An inverse transform approach for Gibbs-sampling using data augmentation for latent components in state-space models

March, 2009

Abstract

One of the steps in the Gibbs sampler for the estimation of state-space models requires drawing values for the unobserved state vector. A common approach for this step is the usage of the simulation smoother conditioned to the data. In this document we discuss another approach based on resampling techniques. In particular, we explore Vinod's maximum entropy bootstrap as a technique for drawing the state vector.

1 Introduction

The Gibbs sampler is a Monte Carlo technique that can be used for the estimation of state space models. One of the steps involved in the Gibbs sampler for the estimation of state-space models requires generating draws for the state vector conditioned on the observed data.

The simulation smoother developed in de Jongh and Shephard (1995) and Durbin and Koopman (2002) is a technique that generates draws for the latent state or disturbance in the state equation of a state space model. Thus the usage of the simulation smoother is a common approach for the step mentioned above.

In this document we discuss another approach based on a resampling technique for time dependent data (also conditioned to the observed data). This technique is the Vinod's Maximum Entropy Bootstrap for time series (MEB) Vinod (2004, 2006).

In the remaining of the document we will focus on the simplest state space model: the local level model (LLM). We consider maximum likelihood estimation via de Kalman filter, Gibbs-sampler estimation using the simulation smoother introduced in Durbin and Koopman (2001) and investigate on the performance of MEB for Gibbs-sampling.

The purpose of this document is to give an insight into the maximum entropy bootstrap developed by Vinod (2006) as a technique for drawing the state vector.

2 The local level model

In this document, we elaborate on computer intensive techniques for the estimation of the random walk plus noise model. This model is widely known as the local level model (LLM). We will considerer maximum likelihood (ML) and Gibbs-sampling estimation.

A comprehensive review of state methods for time series analysis can be found in Durbin and Koopman (2001). Chapter 2 in that book is entirely devoted to the local level model. The Gibbs-sampling approach for state-space models is reviewed in the second part of the book Kim and Nelson (1999). We recap the main points of relevance for the purposes of this exposition.

The local level model consists of an unobserved state variable plus an stationary noise:

$$y_t = \alpha_t + \epsilon_t\,, \quad \epsilon_t \sim NID(0, \sigma_\epsilon^2) \tag{1}$$

$$\alpha_{t+1} = \alpha_t + \eta_t\,, \quad \eta_t \sim NID(0, \sigma_\eta^2)\,. \tag{2}$$

With $\alpha_1 \sim N(a_1, P)$ and ϵ_t and η_t are independent of each other. Equation (1) is referred to as the observation or measurement equation whereas the latent component is modelled as a random walk in the state or transition equation in (2).

Taking the stationary representation we can see that the LLM is a restricted ARIMA(0,1,1) model.

2.1 Gibbs-sampling estimation

The Gibbs-sampler is a computer intensive algorithm that approximates marginal or joint distributions by sampling from the conditional distributions.

In the following we sketch the steps in the Gibbs-sampler implemented for the estimation of the local level model.

Let us denote as $\theta = (\sigma_\epsilon^2, \sigma_\eta^2)$ the parameters of the local level model and $\alpha \equiv \{\alpha_t\}_{t=1}^n$ is the unobserved state vector and y is the vector of observed data.

The posterior distribution of the Gaussian local level model is given by:

$$p(\theta|y) \propto p(y|\theta)p(\theta)\,,$$

where $p(y|\theta)$ is functionally equivalent to the likelihood function $L(\theta|y)$ and $p(\theta)$ is the prior distribution of the parameters. This is the well-known statement that the posterior distribution is proportional to the likelihood times the prior distribution.

In order to get draws from the posterior we can design a Gibbs-sampler using data augmentation.[1]

The posterior distribution after data augmentation is given by:

$$p(\theta|y) \propto p(y|\alpha, \theta)p(\alpha|\theta)p(\theta)\,.$$

[1]It could also be possible to use the Metropolis-Hasting algorithm taking as the target as the target distribution the posterior $p(\theta|y) \propto p(y|\theta)p(\theta) \equiv L(\theta|y)p(\theta)$.

The Gibb-sampler allows us to approximate the marginal posteriors ($p(\sigma_\epsilon^2), p(\sigma_\eta^2)$ and $p(\alpha)$) by drawing from the corresponding conditional distributions.

Below we give the conditional distributions involved in the Gibbs-sampler for the Gaussian local level model. The parameters related for the prior distributions are denoted with subindex 0 whereas those related to the posteriors are denoted with subindex 1.

We use conjugate priors for the variances: $\sigma_\epsilon^2 \sim IG(a_0, b_0)$ and $\sigma_\eta^2 \sim IG(c_0, d_0)$ which combined with the Gaussian likelihood yields the conditional posterior distributions shown below.

The results below follow from the relationship that the posterior is proportional to the likelihood times the prior. Our model is Gaussian and we choose an IG prior for the variances. For an exposition of these results within the context of the regression linear model see for instance Judge et al. (1980) and Koop et al. (2007).

Conditional distribution of σ_ϵ^2 given the data y and the state vector α

$$\begin{aligned} p(\sigma_\epsilon^2|y,\alpha) &= IG(a_1, b_1) \quad \text{with} \\ a_1 &= a_0 + \frac{n}{2} \\ b_1 &= b_0 + \frac{\sum_{t=1}^n (y_t - \alpha_t)^2}{2} \end{aligned} \tag{3}$$

Conditional distribution of σ_η^2 given the state vector α

$$\begin{aligned} p(\sigma_\eta^2|\alpha) &= IG(c_1, d_1) \quad \text{with} \\ c_1 &= c_0 + \frac{n-1}{2} \\ d_1 &= d_0 + \frac{\sum_{t=2}^n (\alpha_t - \alpha_{t-1})^2}{2} \end{aligned} \tag{4}$$

Conditional distribution of $\{\alpha_t\}_{t=1}^n$ given the observed data and the two variances θ .

Under Gaussian disturbance term in the state equation the so-called *transition density* is:

$$p(\alpha|\theta) = p(\alpha_1) \prod_{t=2}^n p(\alpha_t|\alpha_{t-1}) \,.$$

With Gaussian disturbance term in the observation equation, the conditional distribution $p(\alpha|y,\theta)$ is also Normal. However the values in the state vector are not independent, moreover, the process is not stationary. We will discuss how to deal with this issue when designing the Gibbs-sampler for the local level model shortly.

Gibbs-sampler for the Gaussian local level model Now we are in a position to design the Gibbs-sampler by iterating the following steps from $i = 1, 2, ..., L+M$. The first L iterations are discarded for subsequent estimation, that is, parameter

estimates will be obtained by taking the mean of the last M draws, respectively for each variance and values in the state vector.

Step 1 Draw the state vector, $\left\{\alpha_t^{(i)}\right\}_{t=1}^n$, conditional on the observed data, $\sigma_\epsilon^{2^{(i-1)}}$ and $\sigma_\eta^{2^{(i-1)}}$.

Step 2 Draw the variance of the disturbance term in the state equation, $\sigma_\eta^{2^{(i)}}$, from the posterior distribution in equation (4) conditional on the state vector $\left\{\alpha_t^{(i)}\right\}_{t=1}^n$.

Step 3 Draw the variance of the disturbance term in the observation equation, $\sigma_\epsilon^{2^{(i)}}$, from the posterior distribution in equation (3) conditional on the state vector $\left\{\alpha_t^{(i)}\right\}_{t=1}^n$ and the observed data.

The Gibbs-sampler has to be initialised for $\sigma_\epsilon^{2^{(0)}}$ and $\sigma_\eta^{2^{(0)}}$. The former initialisation is required by the design of the Gibbs-sampler whereas the latter is required for running the Kalman filter previous to the simulation smoother.

Under Gaussian disturbance terms drawing the variances from the corresponding conditional distributions as required in steps 2 and 3 can be obtained by means of standard random number variates. There are several algorithms for generating random variates: the strategies from the inverse transform and the acceptance/rejection algorithms are often followed. Sometimes, when the posterior distribution does not belong to a a familiar distribution, Metroplis-Hasting within the Gibbs-sampler is used.

Drawing the state vector is an instance of such a peculiar (non-familiar) distribution. Despite the state vector is Gaussian the values are not independent. Moreover, the state equation is often a non-stationary process. For instance, in the local level model the state vector follows a random walk.

A simulation smoother (de Jongh and Shephard, 1995; Durbin and Koopman, 2002) is a common tool used in this situation. In what follows we discuss the usage of Vinod's maximum entropy bootstrap for dependent data for generating draws of the state vector conditional to the observed data.

3 Generating random variates

The development of algorithms for the generation of pseudo-random [2] numbers from the most common distributions has been an important area of research since von Neumann and Metropolis proposed their *midsquare method* for random numbers in 1940.

[2] Any value generated by a computer algorithm is based on a deterministic process and therefore strictly speaking it cannot be considered random. However, good algorithms exist that generate values such that tests (for instance goodness of fit test) cannot reject that the values come from the random distribution at issue. We will use both the names pseudo-random or random number for computer algorithms.

For a review of the most common strategies currently used for random numbers see for instance the relevant chapters in Law and Kelton (2000) and Ross (2006). Here we briefly introduce the inverse transform algorithm previous to the introduction of MEB.

In the Gibbs-sampler described in subsection 2.1 we need to draw from the inverse Gamma distribution and from the particular distribution related to the state vector.

In this section we introduce two algorithms that allow us to generate such draws: the inverse transform algorithm for a Gamma distribution and Vinod's maximum entropy bootstrap for the state vector conditional on the observed data. In a way, the latter may be regarded as a version of the former algorithm allowing for the time dependence exhibited by the data.

3.1 The inverse transform algorithm

We first illustrate the strategy in the discrete setting and then apply the procedure for the case of a continuous exponential distribution.

Let us considered the following discrete random variable X with probability function:

$$Pr(X = x_j) = p_j\,, \quad j = 1, 2, ..., J \quad \sum_{j=1}^{J} = 1\,,$$

The inverse transform algorithm generates a random number U from the uniform [0,1] distribution, $u \in U[0, 1]$, and returns the value X as follows:

$$X = \begin{cases} x_1 & \text{if } U < p_0 \\ x_2 & \text{if } p_0 \leq U < p_0 + p_1 \\ \cdots & \\ x_j & \text{if } \sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i \\ \cdots & \\ x_J & \text{if } \sum_{i=0}^{J-1} p_i \leq U \end{cases} \tag{5}$$

It is illuminating to think of the picture of the cumulative distribution function. The cumulative probability in the vertical axis ranges from 0 to 1. The inverse transform algorithm locates in the vertical axis by generating a random number and then apply the inverse function in order to find value of X corresponding to that probability the abscissa axis.

In short, a random number $U \in U[0, 1]$ is generated and the value $X = F^{-1}(u)$ is returned, where F is the cumulative distribution function.

In the discrete setting this procedure can be implemented by means of a sequence of `if` statements as shown above.

In some instances is not necessary to look for the interval. A relevant case for our puposes is the exponential distribution. The distribution function of an exponential random variable with rate λ is $F(x) = 1 - \exp(-\lambda x)$ with $x > 0$. From $X = F^{-1}(U)$ it follows that $u = F(x) = 1 - \exp(-\lambda x)$. Rearranging terms and taking logarithms it gives $x = -\frac{1}{\lambda}\log(1 - u)$. Notice that $1 - U$ is also uniform in the interval [0,1], hence, it has the same distribution as $-\log(U)$.

Therefore, according to the inverse transform method, we can generate a value X from an exponential distribution with rate λ by generating a random number $U \in U[0,1]$ and setting $X = -\frac{1}{\lambda}\log(U)$.

This procedure can be extended to the Gamma distribution, using the fact the Gamma(a,n) is the sum of n exponentials with rate a. From that point we can go to the inverse Gamma distribution involved in our Gibbs-sampler.

We have seen that the inverse transform method allows us to tackle steps 2 and 3 in the Gibbs-sampler described in subsection 2.1.

3.2 The maximum entropy bootstrap

Running several times the algorithm introduced in the previous subsection generates a vector containing a sequence of independent and identically distributed variates from the distribution at issue.

In the Gibbs-sampler we are dealing with in this document, we need to generate a random vector from the a joint conditional distribution where the values in the vector are not independent. Furthermore, the distribution of the state vector is not stationary.

Here we introduce Vinod's maximum entropy algorithm as a means for dealing with this issue. This description of the MEB procedure is based on the original source Vinod (2004, 2006) and the documentation attached to the `R`-package `meboot` available at CRAN (R Development Core Team, 2006). We also give further insights into the procedure's rationale by giving some correspondence with the inverse method's strategy.

The use of interpolation avoids restriction of the resampling to the set of the observed values. Thus, although a close-form expression based on the MEB strategy cannot be obtained for any vector (as it is the case in the inverse transform method for the exponential distribution) and the iterative process typical of the inverse transform for discrete data, the resultant values come from a continuum of values through the sample path. (See first remark in subsection 2.3 in meboot vignette.)

As a matter of facts Vinod's maximum entropy procedure provides an appealing framework for the purposes of the Gibbs-sampler described in subsection 2.1. In what follows we show some simulation experiments where we can check the potential usefulness of this approach.

3.2.1 Another view to the MEB procedure

We describe and implement an algorithm that overlaps the strategies of the inverse transform method (ITM) and MEB. The algorithm takes the empirical cumulative distribution function (ecdf) and implements the ITM by linear interpolation (as MEB). More importantly, time dependence is preserved by means of the mapping between values and time as proposed in MEB.

As explained in the previous section, the inverse transform algorithm is based on the cumulative distribution function. In some cases, the relationship $X = F^{-1}(U)$

leads to a closed-form solution as we showed in the continuous setting for the exponential distribution.

Some reminiscences to the inverse transform arise when looking at Figure 2 in meboot vignette. MEB is based on the Maximum Entropy distribution. Here we propose using the empirical cumulative distribution function of the data. In practice, an expression for the empirical $F(X)$ for the data is not available and hence we cannot reach to a final solution such as $X = -\frac{1}{\lambda}\log(U)$.

We will consider the discrete setting in IMB as described in equation (5). Given the ecdf, we draw a random number $U \in U[0,1]$ and look for the interval where it lies. We perform linear interpolation in the interval in order to obtain the final draw.

Repeating the previous process as many times as the number of observations in the data gives a new series of values. THe MEB mapping between values and time is applied in order to recover time dependence.

Steps in the procedure:

1. Sort the original data in increasing order and store the ordering index vector.

2. Compute the minimum and maximum values allowed in the replicates (based on trimming means as in MEB).

3. Compute the ecdf and design the corresponding intervals both for the probabilities ranging between 0 and 1 in the vertical axis and the values in the horizontal axis.

4. Draw a random vector U of the same length as the data from the uniform [0,1] distribution.

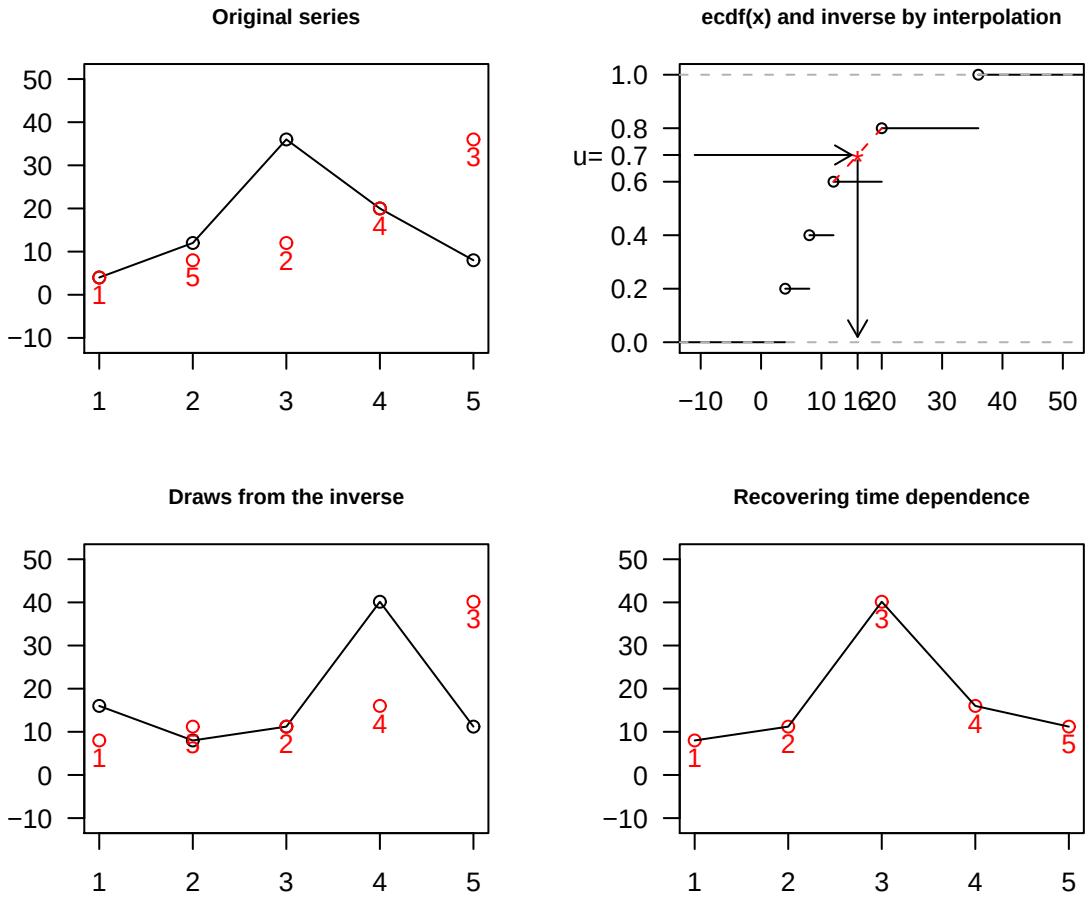
5. Apply the inverse function for the ecdf by linear interpolation at those points in the vector U.

6. Recover time dependence by sorting the series of values obtained in the previous step and applying the index ordering stored in the first step.

The procedure is illustrated in Figure 1 for a 5-points example data. The top-left graphic in this figure displays the original and the sorting index ordering. The top-right graphic shows the ecdf for the data and indicates the inverse transform for a given uniform draw. in the example the uniform draw happens to be 0.7, which gives, by linear interpolation, the value 16 as the final draw. This is repeated 5 times for the following uniform draws $0.70, 0.40, 0.56, 0.93$ and 0.56. The respective values obtained by the inverse method are shown in the bottom-left graphic, where the red points are the draws sorted in increasing order. The numbers indicate sorting index from the first graphic to be applied to the sorted draws in order to recover time dependence. The final vector of draws for the data is shown in the bottom-left graphic.

Remark: Another approach could be to apply the inverse transform in the cumulative distribution based on the maximum entropy distribution computed in MEB.

Figure 1: Inverse method for time dependent data



4 Simulation experiments

4.1 Design

We perform a simulation experiment where a random walk plus noise model is generated according to model (1-2) with $\sigma_\epsilon^2 = 1.5$ and $\sigma_\eta^2 = 0.5$. The simulated data and the underlying true level (in a real situation we couldn't observe directly this component) are displayed in the top-left graphic in Figure 2.

We estimate the local level model by maximum likelihood and Gibbs-sampling. In the latter case we consider the usage of both the simulation smoother exposed in Durbin and Koopman (2001) and Vinod's maximum entropy procedure.

We take the following naming conventions:

- ML_cs: Contemporaneous estimation of the level by maximum likelihood.
- ML_fis: Fixed interval smoothed estimate of the level by maximum likelihood.
- GSS: Smoothed estimate of the level by Gibbs-sampling using the simulation smoother.

- GMEB: Contemporaneous estimate of the level by Gibbs-sampling using MEB.
- GMEB_KS: Smoothed estimate of the level by Gibbs-sampling using MEB.

The estimate ML_fis is obtained by applying the Kalman smoother to the filtered state (contemporaneous estimate) returned by the Kalman filter at the optimum.

The contemporaneous estimate estimates the value at each time t using contemporaneous information, that is, using the observations up to time t. The fixed interval smoothed estimate used all the sample data for each value at time t of the level. Thus, in principle the latter is expected to be more accurate and typically it happens to be smoother.

When running the Gibbs-sampler the simulation smoother generates a smoothed replicate of the state vector upon the filtered state vector (contemporaneous estimation) returned by the Kalman filter upon the corresponding input parameters obtained at each iteration in the Gibbs-sampler. The simulation smoother returns a smoothed replicate for the state vector, that is, the draw for the state vector is not contemporaneous but uses all the information in the sample data. The final estimate of the state vector, GSS, is obtained by taking the mean for each time t of all the replicates generated in the Gibbs-sampler (except for the first L replicates, which are discarded).

In the case of the Gibbs-sampler using MEB, the draw for the state vector is generated by replicating the data y_t and the applying the Kalman filter with the corresponding parameters obtained at each interaction.

Thus we should notice that the implementation of the simulation smoother uses a smoothed replicate of the state vector whereas the implementation of the Gibbs-sampler using MEB uses a contemporaneous replicate of the level.

The final estimate of the state vector GSS is obtained by taking the mean for each time t of all the replicates generated in the Gibbs-sampler (except for the first L replicates, which are discarded). The same is done when using MEB obtaining GMEB.

Finally, GMEB_KS is obtained by applying the Kalman smoother to GMEB for the parameter estimates obtained in the Gibbs-sampler. These parameters[3] are estimated in the Gibbs-sampler as the mean of all the draws obtained in the Gibbs-sampling. In the case of Maximum likelihood these estimates are the output from the optimization routine.

REMARK Remains to include a contemporaneous estimate of the state vector when the SS is used. It can be done by running the Kalman filter for the estimates obtained from GSS. Although not reported here, it can be seen that the first differences of this version does not exhibit correlation.

[3]The variances of the disturbance terms. For the moment we do not discuss the initial value and variance of the of the state at time $t = 1$.

4.2 Parameter estimates

Table 1 reports parameter estimates for the variance of the disturbance term in the observation and state equation (σ^2_ϵ and σ^2_η, respectively) by ML and Gibbs-sampling using the simulation smoother (GSS) and MEB (GMEB).

The ML and GSS estimate for σ^2_ϵ are close to the true value 1.5. Estimate by GMEB is more than two times the standard deviation farther from the true value. This may be due to the fact that the state vector replicates are based on contemporaneous information.

The estimates for the variance in the state vector, σ^2_η, are close to the true value 0.5 in all cases, especially in the Gibbs-sampling approach.

Table 1: Simulation 1. Parameter estimates

	True	ML	GSS	GMEB
σ^2_ϵ	1.5	1.6569	1.4549 (0.2593)	2.6437 (0.3346)
σ^2_η	0.5	0.3227	0.4341 (0.1519)	0.4890 (0.1054)

The Gibbs-sampler is initialised at the true values $\sigma^2_\epsilon = 1.5$ and $\sigma^2_\eta = 0.5$. The prior parameters in the IG distribution are $a = c = 10$ and $b = d = 4$.

4.3 State vector estimates

Figure 2 displays the smoothed estimates for the three procedures. We can see that the ML_fis, GSS and GMEB_KS are close to each other. The autocorrelation functions (ACF) of the first differences of the estimates are plotted in the right-hand side of Figure 2. The first difference of the level in the data generating process is the white noise η and therefore there is no autocorrelation in it. However, we can see that in all the smoothed estimates the ACF suggests the presence of some dependence in the estimates. The ACF truncates at the second or third lag and the first lag is highly significant in the three cases.

Figure 3 displays the contemporaneous estimates ML_cs and GMEB. The smoothed estimated GSS from Figure 2 is also displayed for comparison. The estimates ML_cs and GMEB are close to each other. As expected the filtered estimates are rougher than in the previous graphic. Despite lack of smoothness relatively to the previous case the up and downs are in agreement with the movements in the true level. It is striking noticing that the ACF of the first differences are in this case compatible with a white noise.

Figure 4 displays pairwise comparisons of the state estimates when MEB is used. The series GMEB matches ML_cs and GMEB_KS is close to ML_fis and GSS.

Regarding the existence of autocorrelation in the smoothed estimates when the Gibbs-sampler is used we found that the ACF of the state vector replicates generated at each iteration in the Gibbs-sampler are not significant. Significant lags arise when

the final estimate is taking by averaging for all the simulated values at each period t.

The above-mentioned state vector replicates generated at each iteration in the Gibbs-sampler both for the simulation smoother and MEB are displayed in the animation in Figure 4.

Figure 2: Simulated series and level and level estimates 1/3

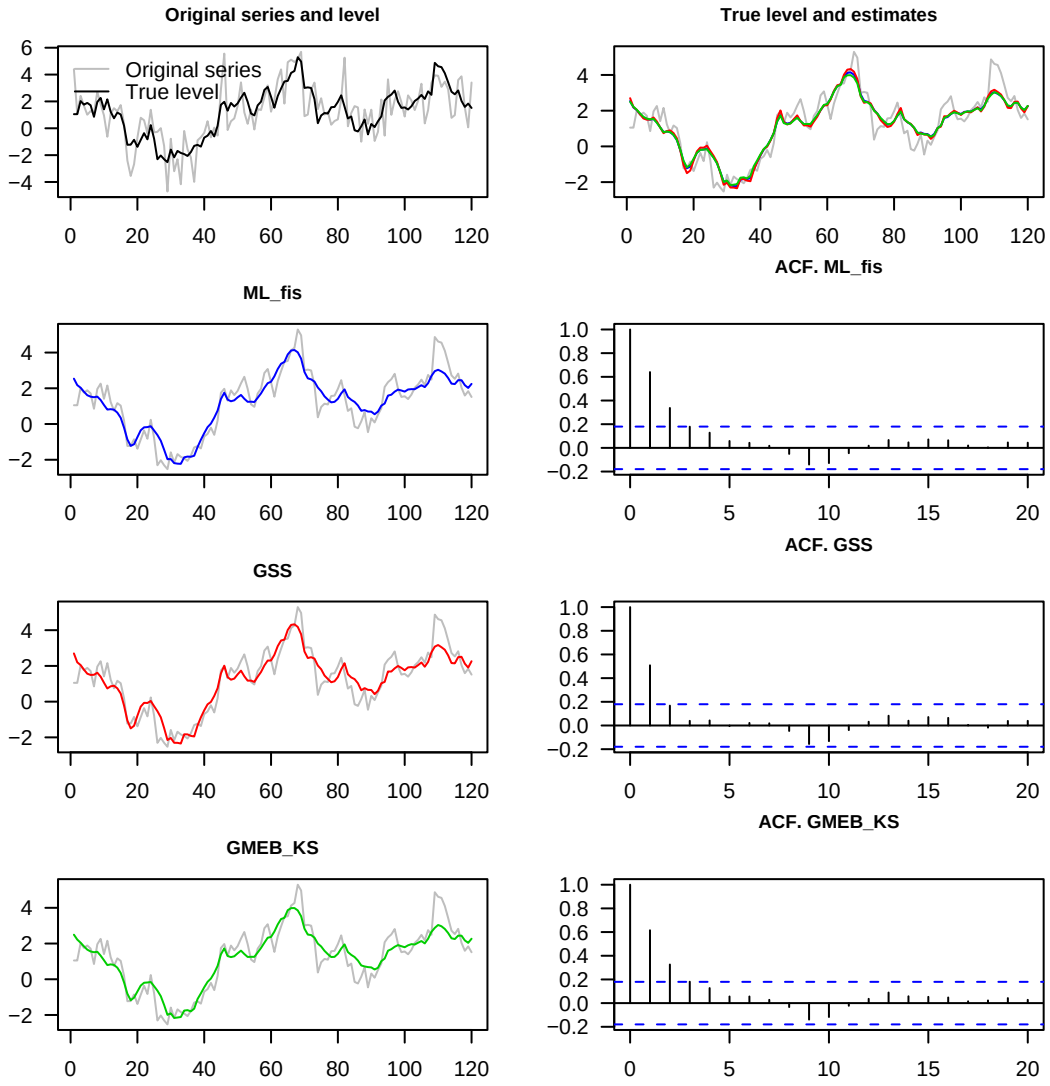


Figure 3: Simulated series and level and level estimates 2/3

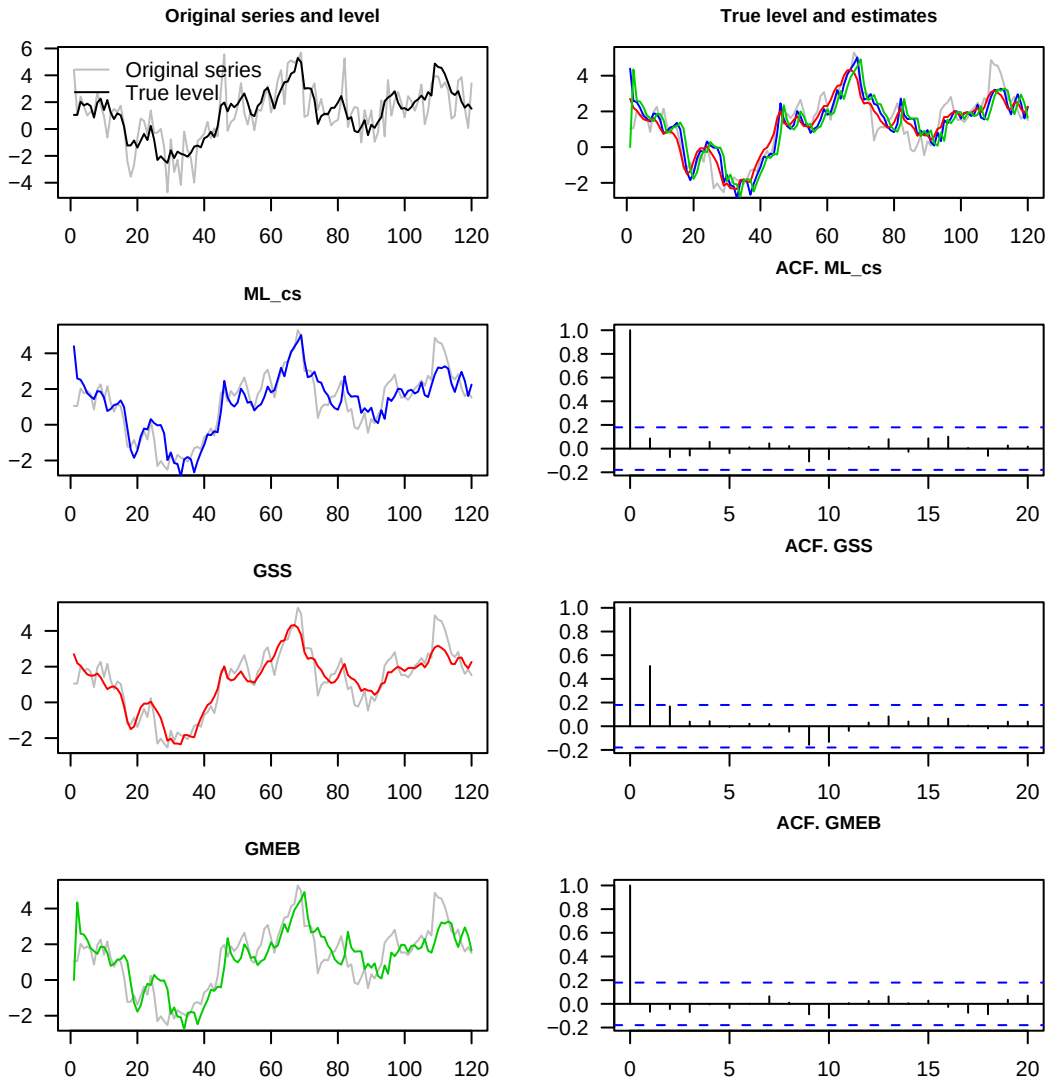
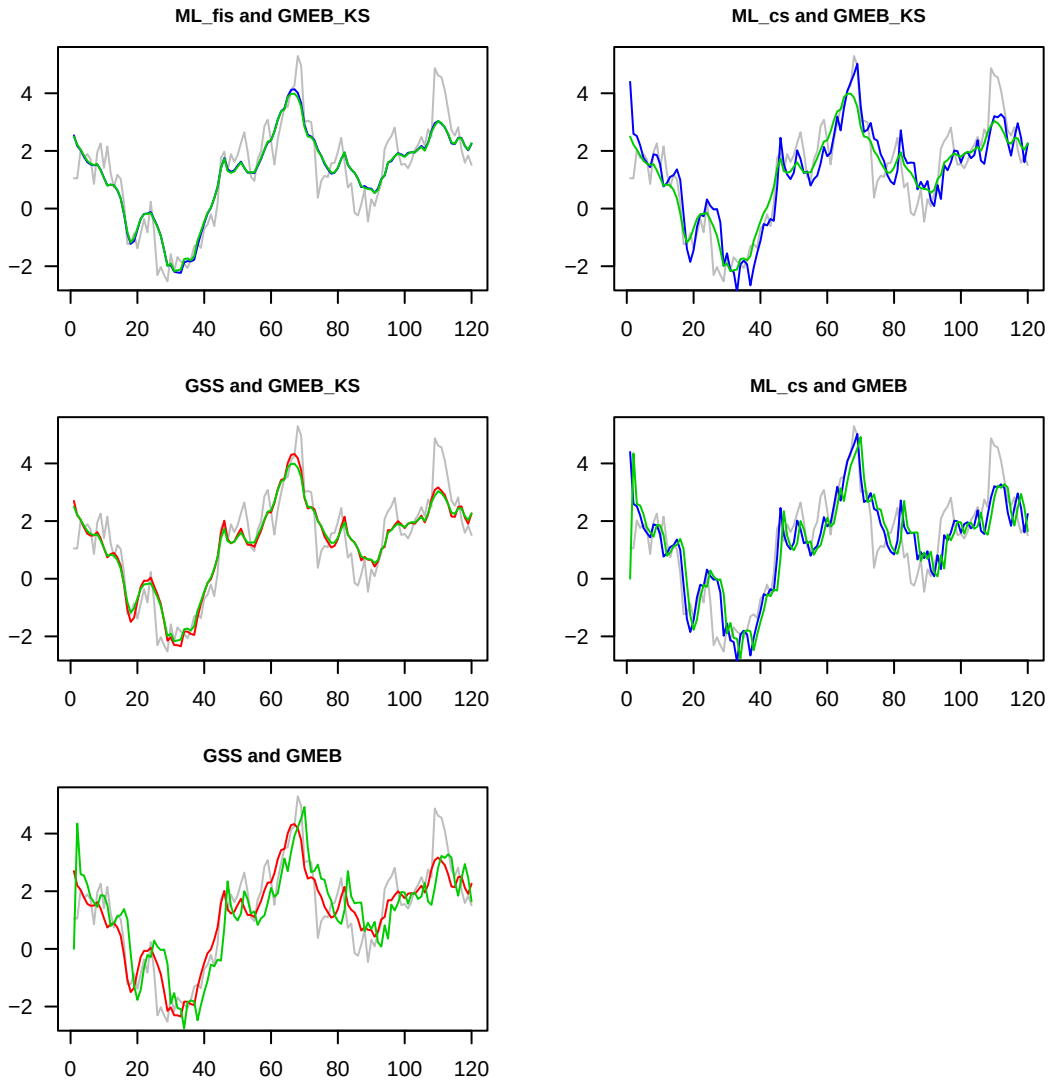


Figure 4: Simulated series and level and level estimates 3/3



5 Conclusions

The purpose of this work was to explore the maximum entropy bootstrap as an alternative to the simulation smoother to obtain parameter estimates of a state space model by means of Gibbs sampling. We also discussed an implemented a version of the maximum entropy bootstrap based on the inverse method for sampling random variates. The main appeal of the maximum entropy approach is the fewer number of operations that are involved in the computations compared to the simulation smoother. We observed the following: 1) The resampling technique for time dependent data proposed by the maximum entropy bootstrap provides replicates of the state vector conditional on the observed data (as required by the Gibbs sampling and imposed in the derivation of the simulation smoother). 2) Gibbs sampling parameter estimates based on the maximum entropy bootstrap are not satisfactory enough. Simulation results reported in this paper and further results not reported showed that parameter estimates are not close to the true values compared to those results obtained by means of the simulation smoother. Further inspection suggested that there is not enough variability on the maximum entropy bootstrap and, hence, it may not be a suitable alternative to the simulation smoother in this context. The animated figure provided with this paper displays resampled state vectors based on each procedure. In this animation the series are overlapped and despite it makes the graphic a little messy, it can be observed that the variance of the state vector based on the maximum entropy technique is much narrower than in the simulation smoother.

Further research is required to adapt the maximum entropy bootstrap to the context of estimation of state space models by Gibbs sampling and exploit its computational advantages compared to the simulation smoother.

References

de Jongh, P. and Shephard, N. (1995), 'The simulation smoother for time series models', *Biometrika* **82**(2), 339–350.

Durbin, J. and Koopman, S. (2001), *Time Series Analysis by State Space Methods*, Oxford University Press.

Durbin, J. and Koopman, S. (2002), 'A simple and efficient simulation smoother for state space time series analysis', *Biometrika* **89**(3), 603–615.

Judge, G., Hill, R., Griffiths and Lee, T.-C. (1980), *The Theory and Practice of Econometrics*, John Wiley & Sons, New York.

Kim, C.-J. and Nelson, C. R. (1999), *State-Space Models with Regime Switching. Classical and Gibbs-Sampling Approaches with Applications*, MIT Press.

Koop, G., Poirier, D. J. and Tobias, J. L. (2007), *Bayesian Econometric Methods*, Cambridge University Press, Cambridge, New York.

Law, A. M. and Kelton, W. D. (2000), *Simulation modelling and analysis*, 3rd edn, McGraw-Hill.

R Development Core Team (2006), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.
URL: *http://www.R-project.org*

Ross, S. M. (2006), *Simulation*, 4th edn, Academic Press.

Vinod, H. (2004), 'Ranking mutual funds using unconventional utility theory and stochastic dominance', *Journal of Empirical Finance* **11**(3), 353–377.

Vinod, H. D. (2006), 'Maximum entropy ensembles for time series inference in economics', *Journal of Asian Economics* **17**(6), 955–978.