Bayesian Analysis of Dynamic Linear Models in R

Giovanni Petris

GPetris@uark.edu

Department of Mathematical Sciences

University of Arkansas

Plan

- Dynamic Linear Models
- The R package dlm
- Examples & applications

Dynamic Linear Models

Definition and notations

$$\begin{cases} y_t = F_t\theta_t + v_t & v_t \sim \mathcal{N}(0, V_t) \\ \theta_t = G_t\theta_{t-1} + w_t & w_t \sim \mathcal{N}(0, W_t) \end{cases}$$

for $t = 1, \ldots, n$
Prior distribution for the initial state

$$\theta_0 \sim \mathcal{N}(m_0, C_0)$$

$(\theta_t)_{t\geq 0}$ sequence of unobservable "state vectors"

$(y_t)_{t\geq 1}$ sequence of (vector-valued) observations
$(v_t)_{t\geq 1}$ and $(w_t)_{t\geq 1}$ independent sequences (within and between).
$\mathcal{Y}_t$ observations up to time t, with $\mathcal{Y}_0 = \emptyset$

Harvey (1989), Durbin and Koopman (2001), West and Harrison (1997), ...

Dynamic Linear Models – Examples

- Linear growth model

$$F_t = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad G_t = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

- Quarterly seasonal factors

$$F_t = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad G_t = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Dynamic Linear Models – Model composition

Linear growth plus seasonal component

$$F_t = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad G_t = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

General model composition of n DLM's

$$F_t = \begin{bmatrix} F_t^{(1)} & F_t^{(2)} & \dots & F_t^{(n)} \end{bmatrix} \qquad G_t = \mathrm{BlockDiag}(G_t^{(1)}, G_t^{(2)}, \dots, G_t^{(n)})$$

$$V_t = \sum_{i=1}^{n} V_t^{(i)} \qquad W_t = \mathrm{BlockDiag}(W_t^{(1)}, W_t^{(2)}, \dots, W_t^{(n)})$$

Distributions of interest

$\theta_t \vert \mathcal{Y}_t$		filtering
$\theta_s \vert \mathcal{Y}_t$	$s < t$	smoothing
$\theta_s \vert \mathcal{Y}_t$	$s > t$	forecasting
$y_s \vert \mathcal{Y}_t$	$s > t$	forecasting

Possibly jointly, e.g. $\theta_1, \dots, \theta_t | \mathcal{Y}_t$

Every conditional distribution is Gaussian, identified by mean and variance

Kalman filter

Unknown parameters

May be present in the matrices defining the DLM – evolution and observation equations or variance matrices

Estimation: MLE, Bayes, other

Likelihood evaluation may by tricky – computational stability

The R package dlm

The intended user

- is familiar with R, at least at a basic level
- has some knowledge of Bayesian statistics, including the ideas of Gibbs sampling and Markov chain Monte Carlo (no need to be an expert!)

The generality dilemma

Flexibility vs robustness and ease-of-use
Package dlm is *flexible*

Computational stability issues are dealt with using filtering and smoothing algorithms based on Singular Value Decomposition

Objects of class “dlm”

Constant models are defined in R as lists with components FF, V, GG, W, with a class attribute equal to “dlm”

Creators for common DLM’s are available

```
> mod <- dlmModPoly(2)
> names(mod)
 [1] "m0"  "C0"  "FF"  "V"   "GG"  "W"   "JFF" "JV"  "JGG" "JW"
> mod$FF
     [,1] [,2]
[1,]    1    0
> mod$GG
     [,1] [,2]
[1,]    1    1
[2,]    0    1
> class(mod)
[1] "dlm"
```


Objects of class “dlm” – model composition

dlm objects can be added together

```
> mod <- dlmModPoly(2) + dlmModSeas(4)
> mod$GG
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    1    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0   -1   -1   -1
[4,]    0    0    1    0    0
[5,]    0    0    0    1    0
```


Filtering & Smoothing

Recursive algorithms for filtering and smoothing are based on the SVD of the relevant covariance matrices
Zhang and Li (1996)

SVD of matrix H: $H = USV'$ with U, V orthogonal, S diagonal
For a nonnegative definite symmetric matrix, $U = V$, $S = D^2$

$$\theta_{t-1}|\mathcal{Y}_{t-1} \sim \mathcal{N}(m_{t-1}, C_{t-1}), \qquad C_{t-1} = U_{t-1}D_{t-1}^2U_{t-1}'$$

$$\theta_t|\mathcal{Y}_{t-1} \sim \mathcal{N}(a_t, R_t), \qquad R_t = G_tC_{t-1}G_t' + W_t = \tilde{U}_t\tilde{D}_t^2\tilde{U}_t'$$

$$U_{t-1}, D_{t-1} \longmapsto \tilde{U}_t, \tilde{D}_t$$

Maximum Likelihood

$$\psi \quad \Longrightarrow \quad \text{Model} \quad \Longrightarrow \quad \text{Loglikelihood}$$

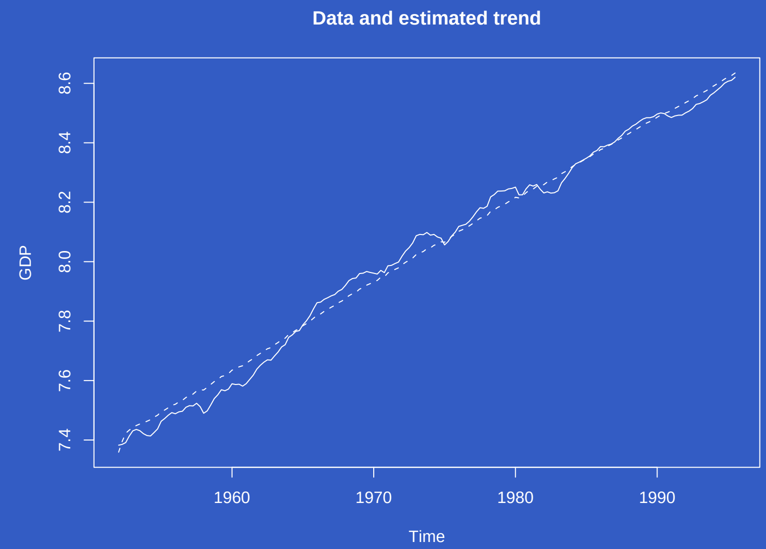
To achieve a maximum of flexibility, the user has to explicitely specify the first step, $\psi \quad \Longrightarrow \quad \text{Model}$

R takes care of the evaluation of the Loglikelihood and of its maximization – via a call to optim

Warning: likelihood maximization is a tricky business!!!

MLE – Example

Data: US quarterly log GDP from 1953 to 1995
Model: linear growth plus AR(2)



MLE – Example

```
> buildGdp <- function(parm) {
+   trend <- dlmModPoly(2, dV=1e-10, dW=exp(parm[1:2]))
+   z <- parm[3:4] / (1 + abs(parm[3:4]))
+   ar2 <- dlmModARMA(ar=c(sum(z),-prod(z)), sigma2=exp(parm[5]))
+   return( trend + ar2 )
+ }
> mleGdp1 <- dlmMLE( gdp, parm=rep(0,5), build=buildGdp)
> set.seed(4521)
> mleGdp2 <- dlmMLE( gdp, parm=rep(0,5), build=buildGdp, method="SANN",
+                                 control=list(temp=20, tmax=25, maxit=20000))
> modFit1 <- buildGdp(mleGdp1$par)
> dlmLL(gdp, modFit1)
[1] 124.8836
> modFit2 <- buildGdp(mleGdp2$par)
> dlmLL(gdp, modFit2)
[1] -693.0615
```


Filtering & Smoothing – Example

```
> filt <- dlmFilter(gdp,modFit2)
> smooth <- dlmSmooth(filt)
> plot(cbind(gdp,smooth$s[,1]), plot.type='s', lty=1:2,
+      ylab="GDP", main="Data and estimated trend")
```

Bayesian analysis

$\Theta_t = (\theta_0, \ldots, \theta_t)$ state vectors up to time t
$\alpha = (\alpha_1, \ldots, \alpha_r)$ vector of unknown parameters

Target posterior distribution $p(\Theta_n, \alpha | \mathcal{Y}_n)$

Obtain a sample from the target distribution using the Gibbs sampler

1. $p(\Theta_n | \alpha, \mathcal{Y}_n)$
2. $p(\alpha | \Theta_n, \mathcal{Y}_n)$

Step 2 may be broken down into several sub-steps involving full conditional distributions of components of α

How can i do it?

For $p(\Theta_n | \alpha, \mathcal{Y}_n)$ the package provides the function dlmBSample, implementing the Forward Filtering Backward Sampling algorithm
Carter and Kohn (1994), Frühwirth-Schnatter (1994), Shephard (1994)

Generating from $p(\alpha | \Theta_n, \mathcal{Y}_n)$ is model-dependent – the generality dilemma strikes again!

R provides functions to generate from standard univariate distributions, dlm provides in addition a function to generate from a Wishart distribution

If the full conditional distribution of α is nonstandard...

If all else fail... arms!

Adaptive Rejection Metropolis Sampling is a black-box algorithm to generate from a univariate continuous distribution on a bounded support
Gilks, Best and Tan (1995)

The package includes a multivariate extension of ARMS
The user needs to write two functions in R:

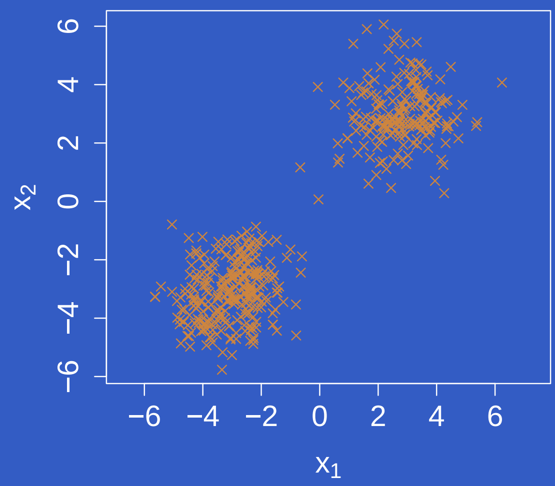
- one to evaluate the logdensity of the target
- the other to evaluate the indicator function of the support

The rest is taken care of by the function arms

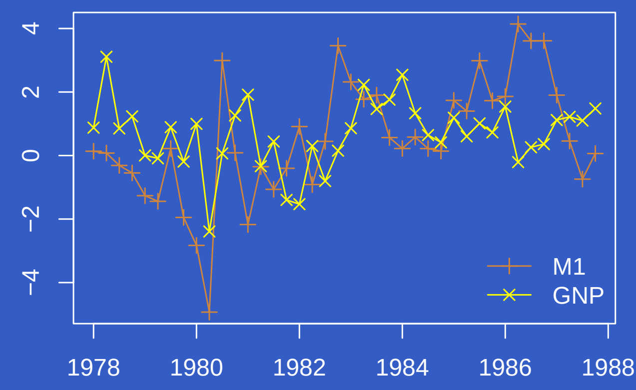
Example

```
> bimodal <- function(x) {
+     log(prod(dnorm(x,mean=3)) + prod(dnorm(x,mean=-3))) }
> y <- arms(c(-2,2), bimodal,
+            function(x) all(x>(-10))*all(x<(10)), 500)
> plot(y, main="Mixture of bivariate Normals", asp=1)
```

Multivariate ARMS



Application



Data: bivariate quarterly time series of differenced log of seasonally adjusted real US money "M1" and GNP.

Model

Seemingly Unrelated Time Series – Local level

$$F_t = I_2, \qquad G_t = I_2,$$
$$V_t = V, \qquad W_t = q \cdot V$$

$$\theta_t = (\mu_t^{M1}, \mu_t^{GNP})'$$

Prior:

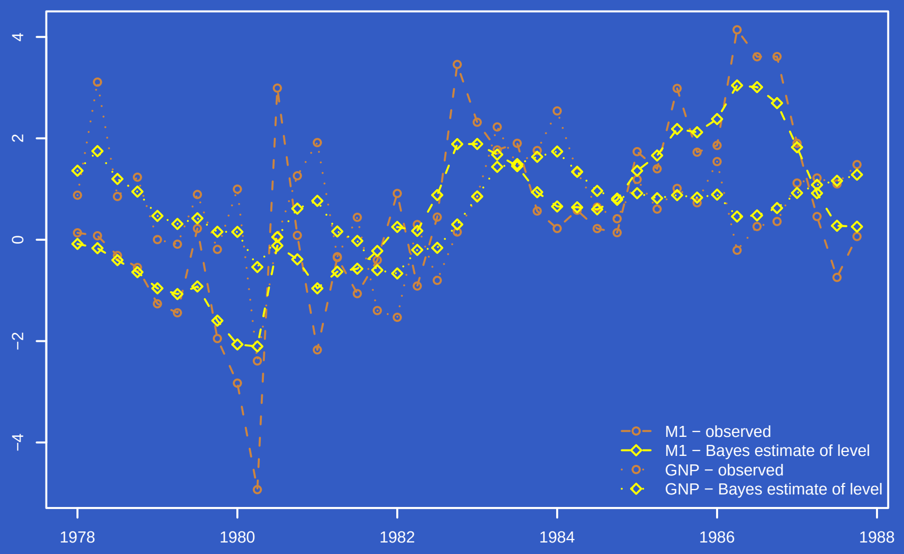
$$V \sim \mathcal{IW}, \qquad q \sim \mathrm{Unif}(\epsilon, M)$$

Gibbs Sampler

Generate in turn

1. $p(\Theta_n \mid V, q, \mathcal{Y}_n)$ — Forward Filtering Backward Sampling
2. $p(V \mid \Theta_n, q, \mathcal{Y}_n)$ — Inverse Wishart
3. $p(q \mid \Theta_n V, \mathcal{Y}_n)$ — ARMS

Estimated levels



Recap

- User-friendly, flexible package for DLM analysis
- Fast and numerically stable
- Focus on Bayesian, but also includes MLE
- A preliminary version of the package dlm can be downloaded at the URL

http://definetti.uark.edu/~gpetris/DLM