Lissage exponentiel (compléments du Chapitre 6)

Yves Aragon*
Université Toulouse Capitole

1er novembre 2023

Somme finie ou infinie

La somme des poids
$$c_i = \alpha(1-\alpha)^i,\ i = 0, 1, \cdots$$
fait 1 si l'on va jusqu'à l'infini. Examinons la somme réelle des poids quand on arrête la somme à 10, 20, 30, 40 observations, pour alpha = .1, .2,.3..

```
> alpha <- seq(.1, .3, by = .1)
> arret <- seq(10, 40, by = 10)
> n.al <- length(alpha)
> n.arret <- length(arret)
> cumul <- matrix(0, nrow = n.al, ncol = n.arret)
> rownames(cumul) <- as.character(alpha)
> colnames(cumul) <- as.character(arret)
> poids <- function(alf, i) {
+   wgh <- rep(0, i)
+   wgh[1] <- alf
+   for(k in 2:i) {
+     wgh[k] <- wgh[k - 1] * (1 - alf)
+   }
+   sum(wgh)
+ }
> for (m in 1:length(alpha)) {
+   for (n in 1:length(arret)){
+     cumul[m, n] <- poids(alpha[m], arret[n])
+   }
+ }
> round(cumul, digits = 2)
```

```
      10   20   30   40
0.1 0.65 0.88 0.96 0.99
```

*yves.aragon@gmail.com

```
0.2 0.89 0.99 1.00 1.00
0.3 0.97 1.00 1.00 1.00
```

On voit qu'on atteint 0.99 en 40 observations si $\alpha = 0.1$, en 20 observations si $\alpha = 0.2$ et en moins de 20 observations si $\alpha = 0.3$. L'approximation est donc acceptable.

Exercice 6.1 (Compléments sur `fmsales`)

1. Examinons la sortie `ets0`.

```
> require("forecast")
> require("expsmooth")
> ets0 <- ets(fmsales, model = "ANN")
> summary(ets0)

ETS(A,N,N)

Call:
 ets(y = fmsales, model = "ANN")

  Smoothing parameters:
    alpha = 0.7316

  Initial states:
    l = 23.4625

  sigma:  3.6083

     AIC     AICc      BIC
418.9693 419.3831 425.3507

Training set error measures:
                    ME     RMSE       MAE        MPE
Training set 0.2013219 3.549585 2.350232 0.09847766
                 MAPE     MASE        ACF1
Training set 6.949233 0.946531 -0.00819019

> str(ets0, width = 60, strict.width = "cut")

List of 19
 $ loglik    : num -206
 $ aic       : num 419
 $ bic       : num 425
 $ aicc      : num 419
 $ mse       : num 12.6
 $ amse      : num 19.8
 $ fit       :List of 4
  ..$ value  : num 413
  ..$ par    : num [1:2] 0.732 23.463
  ..$ fail   : int 0
  ..$ fncount: int 45
 $ residuals : Time-Series [1:62] from 1 to 62: -0.4064 1...
```

```
 $ fitted    : Time-Series [1:62] from 1 to 62: 23.5 23.2 ..
 $ states    : Time-Series [1:63, 1] from 0 to 62: 23.5 23..
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr "l"
 $ par       : Named num [1:2] 0.732 23.463
  ..- attr(*, "names")= chr [1:2] "alpha" "l"
 $ m         : num 1
 $ method    : chr "ETS(A,N,N)"
 $ series    : chr "fmsales"
 $ components: chr [1:4] "A" "N" "N" "FALSE"
 $ call      : language ets(y = fmsales, model = "ANN")
 $ initstate : Named num 23.5
  ..- attr(*, "names")= chr "l"
 $ sigma2    : num 13
 $ x         : Time-Series [1:62] from 1 to 62: 23.1 24.8 ..
 - attr(*, "class")= chr "ets"
```

C'est une liste qui contient entre autres : les résidus, `residuals(ets0)`, c'est-à-dire les $\widehat{\varepsilon}_t$ et les valeurs ajustées `ets0$fit`, c'est-à-dire les $\widehat{y}_t$, qui sont également les prédictions à l'horizon 1 sur la période d'observation, la série état, `ets0$states`.

2. L'état initial est noté `l`, on le trouve en `$fit$par[2]` et dans `$states[1]`.
3. `ets0$mse` = `ets0$sigma2` car le prédicteur est sans biais et donc l'erreur quadratique moyenne se confond avec la variance de l'innovation.
4. Les paramètres de ce modèle sont l'état initial et alpha.
5. Blancheur du résidu.
   Si l'on veut examiner la blancheur du bruit après estimation, on peut exécuter :

```
> require("caschrono")
> Box.test.2(residuals(ets0), nlag = c(3, 6, 9))

     Retard   p-value
[1,]      3 0.3877841
[2,]      6 0.7948727
[3,]      9 0.5588173
```

   Donc le modèle est satisfaisant.

Exercice 6.2 (Lissage exponentiel simple par la méthode de Holt-Winters)

1. Faire la prévision de `fmsale` à l'horizon 4 à l'aide de la fonction `HoltWinters()` ;
2. Comparer dans les deux approches, les valeurs du paramètre α, les vecteurs donnant le niveau.

Réponse.

```
> (ets0.hw <- HoltWinters(fmsales, alpha = NULL, beta = FALSE,
+                          gamma = FALSE))

Holt-Winters exponential smoothing without trend and without seasonal comp

Call:
HoltWinters(x = fmsales, alpha = NULL, beta = FALSE, gamma = FALSE)

Smoothing parameters:
```

```
 alpha: 0.7321555
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 32.59733
```

Et si l'on veut dessiner les deux ajustements par `ets` et par `HoltWinters`

```
plot(ets0.hw$fitted[, 1], ets0$fitted[-1])
```