Overfit & Cross Validation

Ciencia de Datos para la toma de decisiones en Economía

Ignacio Sarmiento-Barbieri

Agenda

Predicción y Error Predictivo

- El objetivo es predecir y dadas otras variables X. Ej: ingreso dadas las características del individuo

- Asumimos que el link entre y and X esta dado por el modelo:

$$y = f(X) + u \tag{1}$$

- donde $f(X)$ es cualquier función,

- u una variable aleatoria no observable $E(u) = 0$ and $V(u) = \sigma^2$

Predicción y Error Predictivo

- En la práctica no conocemos $f(X)$
- Es necesario estimarla $\hat{y} = \hat{f}(X)$
- Nuestra medida de riesgo/performance será generalmente el MSE

$$MSE(y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2$$

Predicción y Error Predictivo

- El *MSE* puede descomponerse al menos en dos partes

$$MSE(y) = MSE(\hat{f}) + \sigma^2 \tag{2}$$

- el error de estimar f con $\hat{f}$. (*reducible*)
- el error de no observar u. (*irreducible*)

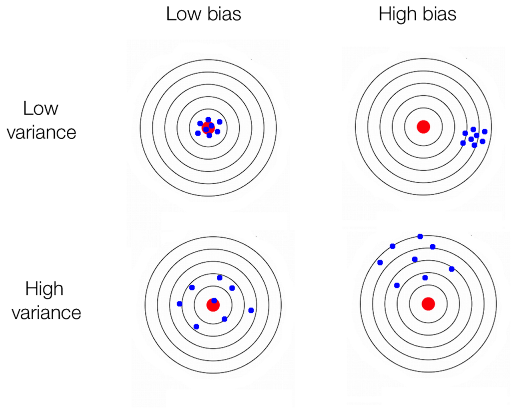
Predicción y Error Predictivo

- Descomponiendo un poco más:

$$
\begin{align}
MSE(y) &= MSE(\hat{f}) + \sigma^2 \tag{3} \\
&= Bias^2(\hat{f}) + V(\hat{f}) + Irreducible\ Error \tag{4}
\end{align}
$$

- Este resultado es muy importante,
  - Aparece el dilema entre sesgo y varianza

Prediction Error



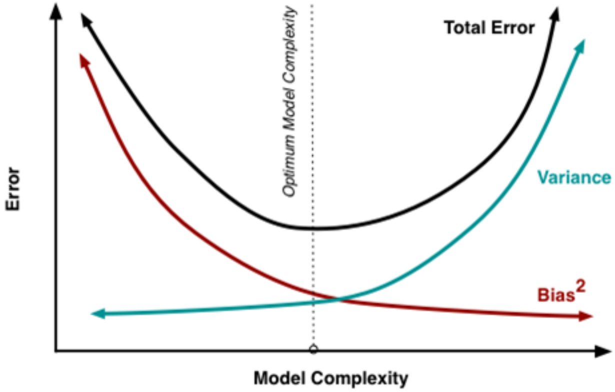
Source: https://tinyurl.com/y4lvjxpc

Dilema sesgo/varianza

- El secreto de ML: admitiendo un poco de sesgo podemos tener ganancias importantes en varianza

Dilema sesgo/varianza

- El secreto de ML: admitiendo un poco de sesgo podemos tener ganancias importantes en varianza



Source: https://tinyurl.com/y4lvjxpc

Predicción y regresión lineal

- El problema es:

$$y = f(X) + u \tag{5}$$

- proponemos que:

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \tag{6}$$

- El problema se reduce a encontrar los βs
  - Un camino es OLS

Predicción y regresión lineal

- Y el dilema sesgo varianza?

Predicción y regresión lineal

- Y el dilema sesgo varianza?
- Bajo los supuestos clásicos (Gauss-Markov) el estimador de OLS es insesgado:

$$
\begin{aligned}
E(X\hat{\beta}) &= E(\hat{\beta}_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_p X_p) && (7)\\
&= E(\hat{\beta}_1) + E(\hat{\beta}_2) X_2 + \cdots + E(\hat{\beta}_p) X_p && (8)\\
&= X\beta && (9)
\end{aligned}
$$

- $MSE(\hat{y})$ se reduce a $V(\hat{\beta})$

Complejidad y compensación de varianza/sesgo

- En la econometría clásica, la elección de modelos se resume a elegir entre modelos más pequeños y más grandes.
- Considere los siguientes modelos para estimar y:

$$y = \beta_1 X_1 + u_1$$

- $\hat{\beta}_1^{(1)}$ el estimador de OLS y on X_1
- La predicción es:

$$\hat{y}^{(1)} = \hat{\beta}_1^{(1)} X_1$$

$$y = \beta_1 X_1 + \beta_2 X_2 + u_2$$

- $\hat{\beta}_1^{(2)}$ y $\hat{\beta}_2^{(2)}$ con β_1 y β_2 los el estimador de OLS de y en X_1 y X_2.
- La predicción es:

$$\hat{y}^{(2)} = \hat{\beta}_1^{(2)} X_1 + \hat{\beta}_2^{(2)} X_2$$

Complejidad y compensación de varianza/sesgo

- Una discusión importante en la econometría clásica es la de la omisión de variables relevantes frente a la inclusión de variables irrelevantes.
  - Si el modelo (1) es verdadero entonces estimar el modelo más grande (2) conduce a estimadores ineficientes aunque no sesgados debido a que incluyen innecesariamente X_2.

Complejidad y compensación de varianza/sesgo

Ejemplo

```r
#Load Packages
require("tidyverse")
```

```
Loading required package: tidyverse

-- Attaching packages --------------------------------------- tidyverse 1.3.1 --

v ggplot2 3.3.6     v purrr   0.3.4
v tibble  3.1.7     v dplyr   1.0.9
v tidyr   1.2.0     v stringr 1.4.0
v readr   2.1.2     v forcats 0.5.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```r
require("fabricatr")
require("stargazer")
```

Complejidad y compensación de varianza/sesgo

Ejemplo

```r
#for reproducibility
set.seed(101010)



db1 <- fabricate(
  N = 10000,
  ability=rnorm(N,mean=.5,sd=2),
  schooling = round(runif(N, 2, 14)),
  logwage =rnorm(N, mean=7+.15*schooling, sd=2)
)
```

Complejidad y compensación de varianza/sesgo

Ejemplo

```
reg1<-lm(logwage~schooling,db1)
reg2<-lm(logwage~schooling+ability,db1)
stargazer(reg1,reg2,type="text")
```

```
##
## =====================================================================
##                                   Dependent variable:
##                      ------------------------------------------------
##                                         logwage
##                                (1)                        (2)
## ---------------------------------------------------------------------
## schooling                   0.145***                   0.145***
##                              (0.006)                    (0.006)
##
## ability                                                  0.007
##                                                         (0.010)
##
## Constant                    7.050***                   7.046***
##                              (0.051)                    (0.051)
##
## ---------------------------------------------------------------------
## Observations                 10,000                     10,000
## R2                            0.059                      0.059
## =====================================================================
## Note:                                   *p<0.1; **p<0.05; ***p<0.01
```

Complejidad y compensación de varianza/sesgo

Ejemplo

```
db1<- db1 %>% mutate(yhat_reg1=predict(reg1),
                     yhat_reg2=predict(reg2))

var(db1$yhat_reg1)
```

```
## [1] 0.2522197
```

```
var(db1$yhat_reg2)
```

```
## [1] 0.2524032
```

Complejidad y compensación de varianza/sesgo

- Una discusión importante en la econometría clásica es la de la omisión de variables relevantes frente a la inclusión de variables irrelevantes.
  - Si el modelo (1) es verdadero entonces estimar el modelo más grande (2) conduce a estimadores ineficientes aunque no sesgados debido a que incluyen innecesariamente X_2.
  - Si el modelo (2) se verdadero, estimar el modelo más pequeño (1) conduce a una estimación de menor varianza pero sesgada si X_1 también se correlaciona con el regresor omitido X_2.

Complejidad y compensación de varianza/sesgo
Ejemplo

```r
db2 <- fabricate(
  N = 10000,
  ability=rnorm(N,mean=.5,sd=2),
  schooling = round(runif(N, 2, 14)),
  schooling = round(ceiling(schooling+1*ability)),
  logwage =rnorm(N, mean=7+.15*schooling+.25*ability, sd=2)
)
```

Complejidad y compensación de varianza/sesgo

Ejemplo

```r
reg3<-lm(logwage~schooling,db2)
reg4<-lm(logwage~schooling+ability,db2)
stargazer(reg3,reg4,type="text")
```

```
##
## ===========================================================================
##                                    Dependent variable:
##                     -------------------------------------------------------
##                                         logwage
##                                (1)                           (2)
## ---------------------------------------------------------------------------
## schooling                   0.216***                      0.153***
##                              (0.005)                       (0.006)
##
## ability                                                   0.254***
##                                                            (0.011)
##
## Constant                    6.563***                      7.007***
##                              (0.051)                       (0.053)
##
## ---------------------------------------------------------------------------
## Observations                 10,000                        10,000
## R2                            0.152                         0.192
## ===========================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

Complejidad y compensación de varianza/sesgo

Ejemplo

```
db2$yhat_reg3<-predict(reg3)
db2$yhat_reg4<-predict(reg4)

var(db2$yhat_reg3)
```

```
## [1] 0.755213
```

```
var(db2$yhat_reg4)
```

```
## [1] 0.9538193
```

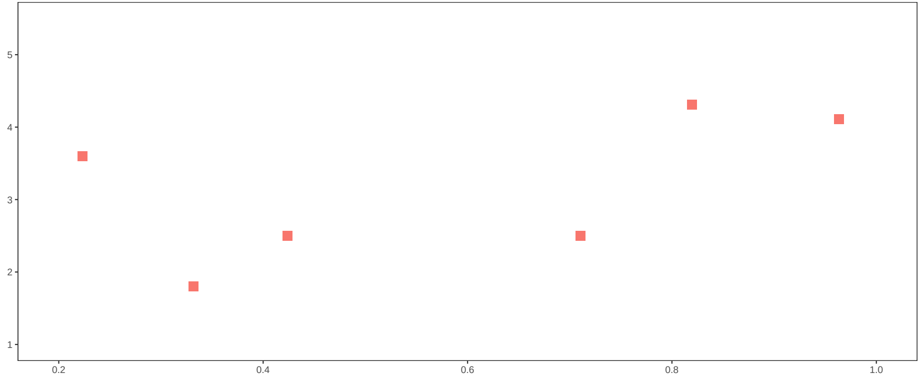
Complejidad y compensación de varianza/sesgo

- Una discusión importante en la econometría clásica es la de la omisión de variables relevantes frente a la inclusión de variables irrelevantes.
  - Si el modelo (1) es verdadero entonces estimar el modelo más grande (2) conduce a estimadores ineficientes aunque no sesgados debido a que incluyen innecesariamente X_2.
  - Si el modelo (2) se verdadero, estimar el modelo más pequeño (1) conduce a una estimación de menor varianza pero sesgada si X_1 también se correlaciona con el regresor omitido X_2.
- Esta discusión de pequeño vs grande siempre es con respecto a un modelo que se supone es verdadero.
- Pero en la práctica el modelo verdadero es desconocido!!!

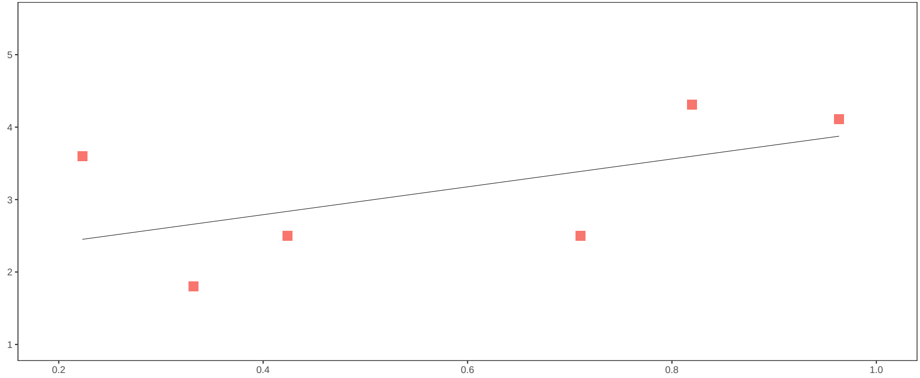
Complejidad y compensación de varianza/sesgo

- Elegir entre modelos/especificaciones implica un dilema *sesgo/varianza*
- La econometría clásica tiende a resolver este dilema abruptamente,
  - requiriendo una estimación no sesgada y, por lo tanto, favoreciendo modelos más grandes para evitar sesgos
- En esta configuración simple, los modelos más grandes son ”más complejos”, por lo que los modelos más complejos están menos sesgados pero son más ineficientes.
- Por lo tanto, en este marco muy simple, la complejidad se mide por el número de variables explicativas.
- Una idea central en el aprendizaje automático es generalizar la idea de complejidad,
  - Nivel óptimo de complejidad, es decir, modelos cuyo sesgo y varianza conducen al menor MSE.

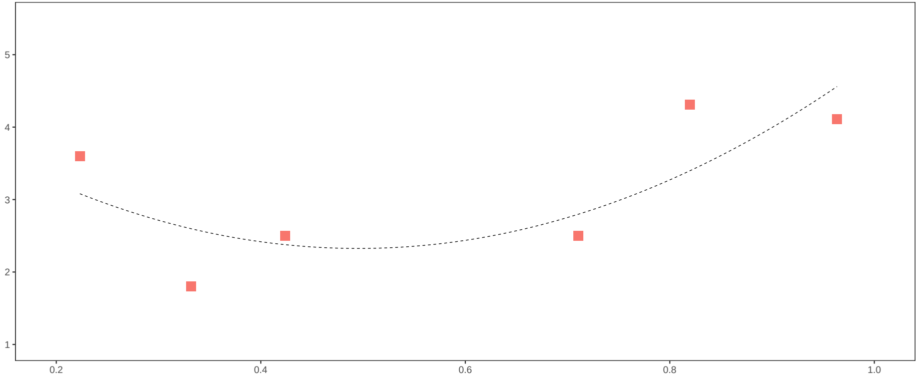
Overfit



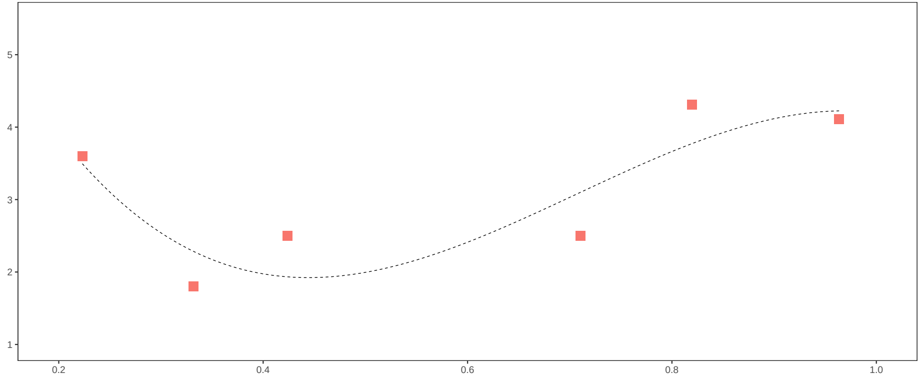
Overfit



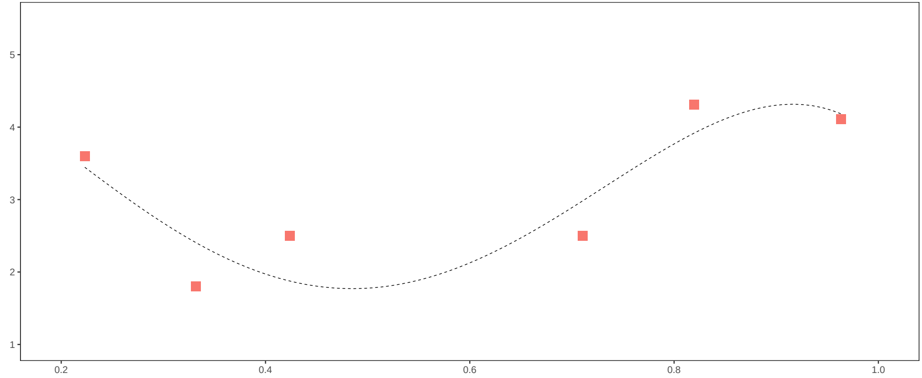
Overfit



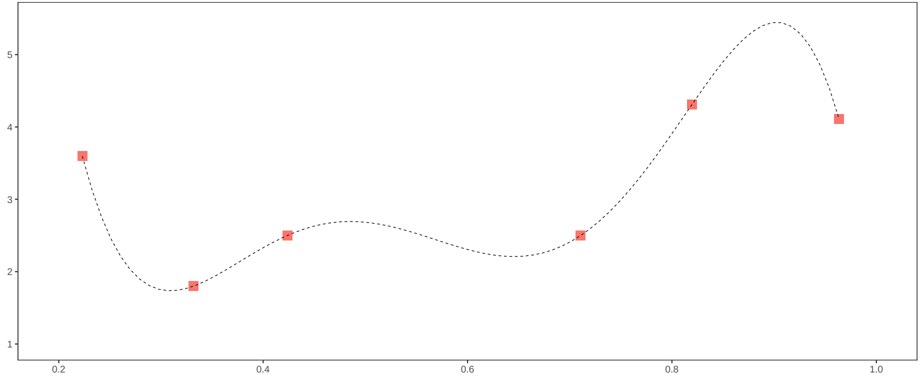
Overfit



Overfit



Overfit



Overfit

- Notemos que esto no es otra cosa que la suma de los residuales al cuadrado

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(X))^2 \tag{10}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{11}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(e)^2 \tag{12}$$

$$= RSS \tag{13}$$

- Esta medida nos da una idea de *lack of fit* que tan mal ajusta el modelo a los datos

Overfit

- Un problema del RSS es que nos da una medida absoluta de ajuste de los datos, y por lo tanto no esta claro que constituye un buen RSS.
- Una alternativa muy usada en economía es el R^2
- Este es una proporción (la proporción de varianza explicada),
  - toma valores entre 0 y 1,
  - es independiente de la escala (o unidades) de y

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{14}$$

$$= 1 - \frac{RSS}{TSS} \tag{15}$$

Overfit

- Suppose that the true model is $y = f(X) + u$ where f is a polynomial of degree p^*, with $E(u) = 0$ and $V(u) = \sigma^2$
- p^* is finite but unknown
- We fit polynomials with increasing degrees $p = 1, 2, ...$
- What happens when we increase the degree of the polynomial?

Overfit

- The expected prediction error of a regression fit $\hat{f}(X)$ at an input point $X = x_0$, is

$$
\begin{aligned}
MSE(x_0) &= MSE(y - \hat{f}(x_0) | X = x_0) \\
&= Bias^2(f, \hat{f}(x_0)) + V(\hat{f}(x_0)) + Irreducible\ Error
\end{aligned}
\tag{16}
$$

- The average expected prediction error

$$
\frac{1}{n} \sum_{i=1}^{N} MSE(x_i) \tag{17}
$$

Overfit

- Bias ?

Overfit

- Bias ?
- Variance:

$$\hat{f}(x_0) = \sum_{s=0}^{p} x_0^s \hat{\beta}_s = x_0' \hat{\beta} \tag{18}$$

where $x_0' = (1, x_0, , x_0^2, \ldots, x_0^p)$

$$V(\hat{f}(x_0)) = V(x_0'\hat{\beta}) = x_0'\sigma^2(X'X)^{-1}x_0 \tag{19}$$

Then

$$\frac{1}{n}\sum_{i=1}^{n} \sigma^2 x_i' \sigma^2 (X'X)^{-1} x_i = \sigma^2 \frac{p}{n} \tag{20}$$

After we "hit" p^* increasing complexity does not reduce the bias, but variance increases monotonically for σ^2 and n given

Proof

The fitted model for a polynomial of degree p is :

$$\hat{y}_i = x_i'\hat{\beta} \tag{21}$$

with $x_i' = (1, x_i, , x_i^2, \ldots, x_i^p)$ Then $V(y_i) = V(x_i'\hat{\beta}) = \sigma^2 x_i'(X'X)^{-1}x_i$. Now:

$$Average\, V(x_i'\hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n} \sigma^2 x_i'(X'X)^{-1}x_i \tag{22}$$

Proof

- Trace.
  - If $A_{m\times m}$ with typical element a_{ij}. The **trace** of A, $tr(A)$ is the sum of the elements of its diagonal: $tr(A) \equiv \sum_{s=1}^{m} a_{ii}$
  - Properties
    - For any square matrices A, B, and C: $tr(A+B) = tr(A) + tr(B)$
    - Cyclic property: $tr(ABC) = tr(BCA) = tr(CAB)$
    - If $m = 1$ tr(A)=A

Now we use traces.

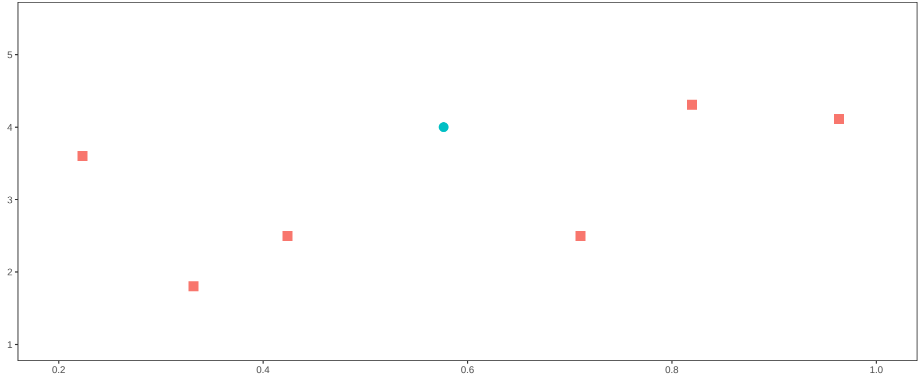
$$Average\, V(x_i'\hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\sigma^2(x_i'(X'X)^{-1}x_i) \tag{23}$$

$$\sum_{i=1}^{n} tr((X'X)^{-1}x_i'x_i) = tr(\sum_{i=1}^{n}(X'X)^{-1}x_i'x_i) = tr((X'X)^{-1}(X'X)) = p \tag{24}$$

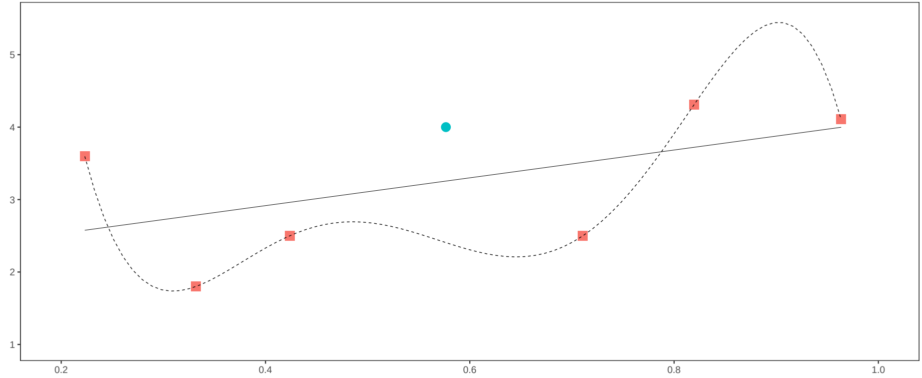
Overfit y Predicción fuera de Muestra

- ML nos interesa la predicción fuera de muestra
- Overfit: modelos complejos predicen muy bien dentro de muestra, pero tienden a hacer un trabajo fuera de muestra
- Hay que elegir el nivel adecuado de complejidad
- Como medimos el error de predicción fuera de muestra?
- R^2 no funciona: se concentra en la muestra y es no decreciente en complejidad

Overfit



Overfit



AIC

- Akaike (1969) fue el primero en ofrecer un enfoque unificado al problema de la selección de modelos.
- Su punto de vista fue elegir un modelo del conjunto f_i que funcionó bien cuando se evaluó sobre la base del rendimiento de la previsión.
- Su criterio, que ha llegado a llamarse criterio de información de Akaike, es

$$AIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - p_j \tag{25}$$

BIC

- Schwarz (1978) mostró que, si bien el enfoque *AIC* puede ser bastante satisfactorio para seleccionar un modelo de pronóstico
- Sin embargo, tiene la desafortunada propiedad de que es inconsistente, (cuando $n \to \infty$, tiende a elegir un modelo demasiado grande con probabilidad positiva)
- Schwarz (1978) formalizó el problema de selección de modelos desde un punto de vista bayesiano:

$$SIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - \frac{1}{2}p_j log(n) \tag{26}$$

AIC vs BIC

$$AIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - p_j \tag{27}$$

$$SIC(j) = log\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2\right) - p_j\frac{1}{2}log(n) \tag{28}$$

- Note que

$$\frac{1}{2}log(n) > 1 \;\; for \;\; n > 8 \tag{29}$$

- La penalidad de SIC es mayor que la penalidad de AIC,
- SIC tiende a elegir modelos más pequeños.
- En efecto, al dejar que la penalización tienda al infinito lentamente con n, eliminamos la tendencia de AIC a elegir un modelo demasiado grande.

Error de Prueba y de Entrenamiento

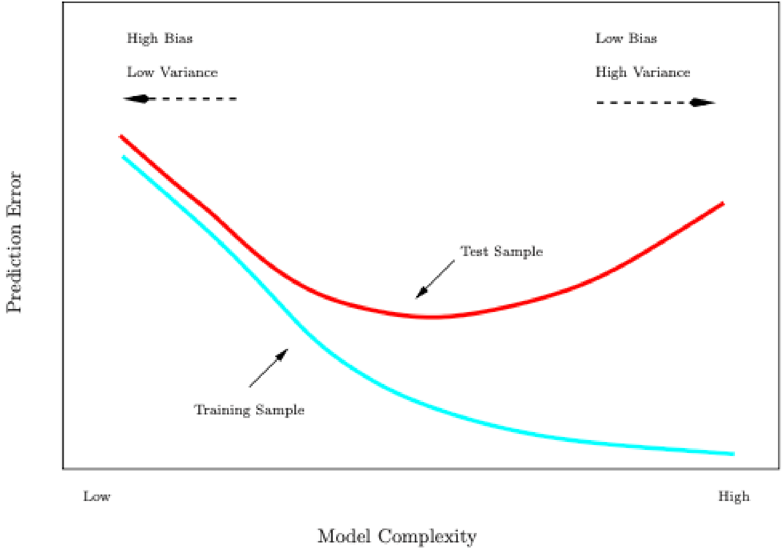
- Dos conceptos importantes
  - *Test Error*: es el error de predicción en la muestra de prueba (test)

$$Err_{\mathcal{T}est} = MSE[(y, \hat{y})|\mathcal{T}est] \tag{30}$$

  - *Training error*:es el error de predicción en la muestra de entrenamiento (training)

$$Err_{\mathcal{T}rain} = MSE[(y, \hat{y})|\mathcal{T}rain] \tag{31}$$

Error de Prueba y de Entrenamiento



Train and test samples

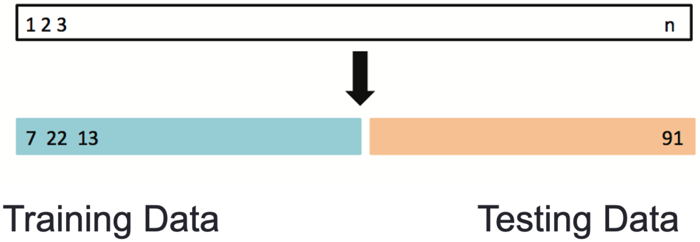
- Cómo elegimos $\mathcal{T}est$?

Train and test samples

- Cómo elegimos $\mathcal{T}est$?
- Una alternativa simple seria dividir los datos en dos:
  - Training sample: para construir/estimar/entrenar el modelo
  - Test sample: para evaluar el desempeño
- Desde una perspectiva estrictamente clásica
  - Tiene sentido si los datos de entrenamiento son iid de la población, incluso funciona si es iid condicional en X
  - Dos problemas con esta idea:
    - El primero es que, dado un conjunto de datos original, si parte de él se deja de lado para probar el modelo, quedan menos datos para la estimación (lo que lleva a una menor eficiencia).
    - Un segundo problema es cómo decidir qué datos se usarán para entrenar el modelo y cuáles probarlo.

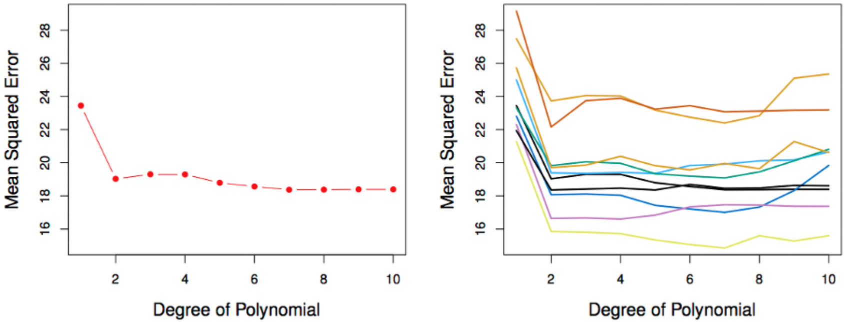
Enfoque de conjunto de validación

- Podemos entonces aproximar esta idea partiendo la muestra en 2



Enfoque de conjunto de validación

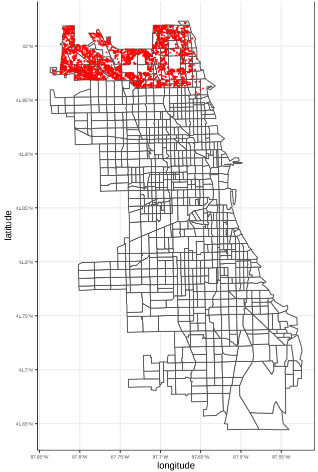
- Modelo $y = f(x) + u$ donde f es un polinomio de grado p^*.
  - Izquierda: error de predicción en la muestra de prueba para una sola partición
  - Derecha: error de predicción en la muestra de prueba para varias particiones
  - Hay un montón de variabilidad. (Necesitamos algo mas estable)



Example: Predicting House Prices in R

- `matchdata` in the *McSpatial package* for R.
- 3,204 sales of SFH Far North Side of Chicago in 1995 and 2005.
- This data set includes 18 variables/features about the home,
  - price sold
  - number of bathrooms, bedrooms,
  - latitude and longitude,
  - etc.
- in R:

```
require(mcspatial)#loads the package
data(matchdata) #loads the data
?matchdata # help/info about the data
```



Example: Predicting House Prices in R

- Train and Test samples
- 80% / 20% split

```
data(matchdata) #loads the data
set.seed(101010) #sets a seed
matchdata <- matchdata %>%
                      mutate(price=exp(lnprice), #transforms log prices
                      #to standard prices
                             holdout= as.logical(1:nrow(matchdata) %in%
                             sample(nrow(matchdata), nrow(matchdata)*.2))
                             #generates a logical indicator to divide
                              #between train and test set
                             )
test<-matchdata[matchdata$holdout==T,]
train<-matchdata[matchdata$holdout==F,]
```

Example: Predicting House Prices in R

- Naive approach: model with no covariates, just a constant
- $y = \beta_0 + u$

```r
specification1<-lm(price~1,data=train)
summary(specification1)
```

```
##
## Call:
## lm(formula = price ~ 1, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -258018 -127093  -24018   92732  598482
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   284018       4782   59.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 148300 on 961 degrees of freedom
```

Example: Predicting House Prices in R

In this case our prediction for the log price is the average train sample average

$$\hat{y} = \hat{\beta}_0 = \frac{\sum y_i}{n} = m$$

```r
coef(specification1)
```

```
## (Intercept)
##    284017.6
```

```r
mean(train$price)
```

```
## [1] 284017.6
```

Example: Predicting House Prices in R

- But we are concerned on predicting well our of sample,:

```
test$specification1<-predict(specification1,newdata = test)
with(test,mean((price-specification1)^2))
```

```
## [1] 22811540844
```

- Then the *test MSE* $= E((y - \hat{y})^2) = E((y - m)^2) = 2.2811541 \times 10^{10}$.
- This is our starting point, Can we improve it?

Example: Predicting House Prices in R

- How to improve it?
  - One way is using econ theory as guide
  - hedonic house price function derived directly from the Rosen's theory of hedonic pricing
  - however, the theory says little on what are the relevant attributes of the house.
- The simple inclusion of a single covariate can improve with respect to the *naive* constant only specification.

```
specification2<-lm(price~bedrooms,data=train)
test$specification2<-predict(specification2,newdata = test)
with(test,mean((price-specification2)^2))
```

```
## [1] 22490147170
```

Example: Predicting House Prices in R

- What about if we include more variables?

```
specification3<-lm(price~bedrooms+bathrooms+centair+fireplace+brick,data=train)
test$specification3<-predict(specification3,newdata = test)
with(test,mean((price-specification3)^2))
```

```
## [1] 21982836467
```

- Note that the *MSE* is once more reduced. If we include all?

Example: Predicting House Prices in R

```
specification4<-lm(price~bedrooms+bathrooms+centair+fireplace+brick+
                lnland+lnbldg+rooms+garage1+garage2+dcbd+rr+
                yrbuilt+factor(carea)+latitude+longitude,data=train)
test$specification4<-predict(specification4,newdata = test)
with(test,mean((price-specification4)^2))
```

```
## [1] 20565890598
```

- Then the MSE for specification 3 goes from 2.1982836×10^{10} to 2.0565891×10^{10}. In this case the MSE keeps improving. Is there a limit to this improvement?
- Is there a limit to this improvement? Can we keep adding features and complexity?

Example: Predicting House Prices in R

- Is there a limit to this improvement?
- Can we keep adding features and complexity?
- Let's try an extreme complex specification

```
specification5<-lm(price~poly(bedrooms,2):poly(bathrooms,3):centair:fireplace:brick
                  :lnland:lnbldg+garage1+garage2+rr+
                  yrbuilt+factor(carea)+poly(latitude,8):poly(longitude,8),data=train)
```

Example: Predicting House Prices in R

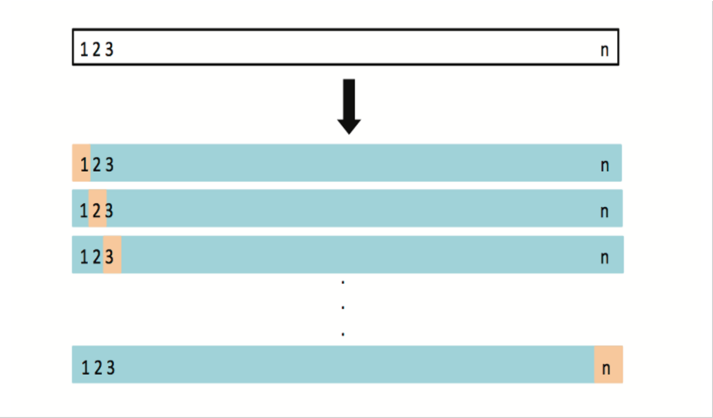
Specification	MSE
1	2.281E+10
2	2.249E+10
3	2.198E+10
4	2.057E+10
5	2.094E+10

Enfoque de conjunto de validación

- Ventajas:
  - Simple
  - Fácil de implementar
- Desventajas:
  - El MSE de validación (prueba) puede ser altamente variable
  - Solo se utiliza un subconjunto de observaciones para ajustar el specificationo (datos de entrenamiento). Los métodos estadísticos tienden a funcionar peor cuando se entrenan con pocas observaciones.

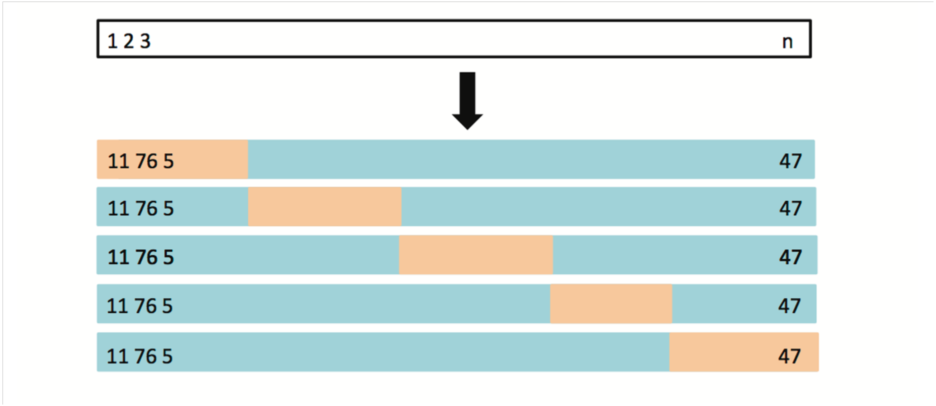
Leave-One-Out Cross Validation (LOOCV)

- Este método es similar al enfoque de validación, pero trata de abordar las desventajas de este último.



Validación cruzada en K-partes

- LOOCV es computacionalmente intensivo, por lo que podemos ejecutar k-fold Cross Validation



Validación cruzada en K-partes

- Dividir los datos en K partes $(N = \sum_{j=1}^{K} n_j)$
- Ajustar el modelo dejando afuera una de las partes (folds) $\rightarrow f_{-k}(x)$
- Calcular el error de predicción en la parte (fold) que dejamos afuera

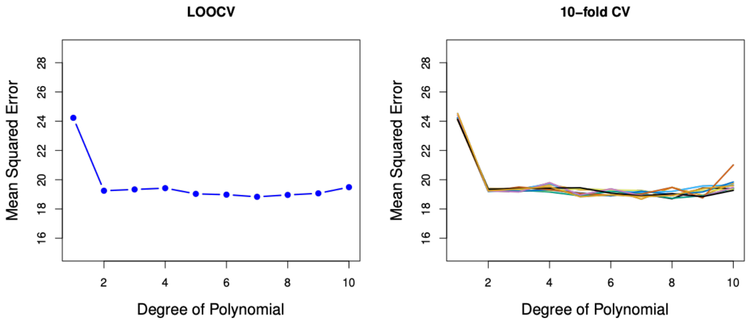
$$MSE_j = \frac{1}{n_j} \sum (y_j^k - \hat{y}_{-j})^2 \tag{32}$$

- Promediar

$$CV_{(k)} = \frac{1}{k} \sum_{j=1}^{k} MSE_j \tag{33}$$

Validación cruzada en K-partes

- Izquierda: LOOCV error
- Derecha: 10-fold CV
- LOOCV es caso especial de k-fold, donde k = n
- Ambos son estables, pero LOOCV (generalmente) es mas intensivo computacionalmente!



Trade-off Sesgo-Varianza para validación cruzada en K-partes

- Sesgo:
  - El enfoque del conjunto de validación tiende a sobreestimar el error de predicción en la muestra de prueba (menos datos, peor ajuste)
  - LOOCV, agrega más datos $\rightarrow$ menos sesgo
  - K-fold un estado intermedio
- Varianza:
  - LOOCV promediamos los resultados de n modelos ajustados, cada uno está entrenado en un conjunto casi idéntico de observaciones $\rightarrow$ altamente correlacionado
  - K partes esta correlación es menor, estamos promediando la salida de k modelo ajustado que están algo menos correlacionados
- Por lo tanto, existe un trade-off
  - Tendemos a usar k-fold CV con ($K = 5$ y $K = 10$)
  - Se ha demostrado empíricamente que producen estimaciones del error de prediccion que no sufren ni de un sesgo excesivamente alto ni de una varianza muy alta Kohavi (1995)

Validation and Cross-validation en la practica

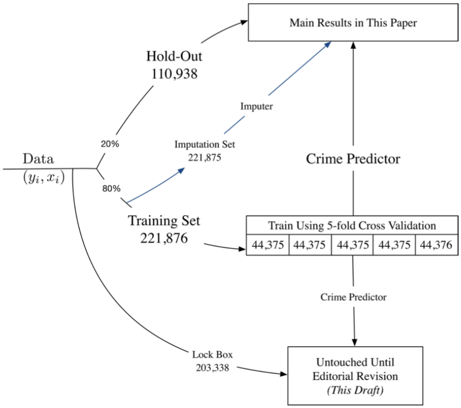


FIGURE I

Partition of New York City Data (2008–13) into Data Sets Used for Prediction and Evaluation

Source: Kleinberg et al (2018)

Example: Predicting House Prices in R

```
library("caret")
```

```
model2 <- train(price ~ bedrooms,
# model to fit
                data = matchdata,
                trControl = trainControl(method = "cv", number = 5),
                method = "lm")
                # fit a simple regression
```

Example: Predicting House Prices in R

```
specification2
```

```
## Linear Regression
##
## 3204 samples
##    1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 2564, 2564, 2563, 2562, 2563
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   147374.5  0.01175485  122163.3
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Example: Predicting House Prices in R

```
specification1<-train(price~.,
# specification to fit
                    data = matchdata,
                    trControl = trainControl(method = "cv", number = 5),
                    # Method: crossvalidation, 5 folds
                    method = "null")
                    # fit a constant only specification
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

A warning will appear since Rsquared cannot be computed. We can again call the results

Example: Predicting House Prices in R

```
specification1
```

```
## Non-Informative Model
##
## 3204 samples
##   19 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 2563, 2564, 2563, 2564, 2562
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   148036.6  NaN       121838.7
```

Example: Predicting House Prices in R

In the same fashion we can `train` the remaining specifications

```
specification3 <- train(price ~ bedrooms+bathrooms+centair+fireplace+brick,
                        data = matchdata,
                        trControl = trainControl(method = "cv", number = 5),
                        method = "lm")


specification4 <- train(price ~ bedrooms+bathrooms+ rooms+centair+fireplace+brick+
                        lnland+lnbldg+garage1+garage2+
                        dcbd+ rr +
                        yrbuilt+ factor(year) +
                        factor(carea)+ latitude+longitude,
                        data = matchdata,
                        trControl = trainControl(method = "cv", number = 5),
                        method = "lm")
```

Example: Predicting House Prices in R

In the same fashion we can `train` the remaining specifications

```
model5 <- train(price ~ poly(bedrooms,2):poly(bathrooms,3):centair:fireplace
                :brick:lnland:lnbldg+garage1+garage2+rr+
                yrbuilt+factor(carea)+poly(latitude,8):poly(longitude,8),
                        data = matchdata,
                        trControl = trainControl(method = "cv", number = 5),
                        method = "lm")
```

Example: Predicting House Prices in R

Specification	RMSE
1	148036.55
2	147374.53
3	145570.15
4	74757.77
5	148875.29

Review

Hoy

- Dilema Sesgo/Varianza
- Sobreajuste y Selección de modelos
  - AIC y BIC
  - Enfoque de Validación
  - LOOCV
  - K-fold Cross-Validation (Validación Cruzada)

Further Readings

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
- Koenker, R. (2013) Economics 508: Lecture 4. Model Selection and Fishing for Significance. Mimeo
- Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., & Mullainathan, S. (2018). Human decisions and machine predictions. The quarterly journal of economics, 133(1), 237-293.
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In Ijcai (Vol. 14, No. 2, pp. 1137-1145)
- Murphy, K. P. (2012). Machine learning: a probabilistic perspective. MIT press.
- Taddy, M. (2019). Business data science: Combining machine learning and economics to optimize, automate, and accelerate business decisions. McGraw Hill Professional.