Estadística no paramétrica

Regresión no paramétrica IV: Splines de suavizamiento

Jorge de la Vega Góngora

Departamento de Estadística,
Instituto Tecnológico Autónomo de México

14 de abril de 2023

Splines (continuación)

Splines de suavizamiento I

- Los *splines de suavizamiento* son una forma de regresión no paramétrica, que se utiliza como suavizadores. Se tienen pares de puntos (X_i, Y_i) para $i = 1, 2, \dots, n$, y de nuevo, queremos estimar la media condicional, que podemos expresar como:

$$Y_i|\mathbf{X}_i = m(\mathbf{X}_i) + \epsilon_i$$

- Queremos un estimador que ajuste bien los datos, pero al mismo tiempo tenga un cierto grado de suavidad. La medida natural de suavidad asociada a una función $s \in l^2[a, b]$ es $\int_a^b s^2(t)\, dt$, mientras que una medida estándar de bondad de ajuste a los datos es $n^{-1} \sum_{i=1}^n (Y_i - s(X_i))^2$. Entonces, un método para combinar estos dos criterios y obtener una medida de desempeño, es evaluar la calidad de s con la funcional

$$T(s) = \alpha \sum_{i=1}^{n} (Y_i - s(X_i))^2 + (1 - \alpha) \int_a^b (s''(t))^2\, dt$$

para $\alpha \in (0, 1)$. Se puede simplificar un poco la notación considerando $\lambda = \alpha/(1 - \alpha)$ y de esta forma obtenemos

Splines de suavizamiento II

$$T(s) = \sum_{i=1}^{n} (Y_i - s(X_i))^2 + \lambda \int_a^b (s''(t))^2 \, dt, \qquad \lambda > 0$$

es exactamente un spline cúbico natural.

- Entonces la funcional anterior, representa como un costo, y tiene dos partes:
  1. $\sum_{i=1}^{n} (Y_i - s(X_i))^2$ se minimiza por el polinomio de interpolación, y
  2. $\int_a^b (s''(t))^2 \, dt$ se minimiza por una línea recta.
- El parámetro λ que se conoce como el *parámetro de suavizamiento*, intercambia la importancia de estos costos competitivos: si λ es pequeño, la solución es un spline de interpolación que favorece la bondad de ajuste y si λ es grande, la minimizante es una línea recta, favoreciendo la suavidad de la curva.

Splines de suavizamiento en estadística

- Los splines de suavizamiento usualmente se consideraron más como herramientas de análisis numérico, hasta que Grace Wahba (1934 –) mostró que los splines tienen propiedades estadísticas muy útiles y merecían ser considerados como un método para análisis de regresión no paramétrico.
- Grace Wahba también desarrolló el método generalizado de validación cruzada.
- Una revisión de las aplicaciones de los splines de suaviamiento se puede encontrar en el artículo de Eubank (1984).
- Finalmente, un estudio detallado de los métodos no paramétricos de regresión se puede encontrar en el libro de Eubank: *Spline Smoothing and Nonparametric Regression* Marcel Dekker (1988).

Grace Wahba

Splines de suavizamiento como estimadores lineales I

- El estimador spline también es lineal en las observaciones, $\hat{s} = \mathbf{S}(\lambda)\mathbf{Y}$, para una matriz de suavizamiento $\mathbf{S}(\lambda)$. El algoritmo de Reinsch (1967) se utiliza para calcular $\mathbf{S}$ con la ecuación:

$$\mathbf{S}(\lambda) = (\mathbf{I} + \lambda \mathbf{Q}\mathbf{R}^{-1}\mathbf{Q}')^{-1},$$

donde $\mathbf{Q}_{n\times(n-2)}$ y $\mathbf{R}_{(n-2)} \times (n-2)$ son matrices con las siguientes estructuras:

$$\mathbf{Q} = \begin{bmatrix} q_{12} & & & \\ q_{22} & q_{23} & & \\ q_{32} & q_{33} & & \\ & q_{43} & & \\ & & \dots & \\ & & & q_{n-2,n-1} \\ & & & q_{n-1,n-1} \\ & & & q_{n,n-1} \end{bmatrix}, \mathbf{R} = \begin{bmatrix} r_{22} & r_{23} & & \\ r_{32} & r_{33} & & \\ & r_{43} & & \\ & & \dots & \\ & & & r_{n-2,n-1} \\ & & & r_{n-1,n-1} \end{bmatrix}$$

donde los valores q y r están dados por la siguiente expresión. Si $h_i = X_{i+1} - X_i$ suponiendo los valores de X's ordenados:

Splines de suavizamiento como estimadores lineales II

$$q_{ij} = \begin{cases} \frac{1}{h_{j-1}} & i = j-1 \\ -\left(\frac{1}{h_{j-1}} + \frac{1}{h_j}\right) & i = j \\ \frac{1}{h_j} & i = j+1 \end{cases}$$

y

$$r_{ij} = \begin{cases} \frac{1}{6}h_{j-1} & i = j-1 \\ -\frac{1}{3}(h_{j-1} + h_j) & i = j \\ \frac{1}{6}h_j & i = j+1 \end{cases}$$

- Los detalles del algoritmo de Reinsch se pueden encontrar en el libro de Green y Silverman (1994).

Seleccionando y evaluando el estimador de regresión I

- La evaluación del estimador se puede hacer considerendo jackknife (validación cruzada con 1-afuera). Sea $\hat{s}_{(i)\lambda}(x)$ el estimador de $s(x)$ el estimador basado en λ dejando fuera (X_i, Y_i). Definimos el score:

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{s}_{(i)\lambda}(x))^2$$

  Entonces el valor de λ que minimiza $CV(\lambda)$ producirá en promedio los mejores estimadores.

- Para los splines de suavizamiento, y en particular, para los suavizadores lineales de la forma $\hat{s} = \mathbf{S}(\lambda)\mathbf{Y}$, el procedimiento computacional se puede simplificar un poco, porque la ecuación anterior se simplifica a

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{Y_i - \hat{s}_\lambda(x)}{1 - S_{ii}(\lambda)}\right]^2$$

  donde los valores $S_{ii}(\lambda)$ son elementos diagonales de $diag(\mathbf{S}(\lambda))$. Esto representa una ventaja porque cuando n es grande, construir y calcular **S** se vuelve complicado.

Seleccionando y evaluando el estimador de regresión II

- Otra simplificación para encontrar los mejores suavizadores es la *validación cruzada generalizada* (GCV). Se reemplaza la expresión $1 - S_{ii}(\lambda)$ por la que considera el promedio de la diagonal:

$$GCV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{Y_i - \hat{s}_\lambda(x)}{1 - tr(\mathbf{S}(\lambda))/n} \right]^2$$

Ejemplo: datos de S& P I

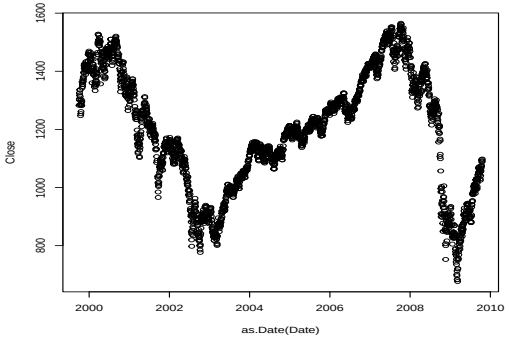
- La función más básica que implementa las ideas que se mostraron previamente es `smooth.spline`. La estructura básica es `smooth.spline(x, y, cv = F)`, donde el parámetro `cv` calcula jackknife (`cv=TRUE`) o GCV (`cv=FALSE`).
- Los siguientes datos contiene el valor del índice S& P para 2529 días. Podemos ajustar un spline a los rendimientos observados del índice.

```
SyP <- read.csv("../data/SPhistory.short.csv")
head(SyP)  # noten que los datos están de más reciente a más viejo.
```

```
        Date    Open    High     Low   Close     Volume Adj.Close
1 2009-10-20 1098.64 1098.64 1086.16 1091.06 5396930000   1091.06
2 2009-10-19 1088.22 1100.17 1086.48 1097.91 4619240000   1097.91
3 2009-10-16 1094.67 1094.67 1081.53 1087.68 4894740000   1087.68
4 2009-10-15 1090.36 1096.56 1086.41 1096.56 5369780000   1096.56
5 2009-10-14 1078.68 1093.17 1078.68 1092.02 5406420000   1092.02
6 2009-10-13 1074.96 1075.30 1066.71 1073.19 4320480000   1073.19
```

```
with(SyP,plot(as.Date(Date),Close))
```

Ejemplo: datos de S& P II




```
Syp <- rev(SyP)  #se invierten los datos
syp <- diff(log(SyP$Close))  #Se calculan los log-rendimientos
```

Ejemplo: datos de S& P III

- El objetivo es usar los log-rendimientos de un día para ver si podemos pronosticar los del día siguiente. Graficando los datos de t contra los de $t-1$:

```
syp.hoy <- syp[-length(syp)]
syp.manana <- syp[-1]
(syp.spline <- smooth.spline(x=syp.hoy, y =syp.manana, cv =T) )
```

```
Warning in smooth.spline(x = syp.hoy, y = syp.manana, cv = T): cross-validation with non-unique 'x' values seems doubtful
```

```
Call:
smooth.spline(x = syp.hoy, y = syp.manana, cv = T)

Smoothing Parameter  spar= 1.418323  lambda= 0.08179568 (14 iterations)
Equivalent Degrees of Freedom (Df): 3.772721
Penalized Criterion (RSS): 0.4835979
PRESS(l.o.o. CV): 0.0001929004
```

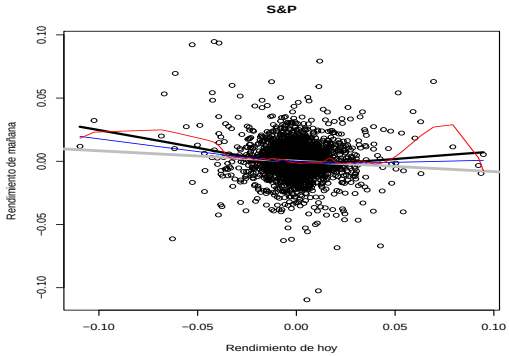
```
syp.spline$lambda # lambda estimada por validación cruzada
```

```
[1] 0.08179568
```

- el `warning` es porque tenemos dos días con rendimientos de cero.
- PRESS significa 'prediction sum of squares', y es el valor de $CV(\hat{\lambda})$ en el óptimo, $\hat{\lambda} = 0.0817957$

```
plot(syp.hoy, syp.manana, xlab="Rendimiento de hoy", ylab="Rendimiento de mañana",
     main ="S&P")
abline(lm(syp.hoy ~ syp.manana),col="gray",lwd=5)
lines(syp.spline,lwd=4)
lines(smooth.spline(x=syp.hoy, y =syp.manana, cv =T, lambda=0.35),col="blue")
lines(smooth.spline(x=syp.hoy, y =syp.manana, cv =T, lambda=0.00001),col="red")
```

Ejemplo: datos de S& P IV



Ejemplo: datos de S& P V

- Podemos usar la función de predicción para ese propósito (aunque seguramente hay mejores modelos a considerar que este):

```
predict(syp.spline, x=c(-0.01,0.01))
```

```
$x
[1] -0.01  0.01

$y
[1]  0.0007348649 -0.0010195141
```

Intervalos de confianza para splines I

- En el ejemplo anterior, comparamos el spline con un ajuste lineal. ¿La curvatura del spline es real o es debido a la variación aleatoria?
- Podemos ajustar una banda de confianza utilizando bootstrap como una opción para estimar su variabilidad.

```
X <- data.frame(today = syp.hoy, manana = syp.manana)
remuestreo <- function(x) x[sample(1:nrow(x),size=nrow(x),replace=T),]

estimador.splines <- function(x, m = 300){
        fit <- smooth.spline(x = x[,1], y = x[,2], cv=T)
        eval.grid <- seq(from = min(syp.hoy), to = max(syp.manana), length.out = m)
        return(predict(fit,x=eval.grid)$y)
}
```

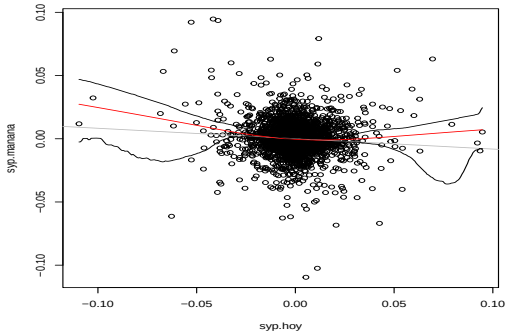
- Con las dos funciones anteriores, podemos calcular las bandas de confianza basadas en bootstrap. Tenemos que armarlos “a mano”, para obtener los valores que esperamos:

```
sp.ci <- function(X, B, alfa, m = 300){
        sp.principal <- estimador.splines(X,m=m)
        sp.boot <- replicate(B, estimador.splines(remuestreo(X),m = m))
        ci.l <- 2*sp.principal - apply(sp.boot,1,quantile,probs=1-alfa/2)
        ci.u <- 2*sp.principal - apply(sp.boot,1,quantile,probs=alfa/2)
        return(list(sp.principal = sp.principal, ci.inf = ci.l, ci.sup = ci.u,
                    x = seq(from = min(syp.hoy), to = max(syp.hoy), length.out = m)))
}
```

Intervalos de confianza para splines II

- Finalmente, aplicando los procedimientos anteriores

```
Resultados <- sp.ci(X,B=1000,alfa=0.05)
plot(syp.hoy,syp.manana)
abline(lm(syp.manana ~ syp.hoy),col="grey")
lines(x = Resultados$x, y = Resultados$sp.principal, col ="red")
lines(x = Resultados$x, y = Resultados$ci.inf)
lines(x = Resultados$x, y = Resultados$ci.sup)
```



Funciones base y grados de libertad I

- En esta sección revisaremos la representación de los splines en términos de un ajuste de mínimos cuadrados y cómo esta representación se asocia a los grados de libertad de un modelo.
- Hemos dicho que los splines son polinomios que se unen en piezas. En general, para un polinomio cúbico, podemos definir como una base funcional el conjunto $\{B_i(x) = x^{i-1} | i \in \{1, 2, 3, 4\}\}$.
- Podemos también elegir considerar funciones de regresión que sean combinaciones lineales de las funciones base:

$$m(x) = \sum_{j=1}^{4} \beta_j B_j(x)$$

Funciones base y grados de libertad II

- Para estimar los coeficientes del polinomio cúbico evaluamos la base en cada punto y podemos poner todos los resultados en una matriz de $n \times 4$:

$$\mathbf{B} = (B_{ij} = B_j(X_i))$$

y se podría utilizar mínimos cuadrados usando la matriz **B** en lugar de los datos **x**, para obtener los coeficientes β:

$$\hat{\beta} = (\mathbf{B}'\mathbf{B})^{-1}\mathbf{B}'\mathbf{Y}$$

- Como los splines son cubicos en partes, se puede proceder de manera similiar, pero hay que tener cuidado en la definición de las funciones base. Supondremos como antes que los valores ya están ordenados en magnitud.

Funciones base y grados de libertad III

- Los n nodos definen $n+1$ piezas o segmentos, de los que $n-1$ están entre los nodos, y dos de ellos están en los extremos, de $-\infty$ a X_1 y de X_n a ∞. Cada segmento requerirá un término constante, uno lineal, uno cuadrático y uno cúbico. Entonces el segmento que va de X_i a X_{i+1} requiere las funciones bases:

  $$I(X_i < x < X_{i+1}), (x-X_i)I(X_i < x < X_{i+1}), (x-X_i)^2 I(X_i < x < X_{i+1}), (x-X_i)^3 I(X_i < x < X_{i+}$$

  Parece entonces que se requerirán $4(n+1) = 4n+4$ funciones base.
- Sin embargo, el número de vectores en la base debe ser igual a la dimensión del espacio vectorial en consideración. Como los polinomios deben satisfacer restricciones en los nodos, además de las condiciones de diferenciabilidad en los nodos y la linealidad fuera de los nodos, el número de coeficientes se reduce a n.
- Una elección de base común para splines es:
  - $B_1(x) = 1$
  - $B_2(x) = x$
  - $B_i = \frac{(x-X_{i-2})_+^3 - (x-X_n)_+^3}{X_n - X_{i-2}} - \frac{(x-X_{n-1})_+^3 - (x-X_n)_+^3}{X_n - X_{n-1}}$, $i = 3, \dots, n$

Funciones base y grados de libertad IV

La elección de esta base es por la propiedad de que la segunda y tercera derivada de cada función B_j es cero fuera del intervalo (X_1, X_n).

- La función a minimizar para ajustar el spline cúbico en la base anterior se puede escribir como:

$$L(\lambda) = (\mathbf{Y} - \mathbf{B}\beta)'(\mathbf{Y} - \mathbf{B}\beta) + n\lambda\beta'\beta$$

donde la matriz tiene la información sobre la curvatura de las funciones base:

$$\Omega_{jk} = \int B_j''(x)B_k''(x)$$

- El spline se puede encontrar diferenciando con respecto a β:

$$\widehat{\beta} = (\mathbf{B}'\mathbf{B} + n\lambda)^{-1}\mathbf{B}'\mathbf{Y}$$

y entonces podemos ver que el estimador de splines es un estimador lineal:

$$s(x) = \widehat{m}(x) = \mathbf{B}\widehat{\beta} = \mathbf{B}(\mathbf{B}'\mathbf{B} + n\lambda)^{-1}\mathbf{B}'\mathbf{Y}$$

Funciones base y grados de libertad V

- Basados en esta representación, podemos proceder como en el caso de loess, definiendo los *grados de libertad efectivos* como $tr\left(\mathbf{B}(\mathbf{B}'\mathbf{B} + n\lambda)^{-1}\mathbf{B}'\right)$
- Intuitivamente, se puede ver que los grados de libertad efectivos disminuirán si se incrementa el valor de λ.En algunos paquetes de `R` en lugar de especificar λ se pueden especificar los grados de libertad, como en la función `s` para especificar un suavizador spline en el paquete `gam`.