TS 0.5 Smoothing

Benjamin Kedem

January 2021

1 Error Model

Given a time series y_t, $t = 1, 2, ..., N$, an error model consisting of a trend $f(t)$ plus a random error is defined as,

$$y_t = f(t) + u_t \tag{1}$$

where $f(t)$ is an unspecified deterministic function of time, and u_t a noise term with mean 0 and variance σ^2. See See (2), Section 3.3.1.

For simplicity we shall assume that $f(t)$ is slowly varying, and that the u_t are uncorrelated. It should be noted that model (1) is not a regression model in a traditional sense since $f(t)$ is unspecified. Thus traditional regression methods do not apply here.

The trend $f(t)$ at time t can be estimated by a weighted average, y_t^*, of y-values in a small neighborhood of t:

$$y_t^* = \sum_{s=-m}^{m} c_s y_{t+s} \tag{2}$$

where the weights sum up to 1,

$$\sum_{s=-m}^{m} c_s = 1 \tag{3}$$

The procedure which gives y_t^* as a weighted average of neighboring y-values is referred to as *smoothing*. As t varies, y_t^* defines a weighted moving average. It should be noted that by our construction, the y_t are uncorrelated whereas the y_t^* are correlated.

Since we assume that $f(t)$ is slowly varying, the values of $f(t+s)$, $s = \pm 1, \pm 2, ..., \pm m$, for small m are close to $f(t)$ and y_t^* is nearly unbiased,

$$\mathrm{E}(y_t^*) = \sum_{s=-m}^{m} c_s f(t+s) \approx f(t) \sum_{s=-m}^{m} c_s = f(t) \tag{4}$$

with variance,

$$\mathrm{Var}(y_t^*) = \sigma^2 \sum_{s=-m}^{m} c_s^2 \tag{5}$$

The idea is to choose the c_s such that $\mathrm{Var(y_t^*)}$ is considerably smaller than $\mathrm{Var(y_t)}$. Clearly, when $c_s \geq 0$, $\mathrm{Var(y_t^*)} \leq \sigma^2 = \mathrm{Var(y_t)}$.

1.1 A Method for Getting c_s

We shall now outline a method for getting the c_s weights by local fitting of a polynomial to neighboring y-values.

For a fixed t, we fit a polynomial of degree q, $a_0 + a_1 s + \cdots + a_q s^q$, $s = -m, ..., 0, ..., m$, to $2m+1$ y-values, $y_{t-m}, ..., y_t, ..., y_{t+m}$, by least squares. Thus, we minimize

$$\sum_{s=-m}^{m} [y_{t+s} - (a_0 + a_1 s + \cdots + a_q s^q)]^2 \tag{6}$$

with respect to $a_0, ..., a_q$. The normal equations are,

$$a_0 \sum_{s=-m}^{m} s^j + a_1 \sum_{s=-m}^{m} s^{j+1} + \cdots a_q \sum_{s=-m}^{m} s^{j+q} = \sum_{s=-m}^{m} s^j y_{t+s}, \quad j = 0, 1, .., q \tag{7}$$

For every even even j, the coefficients of $a_1, a_3, ...$ are 0. For every odd j, the coefficients of $a_0, a_2, ...$ are 0. Since a_0 corresponds to $y_t \approx f(t)$, we are interested in the least squares estimate of a_0, and consider even j only. So let $j = 2i$, $i = 0, 1, ..., [q/2]$, where $[q/2] = q/2$ for even q, and $[q/2] = (q-1)/2$ for odd q. Then the normal equations reduce to

$$a_0 \sum_{s=-m}^{m} s^{2i} + a_2 \sum_{s=-m}^{m} s^{2i+2} + \cdots a_{2[q/2]} \sum_{s=-m}^{m} s^{2i+2[q/2]} = \sum_{s=-m}^{m} s^{2i} y_{t+s}, \quad i = 0, 1, .., [q/2] \tag{8}$$

and we see that the optimal a_0 is a weighted average,

$$\hat{a}_0 = \sum_{s=-m}^{m} c_s y_{t+s} \tag{9}$$

The c_s are clearly symmetric, $c_s = c_{-s}$, and they sum up to 1. To see this, note that the coefficients of the a_j depend on m only and not on the y's. Therefore, if $y_{t+s} = a$, $s = -m, ..., m$, then the best polynomial is a itself. Thus,

$$\sum_{s=-m}^{m} c_s = 1$$

and we have the desired

$$y_t^* = \hat{a}_0 = \sum_{s=-m}^{m} c_s y_{t+s}$$

For $q = 0$ or $q = 1$ and a general m we have from (8) a simple moving average

$$y_t^* = \frac{\sum_{s=-m}^{m} y_{t+s}}{2m+1} \tag{10}$$

It can be shown that for $q = 2$ or $q = 3$ and a general m,

$$y_t^* = \frac{\sum_{s=-m}^{m}[3(3m^2 + 3m - 1) - 15s^2]y_{t+s}}{(2m-1)(2m+1)(2m+3)} \tag{11}$$

For short time series, say $N = 30$, (11) with $m = 2$ or $m = 3$ is sufficient. For $m = 2$ the smoothed series is

$$y_t^* = \frac{1}{35}(-3y_{t-2} + 12y_{t-1} + 17y_t + 12y_{t+1} - 3y_{t+2}) \tag{12}$$

and for $m = 3$

$$y_t^* = \frac{1}{21}[-2y_{t-3} + 3y_{t-2} + 6y_{t-1} + 7y_t + 6y_{t+1} + 3y_{t+2} - 2y_{t+3}] \tag{13}$$

An R function to compute the weights for $q = 2$ or $q = 3$ and a general m, is

```
#s=-m,...,m, m=2,3,...
cs <- function(m,s){
+ (3*(3*m^2+3*m-1)- 15*s^2)/((2*m-1)*(2*m+1)*(2*m+3))}
```

Using the theory of linear regression it is easy to get confidence intervals for $f(t)$.

2 Local Linear Regression

An extension of the above method replaces the sum of squares in (6) by a weighted sum of squares, and carries out weighted least squares. In this case we minimize

$$\sum_{s=-m}^{m} w(t,s)[y_{t+s} - (a_0 + a_1 s + \cdots + a_q s^q)]^2 \tag{14}$$

This is the basic idea behind the *locally weighted scatterplot smoother* or *lowess*. The R function `lowess` carries out this type of weighted least squares for smoothing purposes. The default for `lowess` uses simple linear models,

$$\sum_{s=-m}^{m} w(t,s)[y_{t+s} - (a_0 + a_1 s)]^2 \tag{15}$$

See (3) Section 6.6, and (4), p. 230.

3 Smoothing Splines

Another well known smoothing idea makes use of spline fitting. Let $f(t)$ be a cubic spline, and consider the penalized sum of squares with smoothing parameter λ,

$$\sum_{s=1}^{N}[y_t - f(t)]^2 + \lambda \int_a^b [f''(t)]^2 dt \tag{16}$$

Minimizing (16) results in a great deal of smoothing. The R function `smooth.spline` carries out spline smoothing, choosing λ by cross validation. See (3) Section 7.2.3, and (4), p. 230.

Figure 1 provides an illustration of polynomial smoothing for $(q = 0, m = 3)$, $(q = 2, m = 2)$, $(q = 2, m = 3)$, and $N = 100$, versus lowess, spline smoothing, and running medians. Running medians is similar to a moving average where the median is taken instead of the average.

Figure 1: Local polynomial smoothing versus spline smoothing, Lowess, and running medians.

References

[1] De Oliveira, V., Kedem, B., and Sort, D. (1997). Bayesian prediction of transformed Gaussian random fields. *Jour. Americ. Statist. Assoc.*, Vol. 92, 1422-1433.

[2] Anderson, T.W. (1971). *The Statistical Analysis of Time Series*, Wiley, New York.

[3] Seber, G.A.F. and Lee, A.J. (2003). *Linear Regression Analysis*, Wiley, New York.

[4] Venables, W.N. and Ripley, B.D. (2002). *Modern Applied Statistics With S* 4th ed., Springer, New York.


```
Local polynomial smoothing versus  spline smoothing,
Lowess, and running medians.

#Fig 1
#q=0 or 1, m=3

t <- 1:100
u <- rnorm(100,0,0.25)
y <- cos(0.05*t)+u
#plot(t,y,type="l")


ys <- y
for(i in 4:97){
ys[i]<-
(y[i-3]+y[i-2]+y[i-1]+y[i]+
y[i+1]+y[i+2]+y[i+3])/7}


plot(t,y,type="l",cex=1.25)
lines(t,ys,type="o")
title("Moving Average: q=0,m=3", cex=1.25)

#Fig 2

t <- 1:100
u <- rnorm(100,0,0.25)
y <- cos(0.05*t)+u
#plot(t,y,type="l")

#m=2:
ys <- y
```

```
for(i in 3:98){
ys[i]<- (-3*y[i-2]+12*y[i-1] + 17*y[i] + 12*y[i+1]-3*y[i+2])/35}


plot(t,y,type="l")
lines(t,ys,type="o")
title("Polynom. Smooth: q=2,m=2", cex=1.25)

#Fig 3
t <- 1:100
u <- rnorm(100,0,0.25)
y <- cos(0.05*t)+u
#plot(t,y,type="l")

ys <- y
for(i in 4:97){
ys[i]<-
(-2*y[i-3]+3*y[i-2]+6*y[i-1]+7*y[i]+
6*y[i+1]+3*y[i+2]-2*y[i+3])/21}

plot(t,y,type="l",cex=1.25)
lines(t,ys,type="o")

title("Polynom. Smooth: q=2,m=3", cex=1.25)

#Fig 4

#compare with cubic spline (Venables and Ripley p. 230)

t <- 1:100
u <- rnorm(100,0,0.25)
y <- cos(0.05*t)+u
#plot(t,y,type="l")

Can control the smoothing by df:
smooth.spline(x,g, df = 10), lty = 2)


plot(t,y,type="l",cex=1.25)
lines(smooth.spline(t, y),type="o")
title("Smoothing Spline", cex=1.25)

#Fig 5

#Lowess  and  Loess (Venables and Ripley p. 230)

t <- 1:100
u <- rnorm(100,0,0.25)
y <- cos(0.05*t)+u
#plot(t,y,type="l")

plot(t,y,type="l",cex=1.25)
lines(lowess(t,y))
title("Lowess", cex=1.25)
```

```
#Give the same as

#plot(t,y,type="l",cex=1.25)
#lines(loess.smooth(t,y))
#title("Loess.smooth", cex=1.25)


#Fig 6

###Smoothing by Runnning Medians: m=3

t <- 1:100
u <- rnorm(100,0,0.25)
y <- cos(0.05*t)+u
#plot(t,y,type="l")

ys <- y
for(i in 4:97){
ys[i]<- median(c(y[i-3],y[i-2],y[i-1],y[i],y[i+1],y[i+2],y[i+3]))}

plot(t,y,type="l",cex=1.25)
lines(t,ys,type="o")
title("Running Medians, m=3", cex=1.25)
```