

Lissage exponentiel (compléments du Chapitre 6)

Yves Aragon*

Université Toulouse 1 Capitole

2 avril 2011

Somme finie ou infinie

La somme des poids

$$c_i = \alpha(1 - \alpha)^i, i = 0, 1, \dots$$

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)
+ {
+ # renvoie les poids alpha*(1 - alpha)^j, j=0, i-1
+ 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)
```

*aragon@cict.fr

	10	20	30	40
0.1	0.65	0.88	0.96	0.99
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)
> require(caschrono)
> ets0 = ets(fmsales, model="ANN")
> summary(ets0)
```

ETS (A,N,N)

Call:

```
ets(y = fmsales, model = "ANN")
```

Smoothing parameters:

```
alpha = 0.7312
```

Initial states:

```
l = 23.4673
```

```
sigma: 3.5496
```

AIC	AICc	BIC
416.9693	417.1727	421.2236

In-sample error measures:

ME	RMSE	MAE	MPE	MAPE
0.20127166	3.54958451	2.35036107	0.09804668	6.94976638
MASE				
0.94658312				

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

List of 18

```
$ loglik      : num -206
$ aic         : num 417
$ bic         : num 421
$ aicc        : num 417
$ mse         : num 12.6
$ amse        : num 19.4
$ fit         :List of 5
..$ par       : Named num [1:2] 0.731 23.467
.. ..- attr(*, "names")= chr [1:2] "alpha" "l"
```

```

..$ value      : num 413
..$ counts     : Named int [1:2] 53 NA
.. ..- attr(*, "names")= chr [1:2] "function" "gradient"
..$ convergence: int 0
..$ message    : NULL
$ residuals   : Time-Series [1:62] from 1 to 62: -0.4111 1...
$ fitted      : Time-Series [1:62] from 1 to 62: 23.5 23.2 ..
$ states      : ts [1:63, 1] 23.5 23.2 24.4 24.3 23.5 ...
..- attr(*, "dimnames")=List of 2
.. ..$ : NULL
.. ..$ : chr "1"
..- attr(*, "tsp")= num [1:3] 0 62 1
$ par         : Named num [1:2] 0.731 23.467
..- attr(*, "names")= chr [1:2] "alpha" "1"
$ m           : num 1
$ method      : chr "ETS(A,N,N)"
$ components  : chr [1:4] "A" "N" "N" "FALSE"
$ call        : language ets(y = fmsales, model = "ANN")
$ initstate   : Named num 23.5
..- attr(*, "names")= chr "1"
$ sigma2      : num 12.6
$ 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 $\hat{\epsilon}_t$ et les valeurs ajustées `ets0$fit`, c'est-à-dire les \hat{y}_t , qui sont également les prédictions à l'horizon 1 sur la période d'observation, la série état, `ets0$states`.

2. L'état initial est noté 1, on le trouve en `fitpar[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 :

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

	Retard	p-value
[1,]	3	0.3880425
[2,]	6	0.7951073
[3,]	9	0.5586496

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])
```