm. Firstly, as in the original EM algorithm, parameter estimates close to the true values of the data generating process are obtained. Secondly, a considerable improvement in the rate of convergence is achieved when information from the derivative terms introduced in § 3.2 is incorporated at some or all the iterations.

We have shown that the proposed enhancement to the EM algorithm makes it a compelling procedure that can be used to obtain maximum likelihood estimates in pure variance structural time series models. We conclude that the algorithm is an appealing alternative to the prevalent quasi-Newton optimization method.

The proposed algorithm is computationally more intensive than the original design. Since some of the computations are independent of each other, we implemented a parallel version that reduced the computational time by around a 30%. The simulation experiments suggest that a parallel implementation of the modified algorithm and a procedure combining the traditional and the modified EM algorithm are the most practical approaches in terms of computational time.

Appendix A. Kalman filter and smoother and disturbance smoother

The Kalman filter computes at each iteration the state vector estimator $a_t \equiv E(\alpha_t|y_1,\ldots,y_t)$ and its variance-covariance matrix $P_t \equiv \mathrm{Cov}(\alpha_t|y_1,\ldots,y_t)$, as follows:

$v_t = y_t - Za_t,$	prediction error;	$M_t = P_t Z';$
$f_t = ZM_t + \sigma_\epsilon^2,$	variance of prediction error;	
$K_t = TM_t/f_t,$	Kalman gain;	$L_t = T - K_t Z;$
$a_{t+1} = Ta_t + K_t v_t,$	state vector's prediction;	$P_{t+1} = TP_t L_t' + RQR',$

for $t = 1,\ldots,n$. The filter starts with $a_1 = [y_1, 0\cdots 0]'$ (with s zeros, where s is the periodicity of the data) and P_1 initialized as a $s+1$ diagonal matrix with value $10^6\times$ variance of $\{y_1,\ldots y_n\}$.

The Kalman smoother computes the state estimator given the complete series of observations, $\hat{\alpha}_t = E(\alpha_t|y_1,\ldots,y_n)$ and its variance-covariance matrix $\mathrm{Cov}(\alpha_t|y_1,\ldots,y_n)$. We denote the diagonal of the former matrix as $\mathrm{Var}(\alpha_t|y_1,\ldots,y_n)$. The smoothed estimates of the disturbances and the corresponding variances are computed by the

disturbance smoother recursions:

$$
\begin{aligned}
&r_{t-1} = Zv_t/f_t + L_t' r_t; && N_{t-1} = Z'Z/f_t + L_t' N_t L_t; \quad \hat{\alpha}_t = a_t + P_t r_{t-1};\\
&\hat{\epsilon}_t = \sigma_\epsilon^2 (v_t/f_t - K_t' r_t); && \hat{\eta}_t = QR' r_t;\\
&\mathrm{Var}(\alpha_t|y) = \mathrm{diag}\left(P_t - P_t N_{t-1} P_t\right); && \mathrm{Var}(\epsilon_t|y) = \sigma_\epsilon^2 - \sigma_\epsilon^4 (1/f_t + K_t' N_t K_t);\\
&\mathrm{Var}(\eta_{j,t}|y) = \mathrm{diag}\left(Q - QR' N_t RQ\right),
\end{aligned}
$$

for $t = n, \ldots, 1$, with $r_n = 0$ and $N_n = 0$.

Appendix B. Derivative terms

Similarly to the Kalman recursions, the derivative terms are computed iteratively according to the equations given below. The term $\sigma_\cdot^2$ refers to any of the variance parameters in the model, $\{\sigma_\epsilon^2, \sigma_\xi^2, \sigma_\zeta^2, \sigma_\omega^2\}$. The indicator variable $I(Z)$ returns the indices of the elements in Z that are equal to 1; $\hat{\alpha}_{t,i}$ refers to the i-th element of $\hat{\alpha}_t$, which is of the same order as Z.

It is convenient to write $\hat{\epsilon}_t$ as $\hat{\epsilon}_t = y_t - \hat{\alpha}_t$, then:

$$
\frac{\partial \hat{\epsilon}_t}{\partial \sigma_\cdot^2} = -\sum_{i \in I(Z)} \frac{\partial \hat{\alpha}_{t,i}}{\partial \sigma_\cdot^2}, \qquad \frac{\partial \hat{\alpha}_t}{\partial \sigma_\cdot^2} = \frac{\partial a_t}{\partial \sigma_\cdot^2} + \frac{\partial P_t}{\partial \sigma_\cdot^2} r_{t-1} + P_t \frac{\partial r_{t-1}}{\partial \sigma_\cdot^2}.
$$

The variance parameters related to the disturbances in the state vector of the model, $\{\sigma_\xi^2, \sigma_\zeta^2, \sigma_\omega^2\}$ are denoted as $\sigma_{\eta_j}^2$. The subscript in $\hat{\eta}_{j,t}$ refers to the position in the state vector at period t that is related to the disturbance term whose variance is $\sigma_{\eta_j}^2$; $r_{t,j}$ refers to the element of r_t that is related to the the disturbance η_j.

$$
\frac{\partial \hat{\eta}_{t,j}}{\partial \sigma_\epsilon^2} = \sigma_{\eta_j}^2 \frac{\partial r_{t,j}}{\partial \sigma_\epsilon^2}; \qquad \frac{\partial \hat{\eta}_{t,j}}{\partial \sigma_{\eta_j}^2} = r_{t,j} + \sigma_{\eta_j}^2 \frac{\partial r_{t,j}}{\partial \sigma_{\eta_j}^2}.
$$

$\sum_{i,j \in I(Z)} M$ denotes the sum of the elements in the matrix M crossing the i-th row and the j-th row. The sum is done for i and j taking the values of the indices of those elements in matrix Z that are equal to 1.

$$
\begin{aligned}
\frac{\partial \mathrm{Var}(\epsilon_t|y)}{\partial \sigma_\cdot^2} &= \sum_{i,j \in I(Z)} \frac{\partial \mathrm{Cov}(\alpha_t|y)_{i,j}}{\partial \sigma_\cdot^2}\\
&= \sum_{i,j \in I(Z)} \left(\frac{\partial P_t}{\partial \sigma_\cdot^2} - \frac{\partial P_t}{\partial \sigma_\cdot^2} N_{t-1} P_t - P_t \frac{\partial N_{t-1}}{\partial \sigma_\cdot^2} P_t - P_t N_{t-1} \frac{\partial P_t}{\partial \sigma_\cdot^2} \right)_{i,j}.
\end{aligned}
$$

The term $\mathrm{diag}\,(M)_{\eta_j}$ denotes the element in the diagonal of the matrix M that is related to the component η_j in the state vector. For example, according to the

specification of the basic structural model, $j = 1$ denotes the first component related to disturbance term with variance σ^2_ξ, then, $(N_t)_{\eta_j}$ corresponds to the first element in the diagonal of N_t.

$$\begin{aligned}
\frac{\partial \text{Var}(\eta_{j,t}|y)}{\partial \sigma^2_\epsilon} &= -\left(\sigma^2_{\eta_j}\right)^2 \text{diag}\left(\frac{\partial N_t}{\partial \sigma^2_\epsilon}\right)_{\eta_j}; \\
\frac{\partial \text{Var}(\eta_{j,t}|y)}{\partial \sigma^2_{\eta_j}} &= -\left(\sigma^2_{\eta_j}\right)^2 \text{diag}\left(\frac{\partial N_t}{\partial \sigma^2_{\eta_j}}\right)_{\eta_j} + 1 - 2\sigma^2_{\eta_j} \text{diag}\left(N_t\right)_{\eta_j}.
\end{aligned}$$

The following derivatives are computed during the recursions of the Kalman filter:

$$\begin{aligned}
&\frac{\partial a_{t|t-1}}{\partial \sigma^2_\cdot} = T\frac{\partial a_{t-1}}{\partial \sigma^2_\cdot}. \\
&\frac{\partial a_t}{\partial \sigma^2_\cdot} = \frac{\partial a_{t|t-1}}{\partial \sigma^2_\cdot} + \frac{\partial P_{t|t-1}}{\partial \sigma^2_\cdot} Z\frac{v_t}{f_t} - P_{t|t-1}Z'/f_t^2\frac{\partial f_t}{\partial \sigma^2_\cdot} + P_{t|t-1}Z'/f_t\frac{\partial v_t}{\partial \sigma^2_\cdot}.
\end{aligned}$$

$$\begin{aligned}
&\frac{\partial P_{t|t-1}}{\partial \sigma^2_\cdot} = T\frac{\partial P_{t-1}}{\partial \sigma^2_\cdot}T' + \frac{\partial Q}{\partial \sigma^2_\cdot}. \\
&\frac{\partial P_t}{\partial \sigma^2_\cdot} = \frac{\partial P_{t|t-1}}{\partial \sigma^2_\cdot} - \frac{\partial P_{t|t-1}}{\partial \sigma^2_\cdot} Z'ZP_{t|t-1}/f_t + P_{t|t-1}Z'/f_t\frac{\partial f_t}{\partial \sigma^2_\cdot} Z/f_t P_{t|t-1} - P_{t|t-1}Z'Z/f_t\frac{\partial P_{t|t-1}}{\partial \sigma^2_\cdot}.
\end{aligned}$$

$$\frac{\partial r_t}{\partial \sigma^2_\cdot} = Z\frac{\partial v_t/f_t}{\partial \sigma^2_\cdot} + \frac{\partial L_t}{\partial \sigma^2_\cdot}' r_{t+1} + L_t'\frac{\partial r_{t+1}}{\partial \sigma^2_\cdot},$$

where

$$\begin{aligned}
&\frac{\partial v_t/f_t}{\partial \sigma^2_\cdot} = \left(\frac{\partial v_t}{\partial \sigma^2_\cdot} f_t - v_t\frac{\partial f_t}{\partial \sigma^2_\cdot}\right)/f_t^2; \qquad \frac{\partial v_t}{\partial \sigma^2_\cdot} = -Z\frac{\partial a_t}{\partial \sigma^2_\cdot}; \\
&\frac{\partial f_t}{\partial \sigma^2_\epsilon} = Z\frac{\partial P_t}{\partial \sigma^2_\epsilon}Z' + 1; \qquad \frac{\partial f_t}{\partial \sigma^2_{\eta_j}} = Z\frac{\partial P_t}{\partial \sigma^2_{\eta_j}}Z'
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial N_t}{\partial \sigma^2_\cdot} &= -Z'Z\frac{\partial f_t}{\partial \sigma^2_\cdot}/f_t^2 + \left(\frac{\partial L_t}{\partial \sigma^2_\cdot}\right)' N_{t+1}L_t + L_t'\frac{\partial N_{t+1}}{\partial \sigma^2_\cdot}L_t + L_t'N_{t+1}\frac{\partial L_t}{\partial \sigma^2_\cdot}, \\
\frac{\partial L_t}{\partial \sigma^2_\cdot} &= -\frac{\partial K_t}{\partial \sigma^2_\cdot}Z.
\end{aligned}$$

References

Brent, R. P., 1973. Algorithms for Minimization Without Derivatives. Prentice-Hall, Inc., Englewood Cliffs, New Jersery.

Brockwell, P. J., Davis, R. A., 1996. Introduction to Time Series and Forecasting. Springer Texts in Statistics. Springer-Verlag.

Byrd, R. H., Lu, P., Nocedal, J., Zhu, C., 1995. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing 16 (5), 1190–1208.

Commandeur, J. J. F., Koopman, S. J., Ooms, M., 5 2011. Statistical software for state space methods. Journal of Statistical Software 41 (1), 1–18, URL http://www.jstatsoft.org/v41/i01/.

Dempster, A. P., Laird, N. M., Rubin, D. B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39 (1), 1–38.

Durbin, J., Koopman, S. J., 2001. Time Series Analysis by State Space Methods. Oxford Statistical Science Series. Oxford University Press.

Efron, B., 1982. Maximum likelihood and decision theory. The Annals of Statistics 10 (2), 340–356.

Hamilton, J. D., 1990. Analysis of time series subject to changes in regime. Journal of Econometrics 45 (1–2), 39–70.

Harvey, A. C., 1989. Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press.

Harvey, A. C., Peters, S., 1990. Estimation procedures for structural time series models. Journal of Forecasting 9, 89–108.

Intel Corporation, 2010. Intel®Cilk™Plus, User and Reference Guide.

Jamshidian, M., Jennrich, R. I., 1993. Conjugate gradient acceleration of the EM algorithm. Journal of the American Statistical Association 88 (421), 221–228.

Jamshidian, M., Jennrich, R. I., March 1994. Conjugate gradient methods in confirmatory factor analysis. Computational Statistics & Data Analysis 17 (3), 247–263.

Koopman, S. J., 1993. Disturbance smoother for state space models. Biometrika 80 (1), 117–126.

Koopman, S. J., Shephard, N., 1992. Exact score for time series models in state space form. Biometrika 79 (4), 823–826.

Laird, N., Lange, N., Stram, D., 1987. Maximum likelihood computations with repeated measures: Application of the EM algorithm. Journal of the American Statistical Association 82 (397), 97–105.

Lange, K., 1995. A quasi-Newton acceleration of the EM algorithm. Statistica Sinica 5 (1), 1–18.

Pollock, D. S. G., 1999. A Handbook of Time-Series Analysis Signal Processing and Dynamics. Academic Press.

R Core Team, 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org/.

Shumway, R., Stoffer, D., 1982. An approach to time series smoothing and forecasting using the EM algorithm. Journal of Time Series Analysis 3 (4), 253–264.

Tusell, F., 3 2011. Kalman Filtering in R. Journal of Statistical Software 39 (2), 1–27, URL http://www.jstatsoft.org/v39/i02/.

Watson, M. W., Engle, R. F., 1983. Alternative algorithms for the estimation of dynamic factor, MIMIC and varying coefficient regression models. Journal of Econometrics 23 (3), 385–400.