

Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

TAREA 3

EJERCICIOS

Esta tarea tiene por objeto familiarizarte con el filtro y suavizador de Kalman aplicado a uno de los modelos más simples (el modelo univariante de nivel local). Tareas sucesivas incidirán sobre las mismas técnicas generalizadas a modelos más complejos.

Tienes entre las sugerencias y ayudas al final una muestra de como efectuar filtrado de Kalman en los ejemplos muy simples a los que te enfrentas en esta tarea. Buena parte de lo que se te pide fue objeto de demostración en clase. Tu trabajo es entender y luego modificar y ampliar lo que se te proporciona.

1. El fichero `nilo.dge` (lee mediante un `dget`) contiene el caudal medio anual del Nilo en Assuan, medido en unidades adecuadas, para los años de 1871 a 1970.

- a) Entre las sugerencias y ayudas al final podrás ver un ejemplo de filtrado en que (arbitrariamente) se ha tomado $\sigma_\epsilon = 500$ y $\sigma_\eta = 100$. Repite el ejercicio tomando valores de σ_η que sean la mitad y la cuarta parte. ¿Qué observas en los resultados? ¿Por qué?
- b) Escribe una función que calcule el logaritmo de la verosimilitud a partir de la salida de la función `kf.11`. Recuerda que, salvo por términos constantes, el logaritmo de la verosimilitud es:

$$-\frac{1}{2} \sum_{i=1}^n \left(\log(F_t) + \frac{v_t^2}{F_t} \right)$$

Observa: la función que has de evaluar depende de los datos, de σ_ϵ y σ_η , que son tus parámetros, y de a_1, P_1 que puedes dejar fijos en $a_1 = 0$ y $P_1 = 10^7$, reflejando casi nula información *a priori* sobre el estado α_1 .

- c) Haciendo uso de la función `optim`, obtén el máximo de la función de verosimilitud escrita en el apartado anterior, y en consecuencia los estimadores máximo verosímiles de σ_ϵ y σ_η . Compara con los que reporta [2] en su página 13.

- d) Empleando los estimadores máximo-verosímiles de los parámetros puedes filtrar los datos una vez más, lo que te dará una estimación máximo verosímil de la trayectoria del estado. Hazlo, incluyendo intervalos de confianza.
- e) Haz una gráfica no del estado, sino de su varianza estimada, P_t . ¿Que observas?
2. Estima un modelo AR(I)MA para los mismos datos y compara los resultados intramuestrales de predicción una etapa hacia adelante. Lógicamente, para que la comparación tenga sentido el número de parámetros estimados no puede ser mucho mayor en un modelo que en otro. ¿Cual es el número de parámetros en el modelo de nivel local? (Ayuda: ¿Que pasaría si dejásemos tender σ_η^2 a infinito? ¿Y si hiciéramos $\sigma_\eta^2 = 0$?.)
3. Una de las ventajas del filtro de Kalman es su carácter acomodaticio a los datos faltantes.
- a) Modifica la función `kf.ll` de modo que admita valores NA en el argumento `y`.
- b) Modifica la función que hayas escrito para calcular el logaritmo de la verosimilitud, de manera que haga el cálculo correspondiente cuando hay valores perdidos.
- c) Fija a NA los valores de la serie `nilo` entre 1890 y 1900. Reestima por máxima verosimilitud los parámetros σ_ϵ y σ_η y vuelve a filtrar la serie. ¿Que valores toma a_t para los t correspondientes a observaciones y_t perdidas? ¿Y P_t ? ¿Es lógico?

AYUDAS, SUGERENCIAS Y COMPLEMENTOS

Sobre R:

- Programar el filtro de Kalman para el modelo de nivel local univariante es inmediato. Mira la función `kf.ll` a continuación. Admite como *inputs* la serie `y`, la media y varianza iniciales del estado (`a1` y `P1`) y los parámetros σ_ϵ y σ_η , desviaciones típicas respectivas de las ecuaciones de observación y estado (en `sigmaeps` y `sigmaeta` respectivamente):

```

kf.ll <-
function(y,sigmaeps,sigmaeta,a1,P1) {
  n <-length(y)
  v<-Fv<-P<-a<-rep(0,n)
  a[1] <- a1 ; P[1] <- P1
  for (i in 1:(n-1)) {
    v[i] <- y[i] - a[i]
    Fv[i] <- P[i] + sigmaeps^2
    K <- P[i] / Fv[i]
    a[i+1]<- a[i] + K*v[i]
    P[i+1]<- P[i]*(1-K) + sigmaeta^2
  }
}

```

```
    }  
    v[n] <- y[n] - a[n]  
    Fv[n] <- P[n] + sigmaeps^2  
    return(list(a=a,P=P,Fv=Fv,v=v))  
  }  
}
```

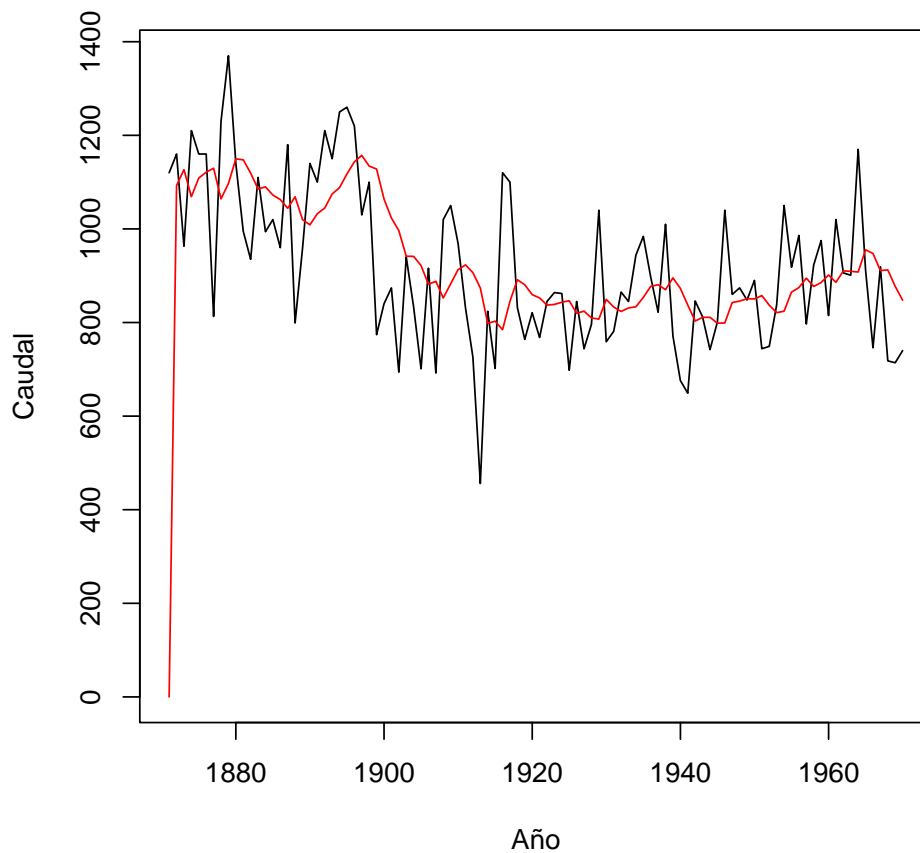
Para emplearla, puede invocarse así (empleamos valores arbitrarios de σ_ϵ y σ_η y un valor $P_1 = 10^7$ tan grande, que equivale a decir que nada sabemos del estado inicial):

```
> nilo <- dget("nilo.dge")  
> nilo.filt <- kf.ll(nilo, 500, 100, 0, 10^7)
```

En `nilo.filt$a` tenemos entonces la estimación de los estados, y en `nilo.filt$P` la de sus respectivas varianzas. Podemos convertir los primeros a formato de serie temporal y representarlos conjuntamente con los datos originales:

```
> a <- ts(nilo.filt$a, start = 1871, frequency = 1)  
> b <- ts.union(nilo, a)  
> plot(b, plot.type = "single", col = 1:2, main = "Caudal real y filtrado del  
+       xlab = "Año", ylab = "Caudal")
```

Caudal real y filtrado del Nilo



Obsérvese lo rápido que se adapta el valor medio inicial del estado, a_1 .

- En clase vimos el modo de empleo de la función `optim`. Basta, en general suministrar a `optim` una función (como segundo argumento) y un vector de valores iniciales de los parámetros como primer argumento. Si se tiene, también se puede suministrar una expresión analítica para el gradiente y el hessiano. Obsérvese el ejemplo siguiente, en que tratamos de minimizar la función

$$f(p_1, p_2) = p_1^2 + (p_2 - 2)^2.$$

(La solución es trivial, $p_1 = 0, p_2 = 2$.) Si facilitamos como valores iniciales $p_1 = 2$ y $p_2 = 3$, la función `optim` llega al mínimo con rapidez:

```
> f <- function(p) {
+   return(p[1]^2 + (p[2] - 2)^2)
+ }
> optim(p = c(2, 3), fn = f)

$par
[1] -0.0001821230  1.9999234271
```

```

$value
[1] 3.903221e-08

$counts
function gradient
      63      NA

$convergence
[1] 0

$message
NULL

```

- Observa: la función que pases a `optim` para su minimización debe depender a través de su argumento sólo de un vector de parámetros —que es respecto de los cuáles se realiza la minimización—. En el caso de la verosimilitud, necesitas que dependa además de las condiciones iniciales a_1 y P_1 y de los datos. No importa: puedes comenzar la definición de la función de este modo:

```

logver <- function(p)
{
  y <- nilo
  a1 <- 0 ; P1 <- 10^7
  nilo.filt <- kf.ll(y,sigmaeps=p[1],sigmaeta=p[2],
                    a1=a1,P1=P1)
  ...
}

```

- Recuerda que la función `optim` *minimiza* la función que le pases. Si quieres *maximizar* una verosimilitud puedes recurrir a: i) Facilitarle una función que calcule la verosimilitud con signo cambiado, o ii) Emplear el argumento `fnscale` (mira la documentación).
- Tienes alternativas como funciones de minimización en R: `nlm` y `nlminb`.

Sobre los datos:

- Los datos de caudal del Nilo son empleados en el libro [2], que seguimos en esta segunda parte del curso. La notación también sigue la de ellos y puedes comparar algunos de los resultados que obtengas con los que el libro proporciona.

Misceláneas:

- Observa: la función `kf.ll` calcula F_t y v_t , entre otras cosas, sirviéndose de los valores que recibe de σ_ϵ y σ_η . Si escribes una función que exprese el logaritmo de la verosimilitud en términos de F_t y v_t devueltos por `kf.ll` y la maximizas (o minimizas su negativo) obtendrás estimadores máximo verosímiles de σ_ϵ y σ_η . Para comparar con los de σ_ϵ^2 y σ_η^2 que reporta [2] en su página 13, has de elevar los primeros al cuadrado.

- La pregunta relativa al número de parámetros en un modelo de nivel local (al final de la Cuestión ??) no es fácil de responder: tiene interés que entiendas la pregunta (y veas por qué la respuesta no es simple). No te preocupes si no das con una respuesta.

Lectura sugerida:

- Puedes emplear el texto repetidamente citado más arriba, [2], especialmente §2.1 a §2.3.
- Manuales que tratan del filtro de Kalman o de modelos estructurales simples —como el de nivel local— son: [5], [4] y el bastante más antiguo y más orientado a ingenieros [1].
- Recientemente ha aparecido [3], que es una introducción clara y bien escrita.
- En español existe [6], pero se separa bastante en contenido y motivación de nuestro trabajo en este curso.

Referencias

- [1] B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Prentice-Hall, 1979.
- [2] J. Durbin and S.J. Koopman. *Time Series Analysis by State Space Methods*. Oxford Univ. Press, 2001.
- [3] R.L. Eubank. *A Kalman filter primer*. Chapman & Hall/CRC, 2006.
- [4] A.C. Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge Univ. Press, 1989.
- [5] R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications*. Springer Verlag, 2000.
- [6] M.J. Valderrama and J.C. Ruiz-Molina. *Filtrado de Kalman. Aplicaciones a la Ingeniería y Economía*. EUB, S.L, 1996.