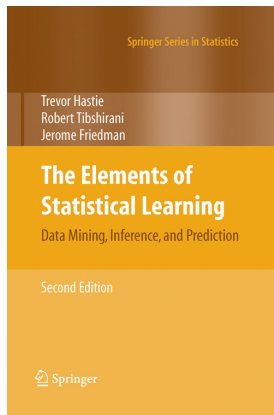
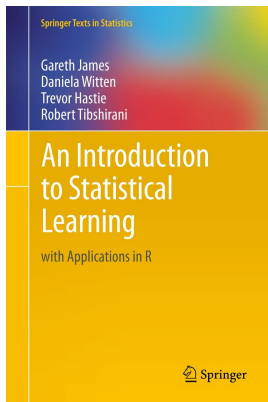


BigData ML and Econometrics

Conceptos Generales sobre Aprendizaje Supervisado



Bibliografía sugerida



ISL: 1, 2 y 5.

ESL: 1, 2.1–2.3 y 7.1–7.3.

Aprendizaje Automático

- Aprendizaje Automático: Encontrar patrones regulares en los datos por medio de algoritmos (modelos estadísticos sofisticados).
- Existen dos grandes paradigmas de aprendizaje con datos:
 - ▶ **Supervisado**: Hay una variable objetivo Y que supervisa el aprendizaje (una variable que nos interesa predecir).
 - ★ Ejemplos: Modelos de regresión y clasificación.
 - ▶ **No supervisado**: No hay una variable a predecir Y ; pero nos gustaría entender ciertos patrones generales que se presentan en los datos.
 - ★ Clustering (estimación de densidad), Componentes Principales, Escalado Multidimensional, Reglas de asociación, etc.

... nos vamos a enfocar en aprendizaje supervisado.

Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
- 3 Caso de Estudio en R
- 4 Apéndice (algunos detalles no tan importantes)

Aprendizaje supervisado

- Asumimos que los features y la variable a modelizar están relacionados a través de alguna función (desconocida) $f(X)$:

$$Y = \underbrace{f(X)}_{\text{modelable}} + \underbrace{\varepsilon}_{\text{no modelable}}$$

- $f(X) \equiv E(Y|X)$ da cuenta de la influencia sistemática de las distintas covariables X (features) sobre la variable de respuesta Y .
- ε representa todos aquellos factores aleatorios no modelables.
 - ▶ En general asumiremos que: $E(\varepsilon) = \text{Cov}(\varepsilon, X) = 0$.
- Objetivo: Aprender (estimar) $f(X)$ con datos:
 - ▶ $S_n : \{(x_1, y_1), \dots, (x_n, y_n)\}$ sampling de $(X, Y) \sim P_{X,Y}(x, y)$.
 - ▶ Estimamos f para hacer predicciones: $x_{\text{new}} \rightarrow \hat{f}(x_{\text{new}}) \equiv \hat{y}$.

¿Porqué estimar $f(X)$?

- Discutimos el caso en que $Y \in \mathbb{R}$.
- Para predecir $Y|X = x_0$ (Y es una v.a. condicional); resolvemos:

$$\min_{\hat{c}} \underbrace{E[(Y - \hat{c})^2 | X = x_0]}_{\text{ECM}(\hat{c}, Y|X=x_0)}.$$

- Como $\text{ECM}(\hat{c}, Y|X = x_0) = \hat{c}^2 - 2\hat{c}E(Y|X = x_0) + E(Y^2|X = x_0)$, luego:

$$\hat{c}^* = E(Y|X = x_0).$$

- **Remark:** Si quiero predecir la variable aleatoria (condicional) $Y|X = x_0$, la regla de predicción que minimiza el ECM (i.e. óptima) se corresponde con la media (condicional) de la variable de respuesta.
- En otras palabras: Si conozco $f(X) \equiv E(Y|X)$, para predecir un valor “futuro” de $Y|X = x_0$ la mejor regla de predicción es $\hat{Y} = f(x_0)$.

$Y \in \{0, 1\}$ (clasificación binaria)

- Notar que en este caso: $E(Y|X = x_0) = P(Y = 1|X = x_0) = f(x_0)$.
- Definamos la pérdida como $L(Y, c) = \mathbb{1}(Y \neq c)$, luego:

$$E\{L_{x_0}(Y, c)\} = 1 - P(Y = c|X = x_0)$$

- Al resolver $\min_c E\{L_{x_0}(Y, c)\}$ surge que $c^* = 1$ cuando se cumple que $P(Y = 1|X = x_0) \geq P(Y = 0|X = x_0)$ y $c^* = 0$ en otro caso.
- En otras palabras: Si conozco $f(X) \equiv E(Y|X)$, para predecir un valor “futuro” de $Y|X = x_0$ la regla de predicción surge de analizar $f(x_0)$:

$c^* = 1$ si $f(x_0) = P(Y = 1|X = x_0) \geq 0.5$ y $c^* = 0$ en otro caso.

- En función de la naturaleza de Y , se suele distinguir:

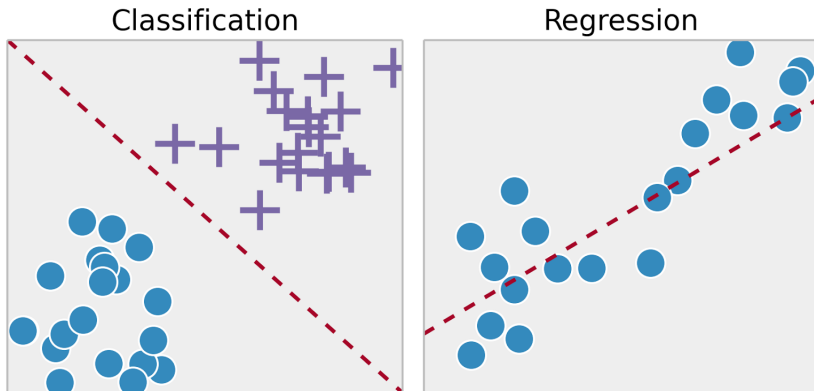


Figure: Problemas de aprendizaje *supervisados* en ML.

- Notar que en ambos casos nos interesa modelar $E(Y|X)$.

- El *modelo* dependerá de ciertos parámetros:

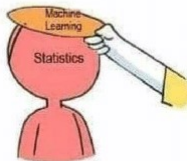
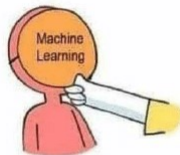
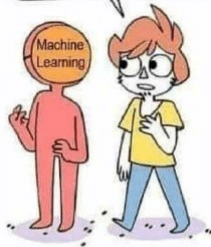
- ▶ Ejemplo: Modelo de regresión lineal

$$Y = \underbrace{\theta_0 + \theta_1 X}_{\text{Modelo } f(X; \theta)} + \varepsilon.$$

- Notar que $f(X; \theta)$ es un modelo para $f(X)$.
- Aprender f es equivalente a estimar los parámetros θ del modelo.

Machine Learning	Estadística/Econometría
Features / Inputs	Regresores / covariables
Outputs / targets	Variable dependiente / de respuesta.
Algoritmo	Modelo
Entrenamiento, Aprendizaje	Estimación
Pesos	Parámetros
Minimizar riesgo	Maximizar verosimilitud
Aprendizaje supervisado	Regresión (modelos aditivos generalizados)
Aprendizaje no supervisado	Estimación de densidad
Predicción	Inferencia

Artificial
Intelligence
HEY WHY
DO YOU ALWAYS
WEAR THAT MASK?



Minimización del riesgo empírico

- Para un modelo $f(\mathbf{x}, \theta)$ y con los datos $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ (de “Train”), donde $\mathbf{x}_i \in \mathbb{R}^p$, aprendemos θ **minimizando el riesgo empírico**:

$$\text{RE}(\theta, \text{Datos}) = \frac{1}{n} \sum_{i=1}^n L(y_i, f(\mathbf{x}_i, \theta)),$$

donde L es una “función de pérdida adecuada”¹.

- ▶ Intentamos **evitar soluciones *miopes*** (queremos predecir!).
- ▶ No existen soluciones cerradas para el problema de minimización.
- **Ingredientes del ML: Datos + Modelo + Optimización.**
- **Predicciones con el modelo aprendido²:**

$$\text{Para cada } x_{\text{new}} \implies \hat{y}_{\text{new}} = f(x_{\text{new}}; \hat{\theta})$$

¹Ejemplo en regresión: $L(y_i, f(\mathbf{x}_i, \theta)) = (y_i - \theta_0 - \theta_1 x_{1i} - \dots - \theta_p x_{pi})^2$.

²En regresión: $\hat{y}_{\text{new}} = \hat{\theta}_0 + \hat{\theta}_1 x_{1\text{new}} + \dots + \hat{\theta}_p x_{p\text{new}}$.

- En este curso (introductorio) vamos a utilizar librerías de R cuando aprendamos los parámetros de un modelo con datos...
- ... aunque no tengamos que resolver por cuenta propia problemas de optimización numérica, parece oportuno hacer aquí un paréntesis para discutir algunos aspectos básicos sobre las técnicas de optimización numérica habitualmente utilizadas en los algoritmos de ML.
- Esto te va a permitir comprender mejor algunos aspectos computacionales fundamentales de cada modelo (ej: entender porqué fitear un modelo de red neuronal es una tarea compleja y que estrategias utilizar para poder aprender los parámetros de éstos modelos en una cantidad razonable de tiempo).

... cambiamos de slides durante unos minutos.

Sobre la calidad predictiva de un modelo...

- En el curso de Análisis Estadístico estudiaste que el error cuadrático medio de un estimador $\hat{\theta}$ (del parámetro θ) se puede descomponer:

$$\text{ECM}(\hat{\theta}, \theta) = E[(\hat{\theta} - \theta)^2] = \underbrace{[E(\hat{\theta}) - \theta]^2}_{\text{Bias}^2(\hat{\theta})} + \text{Var}(\hat{\theta}).$$

- En este curso usaremos como marco de referencia una descomposición parecida, sólo que aquí nos interesa calcular³:

$$\text{ECM}(\hat{Y}, Y) = E[(\hat{Y} - Y)^2] = E[(\hat{Y} - f(X) - \varepsilon)^2].$$

- Teniendo en cuenta que $\hat{Y} \equiv f(X; \hat{\theta})$, se puede construir una descomposición del error cuadrático medio de predicción similar a la que viste en el curso anterior (en el apéndice hay más detalles).

³Deberíamos llamarlo ECMP = Error cuadrático medio de predicción. < > ≡ ≡ ≡

Bias variance trade off (simplificado)

- **Complejidad** \propto **cantidad de parámetros** en el modelo.

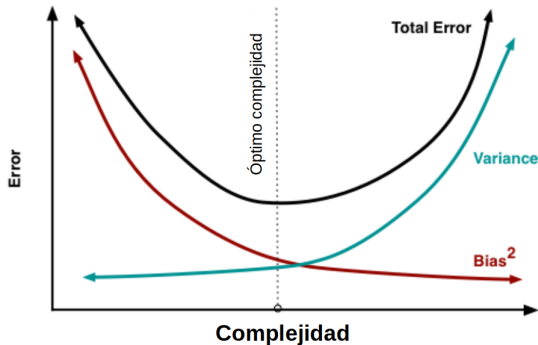
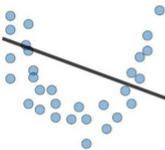
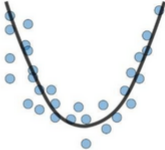

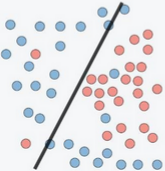
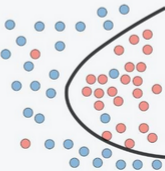
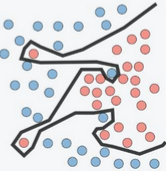


Figure: “Error Esperado Modelo” $\equiv \text{Bias}^2(f(X; \hat{\theta})) + \text{Var}(f(X; \hat{\theta})) + \sigma_{\epsilon}^2$.

- **Bias**: Distancia entre $E(f(X; \hat{\theta}))$ y $f(X)$.
- **Var**: Refleja el cambio en el modelo al perturbar los datos.
- σ_{ϵ}^2 : Cota inferior del error esperado (varianza de ϵ).

Bias vs variance en la práctica

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none">• High training error• Training error close to test error• High bias	<ul style="list-style-type: none">• Training error slightly lower than test error	<ul style="list-style-type: none">• Very low training error• Training error much lower than test error• High variance
Regression illustration			
Classification illustration			

- No free-lunch theorem.

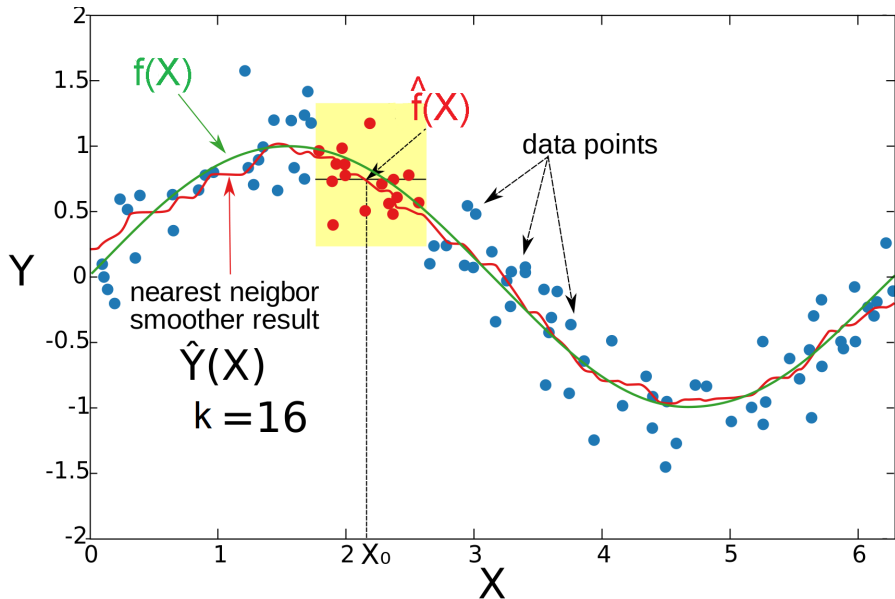
Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
 - Modelo de K-vecinos
 - Selección y evaluación de modelos
 - Estrategias de validación cruzada
- 3 Caso de Estudio en R
- 4 Apéndice (algunos detalles no tan importantes)

Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
 - Modelo de K-vecinos
 - Selección y evaluación de modelos
 - Estrategias de validación cruzada
- 3 Caso de Estudio en R
- 4 Apéndice (algunos detalles no tan importantes)

Regresión (intuición)



El modelo de regresión de los k-vecinos

- Dada una muestra $S_n : \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ y $k \in \mathbb{N}$.
- Nota: $\mathbf{x} \in \mathbb{R}^p$ es un vector con p features continuos.
- Para cualquier \mathbf{x} , llamemos $d_k(\mathbf{x}, S_n) \equiv d_k(\mathbf{x})$ a la distancia (Euclidiana) desde \mathbf{x} hasta su k vecino más cercano de entre los elementos de S_n ; por ejemplo: $d_1(\mathbf{x}) = \min_{i=1, \dots, n} \|\mathbf{x}_i - \mathbf{x}\|$.
- Definimos $N_k(\mathbf{x}) = \{j \in \{1, \dots, n\} \mid \|\mathbf{x}_j - \mathbf{x}\| \leq d_k(\mathbf{x})\}$, luego

$$\hat{f}(\mathbf{x}; S_n, k) \equiv \hat{f}_k(\mathbf{x}) = \frac{1}{|N_k(\mathbf{x})|} \sum_{i \in N_k(\mathbf{x})} y_i$$

- El algoritmo estima $f(X)$ en el punto $X = \mathbf{x}$, haciendo un promedio *local* (basado en los k -vecinos de $X = \mathbf{x}$) de las Y 's.
- La distancia Euclidiana es sensible a la escala: Se recomienda estandarizar los features cuando estén en escalas diferentes.

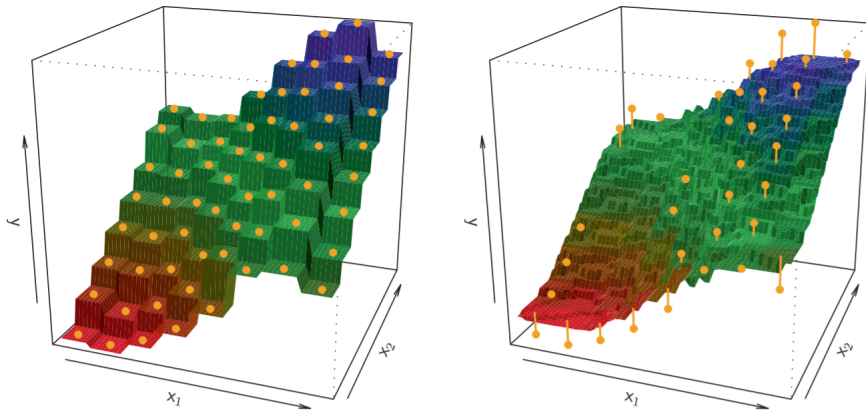


FIGURE 3.16. Plots of $\hat{f}(X)$ using KNN regression on a two-dimensional data set with 64 observations (orange dots). Left: $K = 1$ results in a rough step function fit. Right: $K = 9$ produces a much smoother fit.

- Ilustramos como fitear este modelo en R y como afecta el valor de k al output del mismo con una pequeña simulación.

Simulemos datos de un problema de regresión donde:

$$Y = \underbrace{\sin(2X) + \sqrt{X}}_{f(X)} + \varepsilon, \text{ con } X \in [0, 4\pi] \text{ y } \varepsilon \sim N(0, 1).$$

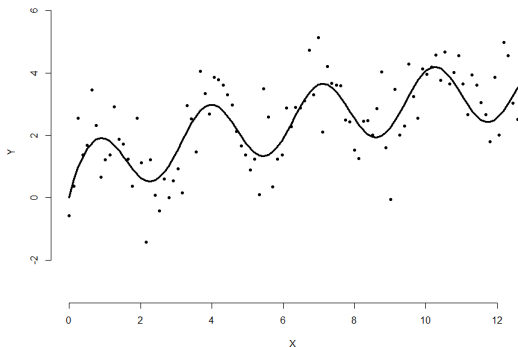


Figure: Muestra de entrenamiento de tamaño $n = 100$ y (la “verdadera”) $f(x)$.

- Exploremos las líneas del código *KNN_reg.R*.

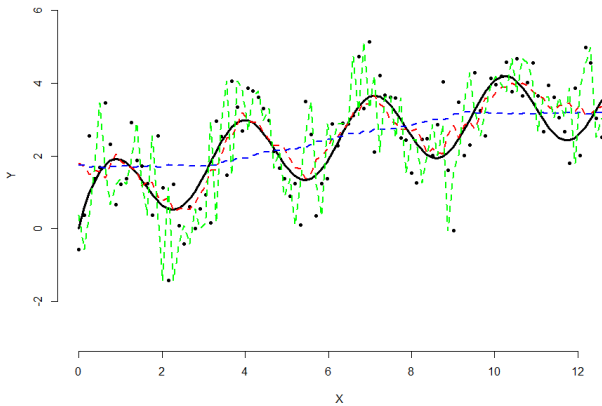


Figure: Estimaciones de f para: $k = 1$ (—), $k = 10$ (—) y $k = 50$ (—).

- ¿Qué valor de k es más adecuado?
- ¿Cuántos parámetros tiene el modelo? (under-vs-over fitting).
- ¿Cómo elegirías k en contextos generales? (validación cruzada).

Clasificación (intuición)

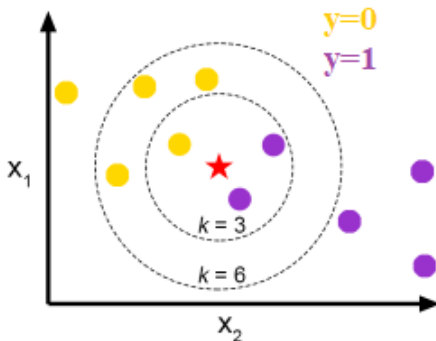


Figure: Clasificación ($Y \in \{0, 1\}$) con $k = \{3, 6\}$ vecinos.

$$\hat{f}_k(\mathbf{x}) = \frac{1}{k} \sum_{i \in N_k(\mathbf{x})} y_i \text{ ó } \hat{f}_k(\mathbf{x}) = \text{sign} \left(\sum_{i \in N_k(\mathbf{x})} y_i / k - 1/2 \right).$$

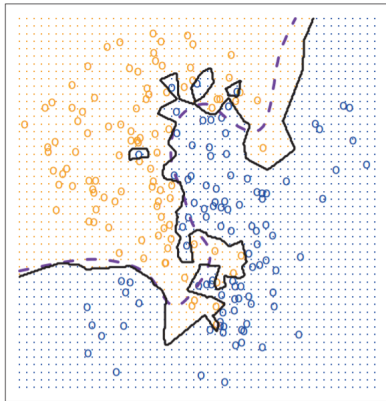
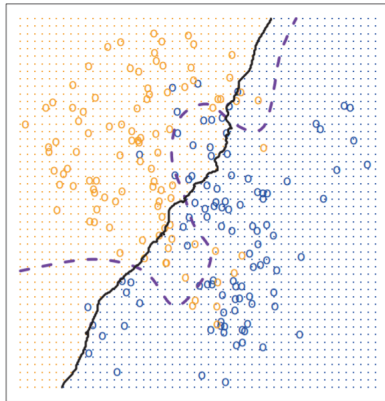
KNN: $K=1$ KNN: $K=100$ 

Figure: Valores pequeños de k pueden causar overfitting, y grandes underfitting.

- Mismo racional que en los problemas de regresión.

Comentarios finales

- Estandariza las covariables!
- La métrica utilizada para determinar relaciones de proximidad:

$$\|\mathbf{x}_1 - \mathbf{x}_2\|_q = \left(\sum_{i=1}^p |x_{1i} - x_{2i}|^q \right)^{1/q}, \quad q > 0,$$

es poco relevante; lo importante es como elijas k .

- The Curse of Dimensionality: Cuando $p \gg 0$ independientemente de la métrica que utilices y de como elijas k , las distancias entre los datos se parecerán mucho y las relaciones de proximidad serán poco útiles para determinar una estimación "local" de Y (reg. y clas.).
- ¿Variables categóricas en el modelo?
 - ▶ ¿One Hot Encoding? (representación binaria).
 - ▶ ¿Distancias entre categorías?

Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
 - Modelo de K-vecinos
 - Selección y evaluación de modelos
 - Estrategias de validación cruzada
- 3 Caso de Estudio en R
- 4 Apéndice (algunos detalles no tan importantes)

- **Selección de modelo:** Estimar/definir el valor adecuado de ciertos “hiperparámetros*” sensibles del modelo (por ejemplo el valor “k”).

Train set: Datos para aprender/estimar parámetros del modelo.

Validation set: Datos para validar ciertos hiperparámetros.

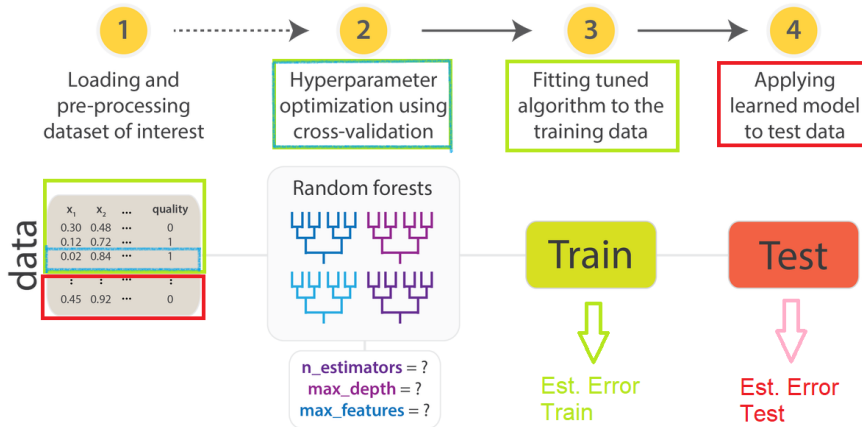
- **Evaluación del modelo:** Estimar la calidad predictiva (fuera de la muestra) ya que nos interesa hacer predicciones con el modelo.

Test set.

- Discutimos el *flow* habitual y luego repasamos las 3 estrategias de validación cruzada más utilizadas para hacer selección de modelos.

*Esta manera de referirse a ciertos meta-parámetros del modelo es confusa (también se utiliza en estadística Bayesiana); pero es parte del léxico común de los científicos de datos y por ello utilizaremos este término con frecuencia.

Flow habitual



- Datos = (Train + Validación) + Test.
- Revisar y depurar los datos antes de empezar a modelar.

- Train + Validación para hacer *selección de modelo* (ej: determinar k).
 - ▶ Diferentes estrategias de *validación cruzada*.
- Finalizada la “selección de modelo”, se estima la calidad predictiva que tendría el modelo seleccionado utilizando los datos en test.
- No es factible dar una regla general sobre el tamaño de los conjuntos de TRAIN–VALIDACIÓN y TEST. En cualquier caso, siempre se recomienda hacer asignaciones aleatoria de observaciones a cada uno de a éstos subconjuntos.

Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
 - Modelo de K-vecinos
 - Selección y evaluación de modelos
 - Estrategias de validación cruzada
- 3 Caso de Estudio en R
- 4 Apéndice (algunos detalles no tan importantes)

Validation set approach

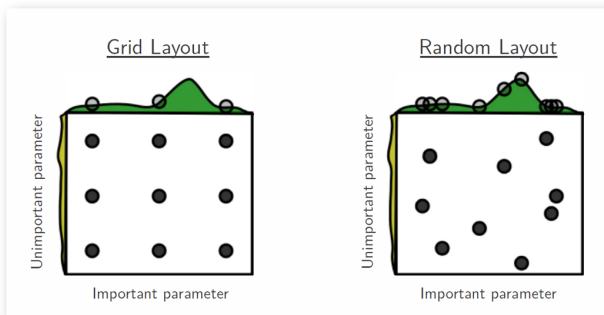
- Modelo $f(x; \theta, \alpha)$: θ indica el conjunto de parámetros propios del modelo y α representa el conjunto de hiperparámetros (Ejem: “k”).
- Definimos una *mall*a $\mathcal{M} = \{\alpha_1, \dots, \alpha_m\}$ de valores razonable para α .
- Separamos en train y validación (típicamente 50%-50%).



FIGURE 5.1. A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.

- Para cada $\alpha \in \mathcal{M}$, con TRAIN aprendemos $\hat{\theta}_\alpha$ y con VALIDACIÓN estimamos (puntualmente) $\text{MSE}(\alpha)$ o $\text{TE}(\alpha)$.

- Algunas estrategias habituales para construir $\mathcal{M} = \{\alpha_1, \dots, \alpha_m\}$



- Una vez terminamos de estimar

$$\widehat{\text{MSE}}(\alpha) = \frac{1}{n_{\text{val}}} \sum_{i=1}^{n_{\text{val}}} (y_i - f(x_i; \hat{\theta}_{\alpha}))^2 \text{ para cada } \alpha \in \mathcal{M},$$

elegimos $\alpha^* = \operatorname{argmin}_{\alpha \in \mathcal{M}} \{\widehat{\text{MSE}}(\alpha_1), \dots, \widehat{\text{MSE}}(\alpha_m)\}.$

- **Pros:**

- ▶ Intuitivo y por tanto fácil de *comunicar*.
- ▶ Fácil de implementar.
- ▶ Veloz en comparación con otros métodos ($O(m)$).

- **Contras:**

- ▶ ¿Error estándar de la estimación del MSE/TE al seleccionar modelo?
 - ▶ Pérdida de eficiencia: El tamaño muestral que utilizo para estimar los parámetros del modelo suele ser bastante menor que n .
 - ▶ La elección de α^* (los hiperparámetros del modelo en general) puede ser sensible respecto a como divido los datos entre train y validación.
- Estos puntos negativos se pueden mitigar con los métodos que presentamos a continuación, a cambio de mayor costo computacional.
 - En lo que sigue consideramos α fijo (repetir para cada valor de α dentro de una malla de valores posibles para el hiperparámetro).

Leave-one-out cross-validation (LOOCV)

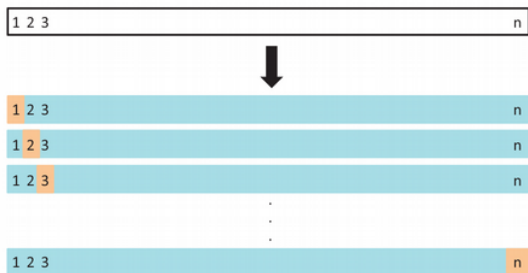


FIGURE 5.3. A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.

- Lamemos $\hat{f}_{-i}^{(\alpha)}$ a la predicción sobre el dato i -ésimo del modelo estimado sin contar con dicho dato y $\widehat{\text{MSE}}(\alpha) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}_{-i}^{(\alpha)})^2$.
- Elegimos $\alpha^* \equiv \arg \min_{\alpha \in \mathcal{M}} \{\widehat{\text{MSE}}(\alpha_1), \dots, \widehat{\text{MSE}}(\alpha_m)\}$.

LOOCV: Pros y contras

- Pros:

- ▶ Cada modelo es estimado con $n - 1$ (en vez de $\approx n/2$) observaciones.
- ▶ No hay “aleatoriedad”: Si repito el procedimiento nunca cambian las estimaciones de la tasa de error ni mi elección de α (¿porqué?).

- Contras:

- ▶ Computacionalmente intensivo! ($O(nm)$)

- Estrategia intermedia: B-fold cross validation.

B-fold cross-validation (típicamente $B = 5$ ó 10)

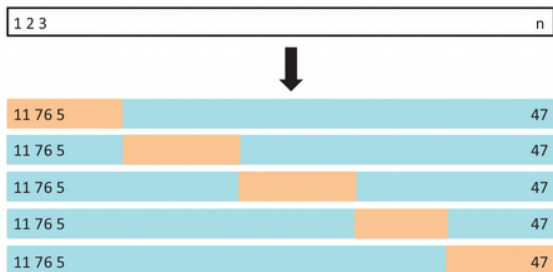


FIGURE 5.5. A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.

- Lamemos $\hat{f}_{-b}^{(\alpha)}$ al modelo estimado sin contar con los datos del fold F_b , luego: $\widehat{\text{MSE}}_b(\alpha) = \frac{1}{n_b} \sum_{i \in F_b} (y_i - \hat{f}_{-b}^{(\alpha)}(x_i))^2$. Luego se tiene que:

$$\widehat{\text{MSE}}(\alpha) = \frac{1}{B} \sum_{b=1}^B \widehat{\text{MSE}}_b(\alpha).$$

- **Pros:**

- ▶ Computacionalmente menos costoso que LOOCV ($O(Bm)$).
- ▶ Podemos estimar el error estándar en la estimación de \widehat{MSE} :

$$se(\widehat{MSE}_\alpha) = \sqrt{\frac{1}{B-1} \sum_{b=1}^B (\widehat{MSE}_b(\alpha) - \widehat{MSE}(\alpha))^2}$$

- **Contras:**

- ▶ Aleatoriedad en el método; pero más robusto que validation-set.

- **Fact:** Es el método más utilizado en la práctica para para aprender valores razonables para los hiperparámetros de un modelo.

Veamos como implementar estos métodos en R!

Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
- 3 Caso de Estudio en R**
- 4 Apéndice (algunos detalles no tan importantes)

Predicción de demanda



Figure: Datos tomados de *capital bikeshare*.

Ping-pong de preguntas:

- Alguien argumenta que LOOCV es un caso especial de B-folds CV, en donde $B = n$. ¿Vos que opinas?
- ¿Podríamos computar el error estándar en las estimaciones del MSE para cada valor del hiperparámetro cuando utilizamos LOOCV?
 - ▶ Utiliza el código que discutimos en clase para computar dicha cantidad cuando ajustas el valor de k con ésta técnica.
- ¿Para qué sirve estimar el error estándar de la estimación del MSE?
- ¿Qué cambios habría que considerar si utilizaras las 3 técnicas anteriores para ajustar el valor del hiperparámetro k en un contexto de clasificación?
- En la sección 5.4.1 de ISLR se discute el trade-off bias-variance en la estimación del MSE cuando utilizamos LOOCV vs B-Fold, resume el punto fundamental de esta discusión con tus propias palabras.

Agenda

- 1 Aprendizaje Supervisado
- 2 Estimando el error del modelo por validación cruzada
- 3 Caso de Estudio en R
- 4 Apéndice (algunos detalles no tan importantes)


Descomposición del ECM de predicción

- En el contexto de un problema de regresión:
 - ▶ $Y_0 \equiv f(x_0) + \varepsilon$.
 - ▶ $f(x_0; \theta) \equiv \hat{f}_{x_0} = \hat{Y}_0$ (**estimador**⁴ de Y cuando $X = x_0$).
- El error cuadrático medio se descompone como (ver slide siguiente):

$$E(Y_0 - \hat{f}_{x_0})^2 = \text{bias}^2(\hat{f}_{x_0}) + \text{var}(\hat{f}_{x_0}) + E(\varepsilon^2).$$

- $\text{bias}^2(\hat{f}_{x_0}) = E(f(x_0; \theta) - f(x_0))^2$.
- $\text{var}(\hat{f}_{x_0}) = E((f(x_0; \theta) - E(f(x_0; \theta)))^2)$.
- Finalmente:

$$\text{Bias}^2(f(X; \theta)) \equiv \int \text{bias}^2(\hat{f}_x) dF_X \text{ y } \text{Var}(f(X; \theta)) \equiv \int \text{var}(\hat{f}_x) dF_X$$

⁴ \hat{f}_{x_0} es una variable aleatoria que toma valores cuando consideramos una muestra de entrenamiento en concreto y estimamos los parámetros del modelo en cuestión ($\theta \equiv \hat{\theta}$). 

(bck-up)

- Sumando y restando la constante $f(x_0)$ tenemos que:

$$\begin{aligned} E\{[(Y_0 - f(x_0)) + (f(x_0) - \hat{f}_{x_0})]^2\} &= E[\overbrace{(Y_0 - f(x_0))^2}^{\varepsilon^2}] + E[(f(x_0) - \hat{f}_{x_0})^2] \\ &= \sigma_\varepsilon^2 + E[(f(x_0) - \hat{f}_{x_0})^2] \end{aligned}$$

- El termino cruzado desaparece bajo los supuestos en ε .
- Ahora volvemos a sumar y restar $E(\hat{f}(x_0))$, y tenemos:

$$E\{[(f(x_0) - E(\hat{f}_{x_0})) + (E(\hat{f}_{x_0}) - \hat{f}_{x_0})]^2\} = \underbrace{(f(x_0) - E(\hat{f}_{x_0}))^2}_{\text{Bias}^2(\hat{f}_{x_0})} + \underbrace{E[(E(\hat{f}_{x_0}) - \hat{f}_{x_0})^2]}_{\text{Var}(\hat{f}_{x_0})}$$

- El termino cruzado desaparece porque $E[(E(\hat{f}_{x_0}) - \hat{f}_{x_0})] \equiv 0$.
- Juntando todas las piezas tenemos la descomposición planteada.