Time Series - Final

Tracy Holsclaw

Dec 2009

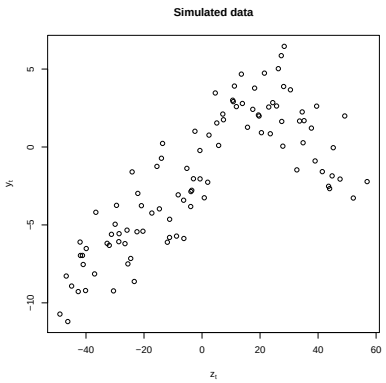


Figure 1: The simulated data set - z_t vs y_t

The data set consists of 100 realizations simulated for y_t and z_t from the following model. This is a first order regression with random walk on both coefficients plus and AR(1) term. The model is given so we will not be discussing model choice but just using the given model.

$$\begin{aligned} y_t &= \beta_{0,t} + \beta_{1,t} z_t + x_t + v_t, & v_t &\sim N(0, \nu) \\ \beta_{0,t} &= \beta_{0,t-1} + w_{0,t}, & w_{0,t} &\sim N(0, u_0) \\ \beta_{1,t} &= \beta_{1,t-1} + w_{1,t}, & w_{1,t} &\sim N(0, u_1) \\ x_t &= \rho x_{t-1} + w_{2,t}, & w_{2,t} &\sim N(0, u_2) \end{aligned}$$

We will set up our DLM in the following manner:

$$\begin{aligned} y_t &= F_t' \theta_t + v_t, & v_t &\sim N(0, V) \\ \theta_t &= G\theta_{t-1} + w_t, & w_t &\sim N(0, W) \end{aligned}$$

Our specific model has these particular values for the DLM are constructed by superposition of the regression model and AR(1) model: $F_t' = [1, z_t, 1]$, $V = \nu$, $\theta_t' = [\beta_{0,t}, \beta_{1,t}, x_t]$, $w_t' = [w_{0,t}, w_{1,t}, w_{2,t}]$,

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix}, W = \begin{bmatrix} u_0 & 0 & 0 \\ 0 & u_1 & 0 \\ 0 & 0 & u_2 \end{bmatrix}$$

1 Known parameters

In this section we will assume that the parameters of the model are known and fixed where $\nu = u_0 = u_1 = u_2 = 1$, and $\rho = 0.9$. The most straight-forward way of solving this problem is with a Kalman

filter. Because all of the parameters are known no MCMC or discount factors are needed. All we will need is a prior for the state space and we will use $N(m_0, C_0)$, where $m_0' = [0, 0, 0]$ and a diagonal matrix $C_0 = 10000 * I$.

Kalman Filter equations:

$$\begin{aligned}
\theta_{t-1}|D_{t-1} &\sim N(m_{t-1}, C_{t-1}) \\
\theta_t|D_{t-1} &\sim N(a_t, R_t) \\
y_t|D_{t-1} &\sim N(f_t, q_t) \\
y_t|D_{t-1} &\sim N(m_t, C_t)
\end{aligned}$$

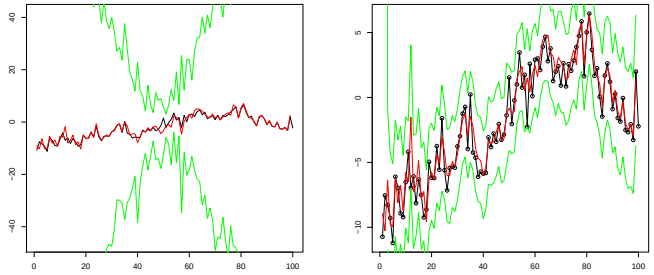
Forward Filter (runs from t=1:T)

$$\begin{aligned}
a_t &= Gm_{t-1} \\
R_t &= GC_{t-1}G' + W \\
f_t &= F_t'a_t \\
q_t &= F_t'R_tF_t + V \\
e_t &= y_t - F_t'a_t \\
m_t &= a_t + A_te_t \\
C_t &= R_t - A_tA_t'q_t \\
A_t &= R_tF_t/q_t
\end{aligned}$$

Backward Smoother: In this notation we set the initial values to $a_t(0) = m_T$ and $R_t(0) = C_T$ and run this from k=(T-1):1

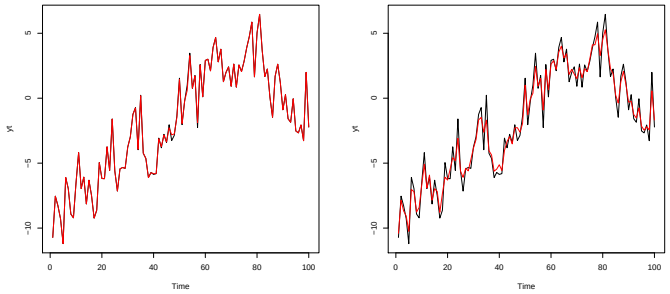
$$\begin{aligned}
\theta_{t-k}|D_t &\sim N(a_t(-k), R_t(-k)) \\
a_t(-k) &= m_{t-k} + B_{t-k}[a_t(-k+1) - a_{t-k+1}] \\
R_t(-k) &= C_{t-k} + B_{t-k}[R_t(-k+1) - R_{t-k+1}]B_{t-k}' \\
B_t &= C_tG'R_{t+1}
\end{aligned}$$

I do not like the way the known parameters make the graph look; the data is in black, mean fit is in red and 95% PIs are in green. The green 95% PI bounds are far too wide and even off the screen and it is probably because the scale between V and W in the known parameters is not great. So, I did the second graph with a adjustments to W to show what I would prefer to have as set parameters. These issue will disappear in 2c) when the parameters are estimated. In the smoothing plots you can see that there is some smoothing happening in the second plot when the parameters of W are adjusted. We see the residual plots have an odd shape because the fit is good on both ends and poor in the middle. But the PI's are poor but the ACF is good; it seems like the model may be fine but the parameters may not be the best. I am hoping that the fit will greatly improve when we allow the parameters to be unknown and estimated.



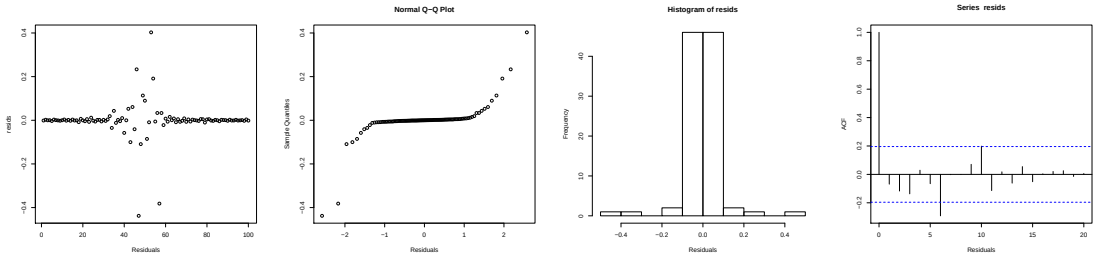
(a) W=1 (b) u_0, u_1=0.001

Figure 2: Filter Fit



(a) W=1 (b) u_0, u_1=0.001

Figure 3: Backward Smoother Fit



(a) Residuals (b) QQ plot of Residuals (c) Histogram of Residuals (d) ACF of Residuals

Figure 4: Smooth Fit Residual Plots

2 Sequential Monte Carlo

Sequential Monte Carlo where all the parameters are known can be done with any of the algorithms in Chapter 6. I will use the SIRS algorithm but could just as easily use a Lui and West, Storvik, or PL algorithm. This is a NDLM will use the optimal importance density which is a Normal distribution with following mean and variance:

$$V(\theta_t|\theta_{t-1}, y_t) = [W^{-1} + F_t V^{-1} F_t']^{-1}$$
$$E(\theta_t|\theta_{t-1}, y_t) = V(\theta_t|\theta_{t-1}, y_t)[W^{-1} G\theta_{t-1} + F_t V^{-1} y_t]$$

The weights associated from this optimal importance density are:

$$w_t \propto w_{t-1} * N(y_t|F_t' G\theta_{t-1}, V + F_t W F_t')$$

I will use a resampling if the particles drop too low so there is not degenerecy in the estimates of $\beta_{0,t}, \beta_{1,t}, x_t$.

The algorithm for SIRS:
This will use the same θ, F_t, G, W, and V as before. The prior for $\pi(\theta) \sim N(m_0, C_0)$ where $m_0' = [0, 0, 0]$ and $C_0 = I * 1e + 7$ and the initial weights will be set to $w_1^m = 1/M$ for m=1:M.

1. We sample θ^m from $MVN(E(\theta_t|\theta_{t-1}, y_t), V(\theta_t|\theta_{t-1}, y_t))$ with the means and variances specified above.
2. Compute the new weights: $w_t \propto w_{t-1} * N(y_t|F_t' G\theta_{t-1}, V + F_t W F_t')$
3. Normalize the weights and calculate $M_{t,eff} = 1/\sum_{m=1}^{M}(w_t^{(m)})^2$. If $M_{t,eff}$ drops below a threshold value then resample $\theta_{1:t}^{(m)}$ with probability $w_t^{(m)}$, reset the wieghts to $w_t^{(m)} = 1/M$.

This results in $(\theta_{1:t}, w_t)^m, m = 1 : M$ and we can plot these results in red and the 95% PI in green; the data is plotted in black. I had 5,000 particles and resampled if the effective size dropped below 2,500. Resampling happened 8 percent of the time. The fit and the probability bands are tighter than the Kalman filter but there is a poor fit in one area. The poor fit makes for odd looking residual plots; this same feature in the fit appeared in the MCMC with these same set known parameters. The model is not fitting the middle region of the data well. Either we need a different model or different parameters in the model.

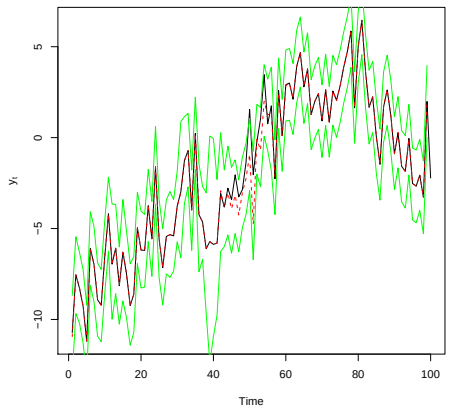


Figure 5: SIRS Algorithm

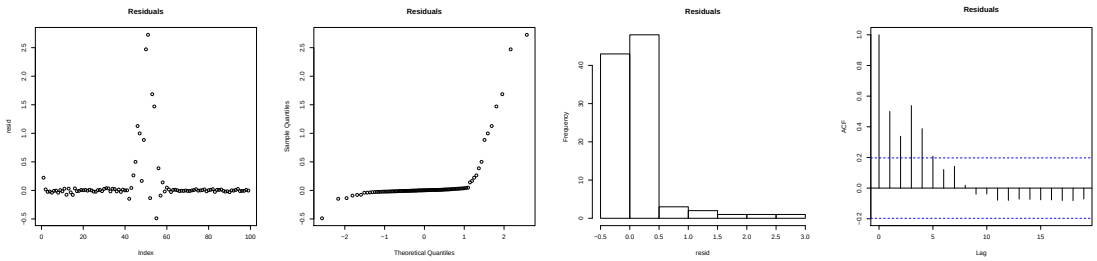


Figure 6: Residuals

3 Unknown parameters

In this section, we will now have ν, u_0,u_1,u_2, and ρ as unknown parameters in our model that need to be updated as well. Because of the nature of our linear and Gaussian DLM all of these parameters can be conjugate priors and sampled through a MCMC algorithm as needed. Parameter learning can also be done through many sequential Monte Carlo algorithms. The SIRS algorithm does not support parameter learning. However, Lui and West (2001) and the particle learning algorithms to name a couple do have schemes for handling unknown parameters and sequential Monte Carlo is possible.

A discount factor or multiple discount factors could be used instead of marginalizing over u_0, u_1, and u_2 in the W matrix. But I am not going to use them because I am already using MCMC and it takes little effort to sample these values with the current algorithm. And discount factors are not part of the sequential Monte Carlo and I want to compare my results between the two methods.

We will replace the backward smoothing step in the Kalman filter process with the backward sampling step. In the MCMC, we need values for θ_t for every iteration to condition on to get draws for the other unknown parameters and the backward sampler is the piece of the algorithm to do this. In this notation we set the initial values to $h_t(0) = m_T$ and $H_t(0) = C_T$

$$\begin{aligned} h_t &= m_t + B_t(\mu_{t+1} - a_{t+1}) \\ H_t &= C_t - B_t R_{t+1} B_t' \\ \theta_t|\theta_{t+1} &\sim N(h_t, H_t) \end{aligned}$$

The unknown parameters will be sampled via Gibbs steps. The unknown parameters need priors: $\pi(\rho) \propto 1$, $\pi(\nu) \sim IG(a_v, B_v)$, $\pi(u_0) \sim IG(a_0, B_0)$, $\pi(u_1) \sim IG(a_1, B_1)$, and $\pi(u_2) \sim IG(a_2, B_2)$.

$$\prod_{t=1}^{T} N(y_t|F_t'\theta - t, V) \prod_{t=2}^{T} N(\theta_t|G\theta_{t-1}, W) MVN\left(\begin{bmatrix} B_{0,0} \\ B_{1,0} \\ x_0 \end{bmatrix} \middle| \begin{bmatrix} m_0 \\ m_1 \\ m_x \end{bmatrix}, \begin{bmatrix} C_0, 0, 0 \\ 0, C_1, 0 \\ 0, 0, C_x \end{bmatrix}\right) IG(\nu|a_v, B_v) *$$

$$* IG(u_0|a_0, B_0) IG(u_1|a_1, B_1) IG(u_2|a_2, B_2)$$

$$\rho|... \sim N\left(\frac{\sum_{t=2}^{T} x_{t-1}x_t}{\sum_{t=2}^{T} x_{t-1}^2}, \frac{u_2}{\sum_{t=2}^{T} x_{t-1}^2}\right)$$

$$v|... \sim IG\left(a_v + \frac{T}{2}, B_v + \frac{1}{2}\sum_{t=1}^{T}(y_t - \beta_{0,t} - \beta_{1,t}z_t - x_t)^2\right)$$

$$u_0|... \sim IG\left(a_0 + \frac{T-1}{2}, B_0 + \frac{1}{2}\sum_{t=1}^{T}(\beta_{0,t} - \beta_{0,t-1}z_t - x_t)^2\right)$$

$$u_1|... \sim IG\left(a_1 + \frac{T-1}{2}, B_1 + \frac{1}{2}\sum_{t=1}^{T}(\beta_{1,t} - \beta_{1,t-1}z_t - x_t)^2\right)$$

$$u_2|... \sim IG\left(a_2 + \frac{T-1}{2}, B_2 + \frac{1}{2}\sum_{t=1}^{T}(x_t - \rho x_{t-1})^2\right)$$

Overall, the algorithm for this set up is as follows for q=1:Q MCMC steps.
1. Use the Kalman filter and then backward sampler to get draws for θ_t conditional on ν, u_0,u_1,u_2, and ρ.
2. Use the Gibbs steps to get draws for ν,u_0,u_1,u_2, and ρ conditional on θ_t At the end we will have Q draws for all of the parameters ν, u_0,u_1,u_2,ρ and every θ_t.

The plots below show the analysis. The first plot includes the time series in black, mean fit in red and 95% PIs in green. The residual plots show that the model does fit better with the unknown parameters but is still having trouble fitting some of the middle time steps. One could try a different model to see if this was improved. Maybe a model with a mixture of Normals for the states to allow for possible outliers in this region. It may just be that that is a short simulated data set and it is this particular realization that the model is having trouble with. It may also be a good idea to put a Normal prior on ρ to tighten it to the interval [-1,1] which is desireable for an AR(1) coefficient or even use a Uniform prior and a Metropolis-Hastings step to sample it. Overall, the fit is pretty good and the 95% PI bands are much better with the parameter uncertainty included in the model. The MCMC with unknown parameters seems to have the best fit but the sequential Monte Carlo (SMC) with known parameters had a better fit than the Kalman filter with known parameters. Parameter learning with SMC maybe difficult because there are 5 parameters to learn and particle degenerecy will be an issue for sure. I went ahead and did forecasting out 20 places for this model and not the others because it fit the best. I had to first pick likely places of z for the next 20 time steps. The forecasting had wide bands and was not worth plotting. The forecasting is direclty related to the value of W. This analysis may be better if I were using discount factors instead. Only when W was set very small did the forecast looks reasonable.

Concluding remarks: The Kalman filter in 2a) works well if the parameters ν, u_0,u_1,u_2, and ρ do not need to be estimated; it is fast and works well. Adjustments can be made with an unknown W by using a discount factor or an unknown V and adjusting the Kalman filter equations to still use this method. But we also have an unknown ρ term. The MCMC method in 2c) handles many unknown parameters but is computationally intensive and takes a long amount of time. Sequential Monte Carlo is an alternative that is used to cut down on the time to analyze data as it comes in. It is just as time intensive as MCMC when running it over the history of a data set but it can take in new values and analyze them on line in a fast and efficient manner. The issues with sequential Monte Carlo are with the need to store a large amount of particles which is requires large amounts of computer memory and some of the algorithms like SIRS do not allow for unknown parameters. The Lui and West algorithm based on kernel densities, the Storvik algorithm based on sufficient statitics, or particle learning algorithm also based on sufficient statitics all can handle parameter learning but particle degeneracy after a hundred or few hundred time steps (depending on the amount of unknown parameters) is an issue and resampling is often required. Resampling may require that MCMC be run again up to that time point, so some of sequential Monte Carlo can be run

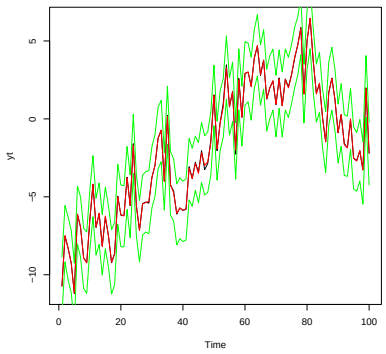


Figure 7: Model Fit

for a time online but then must be updated. Also, the methods based on sufficient statistics require that one can calculate the sufficient statitics for the problem at hand. Other things to consider is the model linearity and Gaussianity, here we assume we have a linear Gaussian DLM but some of the algorithms can be use for more complex models. Overall, there are many algorithms to choose from but they need to be picked carefully depending on the problem at hand.

4 Appendix

I was not happy with the fit in part one so I am adding a discount factor to the model. I am hoping this also will lead to prediction plots that make sense. The discount factor that maximizes the log-likehood is at $\delta = 0.61$. The fits are so much better than the plots in part one. I think the known W used was just too large in part one. The forecast intervals are still large but I have plotted the mean forecast for the next 20 possible iterations (z must be guessed and I used z_{new}=57:77 with steps of one because the average space between z values is 1.07). In general, the discount factor really improves the part one results.

I did not include the residual plots but they were Normal looking in the histogram and qq-plot and did not have the odd variance issue of any of the other residual plots. I think the discount factor works better than even the model with 5 unknown parameters. The third model probably would have been better with a discount factor or multiple discount factors than trying to marginalize over u_0, u_1, and u_2 in W.

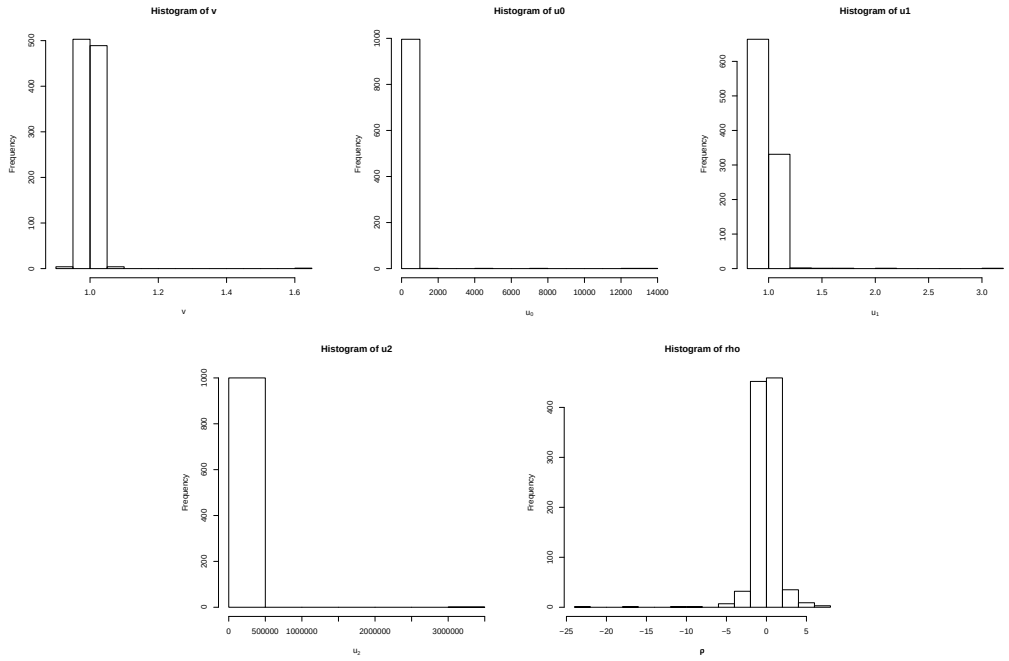


Figure 8: Posteriors

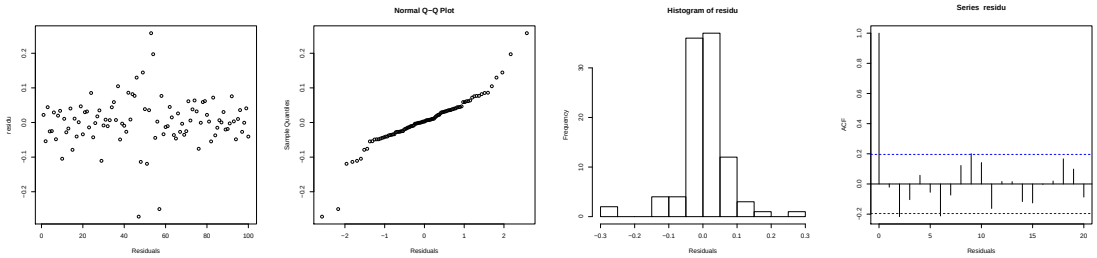
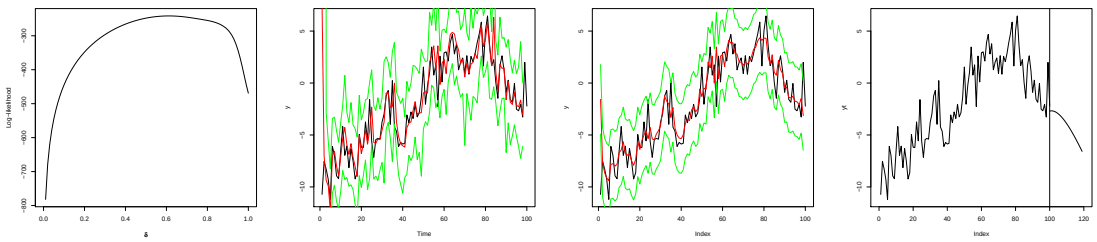


Figure 9: Residuals



(a) Log-likelihood (b) Filtering (c) Smoothing (d) Mean Forecast

Figure 10: Plots using a discount