Kalman Filter

Background with a general form

State space models are a class of time series models that were initially popular with engineers, before being widely adopted in navigating sailing and air traffic trajectory prediction.

There are various types of filtering and tracking algorithms can be used to predict the future state based on the current state. Here I am giving a short example of state prediction based on Kalman Filter and EM estimation, which can be potentially used to contribute to the exploration of the challenge “NATS - UK Air Traffic Services Provider”.

The smoothing, interpolation and prediction of time series data have always been an important research topic. The fundamental idea of Kalman filter is to adjust and predict value based on recursively balancing standard deviation with their trust worthiness. Here is an example of theoritical framework and implementation with sample dataset.

Letting $t = 1, \dots, T$ denote denote time, then univariate data $y_1, \dots y_T$ are modelled by the following three equations.

$$y_t = {Z_t}^T \alpha_t + \epsilon_t$$

$$\alpha_t = F_t \alpha_{t-1} + \xi_t$$

where

$$\epsilon_t - N(0, \sigma^2)$$

$$\xi_t - N(0, Q)$$

$$\alpha_0 - N(a_0, Q_0)$$

- α_t is a $p * 1$ dimensional state vector at time t.
- Z_t is a $p * 1$ dimensional design vector at time t.
- F_t is a $p * p$ transition matrix at time t.
- σ^2 is the univariate observation variance for $y_1, \dots y_T$.
- Q is a $p * p$ dimensional variance matrix for the state vector $\alpha_1, \dots \alpha_T$.
- $a_0 Q_0$ are the $p * 1$ mean vector and $p * p$ variance matrix for the initial state α_0.

Estimation for the unknown state parameters $\alpha_0, \alpha_1, \dots \alpha_T$ and the hyperparameters σ^2, Q, a_0, Q_0 is achieved using a combination of the EM algorithm and the Kalman filter, using an iterative algorithm.

Briefly, the overall algorithm applies the Kalman filter and smoother algorithm conditional on the current values of the hyperparameters ${\sigma^2}^{(p)}, Q^{(p)}, a_0^{(p)}, Q_0^{(p)}$, which are then themselves updated conditional on the new values of the state parameters. This two step process is iterated until convergence of the hyperparameters.

Specifically, the Kalman filter and smoother algorithm, estimates $a_{t|k}$ and$V_{t|k}$, which are the conditional expectation and variance of α_t given data up to and including time k, i.e.$y_1, \dots y_k$. Using this notation, the Kalman filter and smoother algorithm is as follows:

Kalman filter

1. Initialise the state vector by setting $a_{0|0} = a_0$ and $V_{0|0} = Q_0$. Then apply steps 2 and 3 iteratively for $t = 1, \dots, T$.
2. **Prediction** Calculate $a_{t|t-1}$ and $V_{t|t-1}$ as

$$a_{t|t-1} = F_t a_{t-1|t-1}$$

$$V_{t|t-1} = F_t V_{t-1|t-1} {F_t}^T + Q$$

3. ** Filtering** Calculate $a_{t|t}$ and $V_{t|t}$ as

$$a_{t|t} = a_{t|t-1} + K_t(y_t - {Z_t}^T a_{t|t-1})$$

$$V_{t|t} = V_{t|t-1} - K_t {Z_t}^T V_{t|t-1})$$

where

$$K_t = V_{t|t-1} Z_t [{Z_t}^T V_{t|t-1} Z_t + \sigma^2]^{-1}$$

Then once the Kalman filter has run, the Kalman smoother can be implemented as follows.

Kalman smoother

For $t = 1, \ldots, T$ recursively calculate

$$a_{t-1|T} = a_{t-1|t-1} + B_t(a_{t|T} - a_{t|t-1})$$

$$V_{t-1|T} = V_{t-1|t-1} + B_t(V_{t|T} - V_{t|t-1}){B_t}^T$$

where

$$B_t = V_{t-1|t-1} {F_t}^T {V_{t|t-1}}^{-1}$$

Then the overall algorithm to estimate the state variables and the hyperparameters is as follows.

Algorithm

1. Choose starting values for the hyperparameters $({\sigma^2}^{(0)}, Q^{(0)}, a_0^{(0)}, Q_0^{(0)})$. Then for $p = 0, 1, 2 \ldots$
2. Use the Kalman smoother to compute the smoothed estimates of the state vector $(a^{(p)}_{(0|T)}, a^{(p)}_{(1|T)}, \ldots, a^{(p)}_{(T|T)})$, their associated variances V with the values of the hyperparameters replaced by their current values $({\sigma^2}^{(p)}, Q^{(p)}, a_0^{(p)}, Q_0^{(p)})$.
3. Using the EM algorithm, compute the updated versions of the hyperparameters $({\sigma^2}^{(p+1)}, Q^{(p+1)}, a_0^{(p+1)}, Q_0^{(p+1)})$

$$a_0^{(p+1)} = a_{0|T}^{(p)}$$

$$Q_0^{(p+1)} = V_{0|T}^{(p)}$$

$$\sigma^{2(p+1)} = \frac{1}{T} \sum [(y_t - {Z_t}^T a_{t|T}^{(p)})^2 + {Z_t}^T Z_{t|T}^{(p)} Z_t]$$

$$Q^{(p+1)} = \frac{1}{T} \sum (a_{t|T}^{(p)} - F_t a_{t-1|T}^{(p)})(a_{t|T}^{(p)} - F_t a_{t-1|T}^{(p)})^T + V_{t|T}^{(p)} - F_t B_t^{(p)} V_{t|T}^{(p)} - {V_{t|T}^{(p)}}^T B_t^{(p)} F_t^T + F_t V_{t-1|T}^{(p)} F_t^T$$

4. If the values of the hyperparameters have not changed from the previous iteration then stop, else return to step 2. In practice this means calculating

$$diff = |\sigma^{2(p+1)} - \sigma^{2(p)}| + |Q^{(p+1)} - Q^{(p)}| + |{a_0}^{(p+1)} - {a_0}^{(p)}| + |{Q_0}^{(p+1)} - {Q_0}^{(p)}|$$

and determining if this value is less than some fixed tolerance such as 0.001. In this algorithm $B_t^{(p)}$ is defined for the Kalman smoother above.

Implementing in dataset

In here I demonstrate the code by writting into 4 functions with implementing the simple univariate first order random walk model as:

$$y_t = \alpha_t + \epsilon_t$$

$$\alpha_t = \alpha_{t-1} + \xi_t$$

where the state vector α_t and the transition variance Q are univariate.

All the general form above can be simplfied and realized as

Kalman filter

1. Initialize the state vector by setting $a_{0|0} = a$ and $V_{0|0} = Q_0$. Then apply steps 2 and 3 iteratively for $t = 1, \ldots, T$.
2. Prediction Calculate $a_{t|t-1}$ and $V_{t|t-1}$:

$$a_{t|t-1} = a_{t-1|t-1}$$

$$V_{t|t-1} = V_{t-1|t-1} + Q$$

3. Filtering Calculate $a_{t|t}$ and $V_{t|t}$:

$$K_t = V_{t|t-1}/(V_{t|t-1} + \sigma^2) = (V_{t-1|t-1} + Q)/(V_{t-1|t-1} + Q + \sigma^2)$$

$$a_{t|t} = a_{t|t-1} + K_t(y_t - a_{t|t-1}) = a_{t-1|t-1} + K_t(y_t - a_{t-1|t-1})$$

$$V_{t|t} = V_{t|t-1} - K_t V_{t|t-1} = (1 - K_t)(V_{t-1|t-1} + Q)$$

Kalman smoother

For $t = T, \ldots, 1$ recursively calculate :

$$B_t = V_{t-1|t-1}/V_{t|t-1} = V_{t-1|t-1}/(V_{t-1|t-1} + Q)$$

$$V_{t-1|T} = V_{t-1|t-1} + B_t(V_{t|T} - V_{t|t-1})B_t = V_{t-1|t-1} + B_t^2(V_{t|T} - V_{t-1|t-1} - Q)$$

EM algorithm

Choose starting values for hyperparameters $(\sigma^{2\,(0)}, Q^{(0)}, a_0^{(0)}, Q_0^{(0)})$. Then for $p = 0, 1, 2, 3, \ldots$ use the Kalman smoother (with current hyperparameters $(\sigma^{2\,(p)}, Q^{(p)}, a_0^{(p)}, Q_0^{(p)})$)to compute the smoothed estimates of $(a_{0|T}^{(p)}, a_{1|T}^{(p)}, \ldots, a_{T|T}^{(p)})$ and $(V_{0|T}^{(p)}, V_{1|T}^{(p)}, \ldots, V_{T|T}^{(p)})$. Then use the EM algorithm, compute the update hyperparameters $(\sigma^{2\,(p+1)}, Q^{(p+1)}, a_0^{(p+1)}, Q_0^{(p+1)})$

$$a_0^{(p+1)} = a_{0|T}^{(p)}$$

$$Q_0^{(p+1)} = V_{0|T}^{(p)}$$

$$\sigma^{2\,(p+1)} = \frac{1}{T}\int_{t=1}^{T}[(y_t - a_{t|T}^{(p)})^2 + V_{t|T}^{(p)}]$$

$$Q^{(p+1)} = \frac{1}{T}\int_{t=1}^{T}[(a_{t|T}^{(p)} - a_{t-1|T}^{(p)})^2 + V_{t|T}^{(p)} + V_{t-1|T}^{(p)} - 2 * B_t^{(p)} V_{t|T}^{(p)}]$$

Check, whether the hyperparameters change from the previous iteration. If no, then stop.

$$diff = |\sigma^{2\,(p+1)} - \sigma^{2\,(p)}| + |Q^{(p+1)} - Q^{(p)}| + |a_0^{(p+1)} - a_0^{(p)}| + |Q_0^{(p+1)} - Q_0^{(p)}|$$

```
Kalman <- function(y, hp){
  Index_T <- length(y)
  sig2 <- hp[1]
  Q <- hp[2]
  a0 <- hp[3]
  Q0 <- hp[4]

  #all variable only from 1 to t
  a_t <- numeric(Index_T) #a_t[t] means at|t, a_t[t-1] means at-1|t-1
  V_t <- numeric(Index_T) #V_t[t] means Vt|t, V_t[t-1] means Vt-1|t-1
  K <- numeric(Index_T)
  a_T <- numeric(Index_T)
  V_T <- numeric(Index_T)
  B <- numeric(Index_T)

  #the 1st time prediction and filtering, a0|0 = a0, V0|0 = Q0
  K[1] <- (Q0 + Q) / (Q0 + Q + sig2)
  a_t[1] <- a0 + K[1] * (y[1] - a0)
  V_t[1] <- (1 - K[1]) * (Q0 + Q)

  #then apply step 2 and 3 iteratively from t = 2 to T
  for(t in 2:Index_T){
    K[t] <- (V_t[t-1] + Q) / (V_t[t-1] + Q + sig2)
    a_t[t] <- a_t[t-1] + K[t] * (y[t] - a_t[t-1])
    V_t[t] <- (1 - K[t]) * (V_t[t-1] + Q)
  }

  #then Kalman Smoother from T to 2
  a_T[Index_T] <- a_t[Index_T] #set at|t = aT|T
  V_T[Index_T] <- V_t[Index_T] #set Vt|t = VT|T
  for(t in Index_T:2){
    B[t] <- V_t[t-1] / (V_t[t-1] + Q)
    a_T[t-1] <- a_t[t-1] + B[t] * (a_T[t] - a_t[t-1])
    V_T[t-1] <- V_t[t-1] + B[t]^2 * (V_T[t] - V_t[t-1] - Q)
  }
  #when t = 1
  B[1] <- Q0 / (Q0 + Q)
  a_T0 <- a0 + B[1] * (a_T[1] - a0)
  V_T0 <- Q0 + B[1]^2 * (V_T[1] - Q0 - Q)

  a_T <- c(a_T0, a_T) #from 0, 1, 2, ... , T
  V_T <- c(V_T0, V_T) #from 0, 1, 2, ... , T
  list("state" = a_T, "variances" = V_T, "B" = B)
}
```

```
KalmanEM <- function(y, hp){
  Index_T <- length(y)
  t <- 0 #timer

  #then for p from 0 to inf
  diff <- 1111111
  while(diff > 0.001){
    # choose starting values for the hyperparameters
    t <- t + 1
    sig2_p <- hp[1]
    Q_p <- hp[2]
    a0_p <- hp[3]
    Q0_p <- hp[4]

    data <- Kalman(y, hp)
    a_T <- data$state
    V_T <- data$variances
    B <- data$B

    a0_p1 <- a_T[1]
    Q0_p1 <- V_T[1]
    sig2_p1 <- sum((y - a_T[-1])^2 + V_T[-1]) / Index_T
    Q_p1 <- sum((a_T[-1] - a_T[1:Index_T])^2 + V_T[-1] - 2 * B * V_T[-1] + V_T[1:Index_T]) / Index_T

    hp <- c(sig2_p1, Q_p1, a0_p1, Q0_p1)
    diff <- abs(sig2_p1 - sig2_p) + abs(Q_p1 - Q_p) + abs(a0_p1 - a0_p) + abs(Q0_p1 - Q0_p)
  }
  list("state" = a_T, "variances" = V_T, "hp" = hp, "time" = t)
}
```

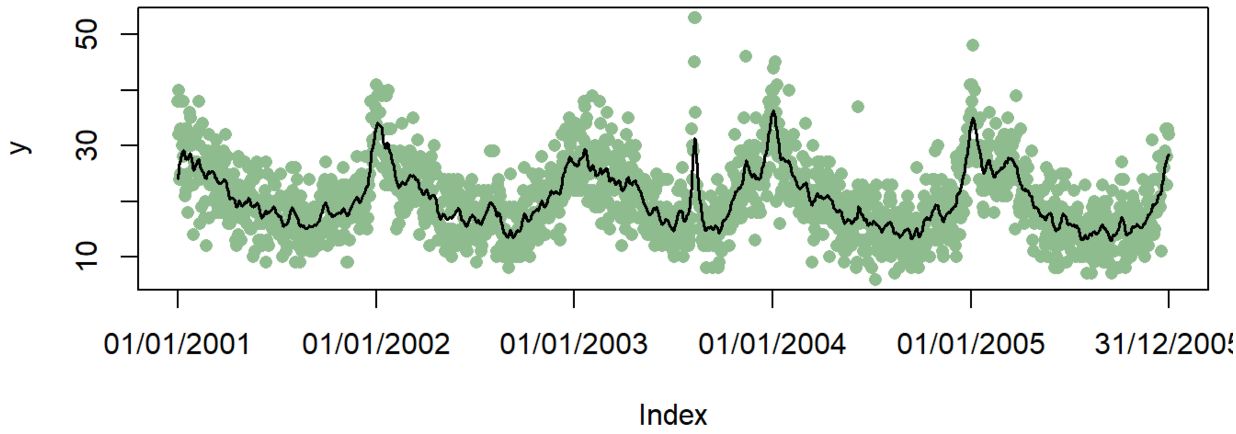
Implementing in dataset

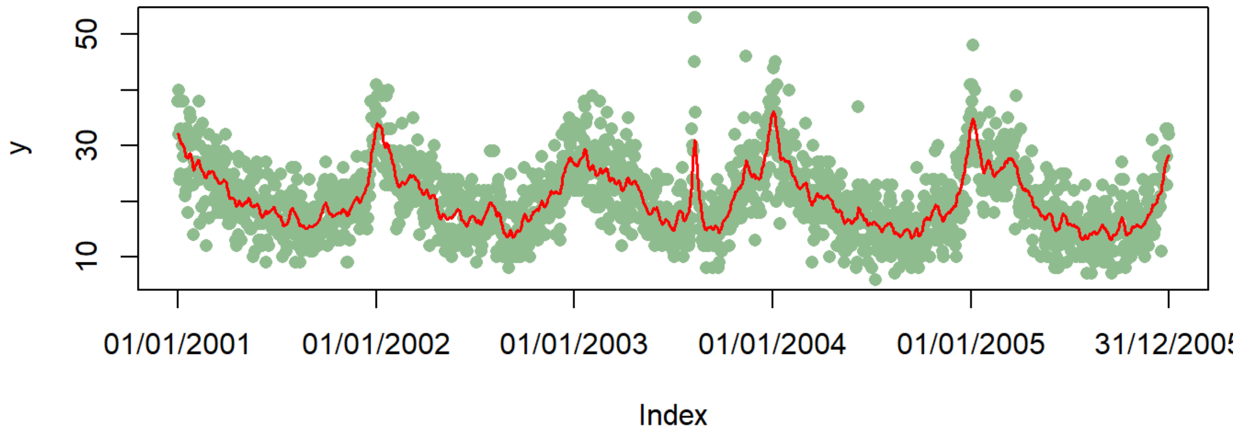
The simplified random walk model is applyed to the respiratory mortality data set. Specifically, the data are the daily numbers of respiratory mortalities from Greater London between 2001 and 2005. The initial values of the hyperparameters are set as (σ2 = 20; Q = 1; a0 = 20; Q0 = 1). A graph of the observed data (as dots) over time is plotted, with a solid line displaying the estimated state vector.

```
x <- read.csv("C:/Users/heqi2/Desktop/data.csv")
hp <- c(20, 1, 20, 1)
y <- x[, 2]
n <- nrow(x)
t <- 1:n

K <- Kalman(y, hp)
KEM <- KalmanEM(y, hp)
round(KEM$hp,3)
```

```
## [1] 19.147  0.895 32.146  0.019
```

```
par(mfrow = c(2, 1))
par(mar = c(4, 4, 0.5, 0.7))

plot(t, y, pch = 16, col = "darkseagreen", xaxt = "n", xlab = "Index", ylab = "y")
axis(side = 1, at = c(1, 366, 731, 1096, 1462, 1826),
     labels = c("01/01/2001", "01/01/2002", "01/01/2003", "01/01/2004", "01/01/2005", "31/12/2005"))
lines(t, K$state[-1], type = "l",col = "black",lwd = 1.7)

plot(t, y, pch = 16, col = "darkseagreen", xaxt = "n", xlab = "Index", ylab = "y")
axis(side = 1, at = c(1, 366, 731, 1096, 1462, 1826),
     labels = c("01/01/2001", "01/01/2002", "01/01/2003", "01/01/2004", "01/01/2005", "31/12/2005"))
lines(t, KEM$state[-1], type = "l",col = "red",lwd = 1.5)
```





The estimates of the hyperparameters are 19.147, 0.895, 32.146, and 0.019 and the plot shows the Kalman filter performs pretty well in the date smoothing.

Reference: *Multivariate statistical modelling based on generalized linear models*. Chapter 8. Ludwig Fahrmeir and Gerhard Tutz