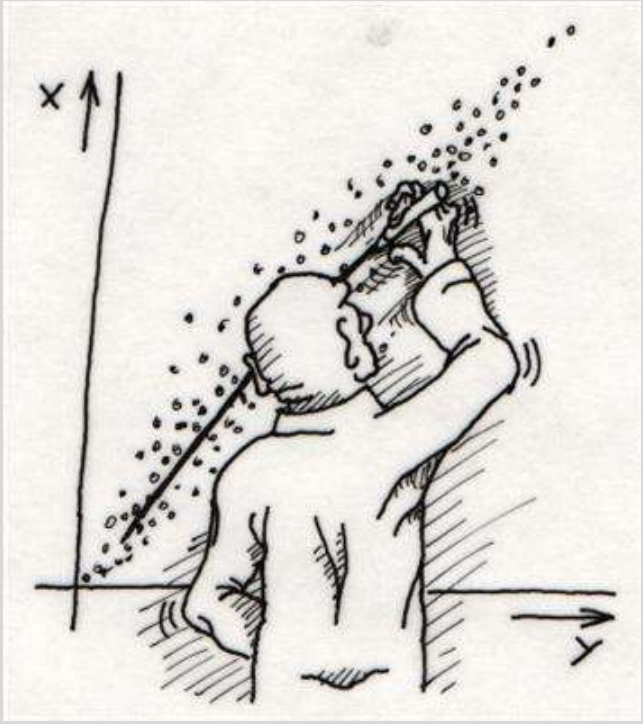


DigitalHouse > Coding School

DATA SCIENCE

MÓDULO 3

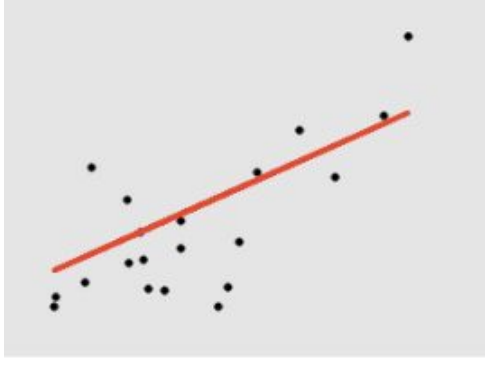
Regularización

OBJETIVOS DE LA CLASE

1. **Entender la regularización como técnica para evitar el sobreajuste**

2. **Aplicar regularización usando scikit-learn**

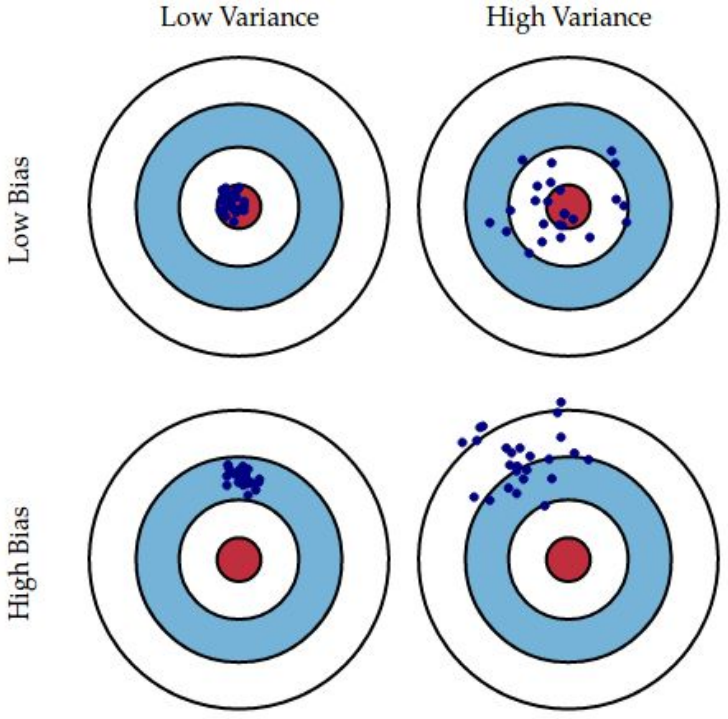
3. **Aprender a hacer validación cruzada para ajustar los hiper-parámetros de regularización**



Recordemos

Sesgo - Varianza

- Podemos caracterizar un modelo según el grado de ajuste a los datos
  - **Varianza alta** → ajuste exagerado ("sobreajuste" o *overfitting)*
  - **Sesgo alto** → ajuste insuficiente ("subajuste" o *underfitting*)



$$y = f(x) + \epsilon = \beta \cdot x + \epsilon$$

Donde epsilon es un término aleatorio con media cero

Buscamos una $\hat{f}$ (*ie.* un vector $\hat{\beta}$) que haga pequeño

$$\underbrace{ECM(\hat{f}(x))}_{\text{Error de generalización}} = E((y - \hat{f}(x))^2)$$

$$= \underbrace{Sesgo(\hat{f}(x))^2 + Var(\hat{f}(x))}_{\text{Trade-off sesgo/varianza}} + \underbrace{\sigma^2}_{\text{Error irreductible}}$$

- Podemos caracterizar un modelo según el grado de ajuste a los datos
  - Si el modelo es demasiado simple (tiene pocos grados de libertad), entonces no importa cuán grande sea la muestra: tenemos sesgo o error sistemático $E(\hat{f}(x)) \neq E(f(x))$
  - Si el modelo es demasiado complejo (ie. tiene demasiados grados de libertad), entonces el estimador puede ajustarse regularidades espurias de la muestra incrementando $Var(\hat{f}(x))$ tenemos sobre-ajuste.
  - Por lo tanto, el modelo no debe ser ni muy simple ni muy complejo

- Si los regresores x_i y x_j están muy correlacionados, pequeñas fluctuaciones en la muestra pueden resultar en amplias variaciones de los coeficientes $\hat{\beta}_i$ y $\hat{\beta}_j$ incrementando $Var(\hat{f}(x))$

- En el extremo, si x_i y x_j son perfectamente colineales, el problema ni siquiera tiene una solución única. .

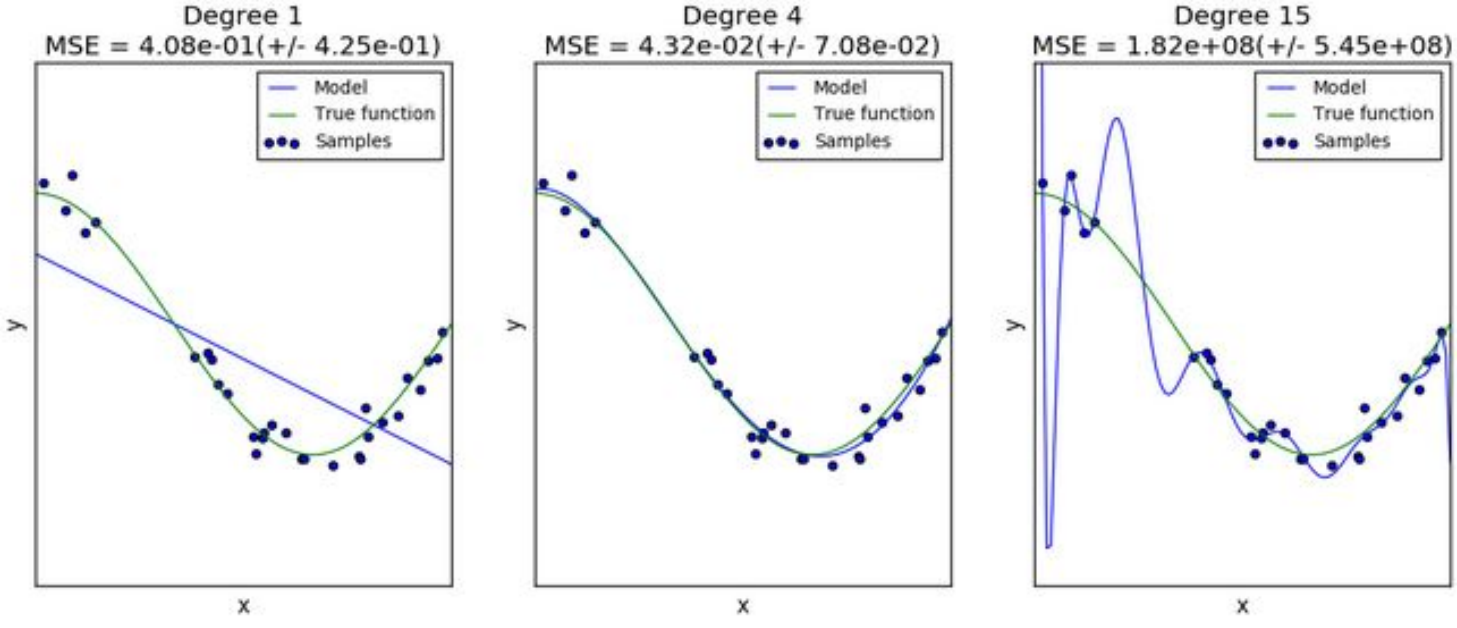
- Querríamos evitar problemas mal condicionados / planteados repartiendo más suavemente los pesos entre los regresores muy correlacionados.

- La regularización produce una familia de problemas relacionados con la minimización de pérdida original a través de un término regulable de contracción (shrinkage).

- La familia está indexada por hiper parámetros (en general uno, que llamaremos λ) que regulan la complejidad del modelo disminuyendo o aumentando el t´ermino de contracción.

- Distintos tipos de regularizaci´on producen soluciones con diferentes caracter´ısticas deseables (parsimoniosas —sparse—, bien condicionadas, etc.).

Por ejemplo


Degree 1
MSE = 4.08e-01(+/- 4.25e-01)
Model
True function
Samples
y
x
Degree 4
MSE = 4.32e-02(+/- 7.08e-02)
Model
True function
Samples
y
x
Degree 15
MSE = 1.82e+08(+/- 5.45e+08)
Model
True function
Samples
y
x

Extendiendo la Regresión Lineal

- Recordemos el modelo lineal $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$

- Nuestro objetivo es **extender la capacidad de este modelo.**

En los modelos de regresión por mínimos cuadrados, resolvemos el problema de la elección de los Betas **minimizando la suma de los residuos al cuadrado**

- A pesar de ser muy simple, el modelo de regresión lineal es
  - **fácil de interpretar**
  - **computacionalmente eficiente.**

¿Podríamos proponer una **nueva forma de ajuste** que mejore la performance del modelo?

Dos razones para buscar alternativas para extender la regresión lineal:

- **Accuracy**: ¿Recuerdan el trade-off entre sesgo y varianza? Cuando la cantidad de datos **n** se acerca a la cantidad de parámetros **p** que tenemos que ajustar, es difícil controlar la varianza de los estimadores y esto hace que caiga la performance en promedio.

- **Interpretabilidad**: Si logramos eliminar los features irrelevantes, la interpretación de los coeficientes del modelo es muy clara y directa. Vamos a ver algunas técnicas automáticas para la selección de features.

Se trata de definir qué variables deberían entrar en un modelo. En regresión lineal las técnicas se dividen en tres grupos.

- Selección de un subset: Buscar los predictores de todo el conjunto que creemos mejor se relacionan con la respuesta.

- Regularización: Ajustamos un modelo con todos los regresores, pero los coeficientes de algunos de ellos se ajustan a cero por las características de la técnica.

- Reducción de dimensiones: Proyectamos los *p* regresores disponibles originalmente en un espacio M de menor dimensión y utilizamos esa proyección como los nuevos regresores.

Se trata de definir qué variables deberían entrar en un modelo. En regresión lineal las técnicas se dividen en tres grupos.

- Selección de [illegible] do el conjunto que creemos me[illegible]

Hoy vamos a estudiar Regularización

- Regularización: Ajustamos un modelo con todos los regresores, pero los coeficientes de algunos de ellos se ajustan a cero por las características de la técnica.

- Reducción de dimensiones: Proyectamos los *p* regresores disponibles originalmente en un espacio M de menor dimensión y utilizamos esa proyección como los nuevos regresores.

Regularización

- Existe una técnica que nos ayuda a buscar el nivel de complejidad óptimo: la **REGULARIZACIÓN**
- Intuitivamente, podemos entender el concepto de regularización en términos del principio de parsimonia (Navaja de Occam).
- Este principio dice: *En igualdad de condiciones, la explicación más sencilla suele ser la más probable*
- En nuestro caso sería:
    - *a misma capacidad de predicción, el modelo **más sencillo es mejor***

- En el modelo de regresión lineal la función de pérdida era la siguiente:

$$CF = \sum_{i}^{N} (\hat{y}_i - y_i)^2$$

- Cuando intentamos utilizar técnicas de regularización, se le agrega una "penalidad" a esa función de costo. La idea es hacer que a mayor complejidad del modelo, mayor sea la cantidad a minimizar.

- La forma general de la función de costo es la siguiente:

$$CF = \sum_{i}^{N} (\hat{y}_i - y_i)^2 + \alpha\theta_i$$

- Aquí theta es el vector que corresponde a los parámetros del modelo (en una regresión lineal, los betas) y alpha es un parámetro que "regula" la fuerza de la penalización: cuanto más grande es, mayor es la penalización.

Vamos a ver a continuación dos técnicas de regularización: **Regresión Ridge** y **Regresión Lasso**.

Estas técnicas proponen cambiar ligeramente el problema de optimización de mínimos cuadrados, para intentar "achicar" (*shrink*) el valor absoluto de los estimadores de los Betas.

¿Por qué esto mejoraría la estimación? Vamos a ver de qué forma este método introduce un sesgo pero reduce la varianza.

- Recordemos la función que se minimiza en la estimación de mínimos cuadrados:

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

- Esta es, en cambio la función que se minimiza en Regresión Ridge:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2,$$

La diferencia es que agregamos un término nuevo. En este término, un **hiperparámetro lambda** penaliza el valor de los coeficientes al cuadrado.
Entonces, tengo que minimizar el cuadrado de los errores, intentando que ningún β_j^2 sea demasiado grande

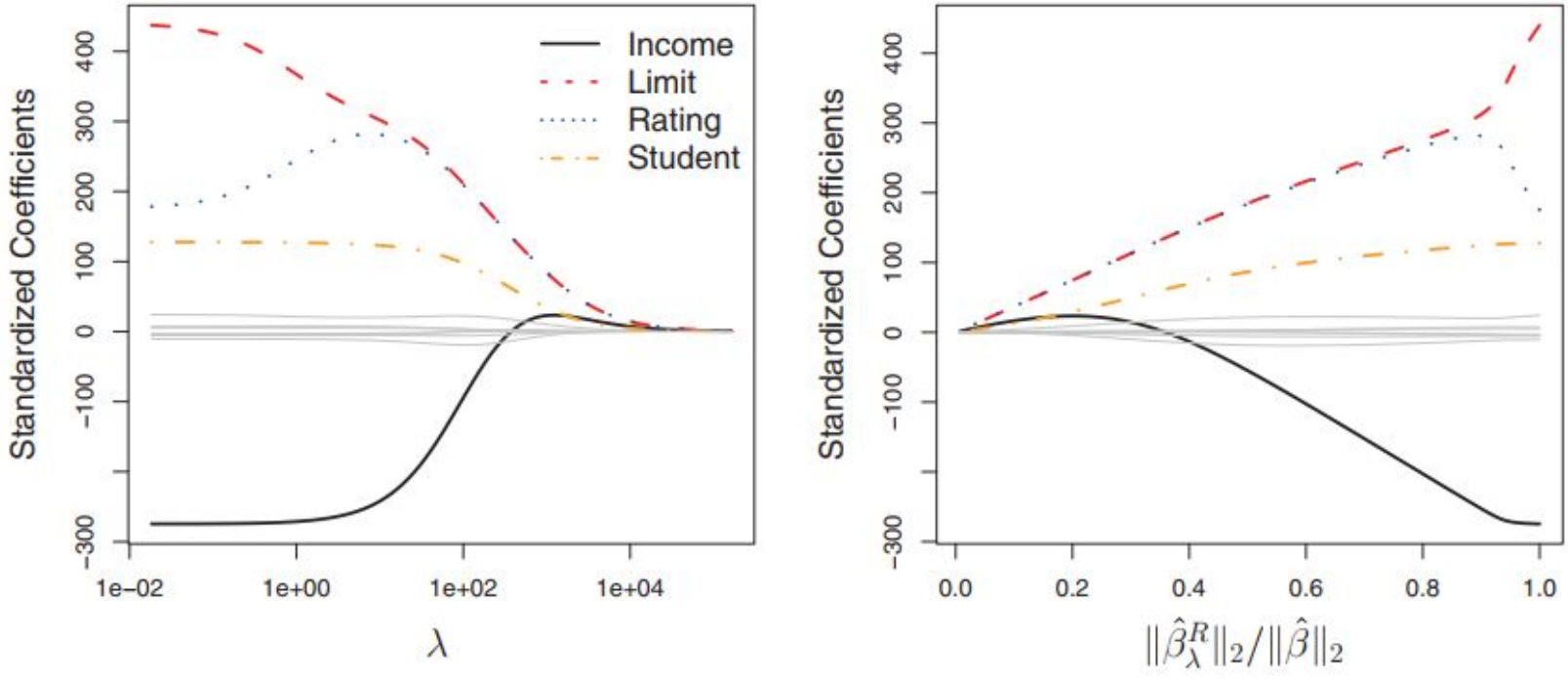
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

- Al igual que MCO, buscamos achicar el RSS.

- Sin embargo, existe un **término de penalización**, que es menor cuando los Betas se acercan a cero, por lo tanto tiene el efecto de achicar los mismos hacia cero (tanto si son negativos como positivos)

- El **hiperparámetro lambda**, maneja la ponderación de cada término.

- ¿Cuál es el mejor valor para lambda? ¿Cómo elegíamos el valor óptimo de un hiperparámetro?
  Como siempre, lo hacemos a través de **CROSS VALIDATION**


Income
Limit
Rating
Student
Standardized Coefficients
400
300
200
100
0
−100
−300
1e−02
1e+00
1e+02
1e+04
λ
Standardized Coefficients
400
300
200
100
0
−100
−300
0.0
0.2
0.4
0.6
0.8
1.0
$\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$

- En la figura de la izquierda, cada curva corresponde a los parámetros estimados de los coeficientes de la regresión Ridge, a medida que aumenta el λ.
- En el panel de la derecha, vemos la relación entre los coeficientes de la regresión Ridge y los de la regresión múltiple tradicional.

  La métrica que calculamos para cada vector de coeficientes es lo que se denomina la "norma", que nos da una idea del tamaño en valor absoluto de cada uno de los componentes, amplificando el efecto de los que son más grandes.

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j{}^2}.$$

A medida que **aumenta λ, los coeficientes de la regresión Ridge se hacen más chicos** con respecto a los de la regresión de Mínimos Cuadrados Ordinarios

- ¿Recuerdan que los coeficientes de la regresión tradicional no eran sensibles a la escala? La predicción del modelo no cambia si los valores están expresados en metros o en centímetros, en grados Fahrenheit o grados Celsius.

Las predicciones de la regresión lineal no se veían afectadas por un cambio de escala porque los coeficientes tenían la capacidad de dar cuenta de este dato.

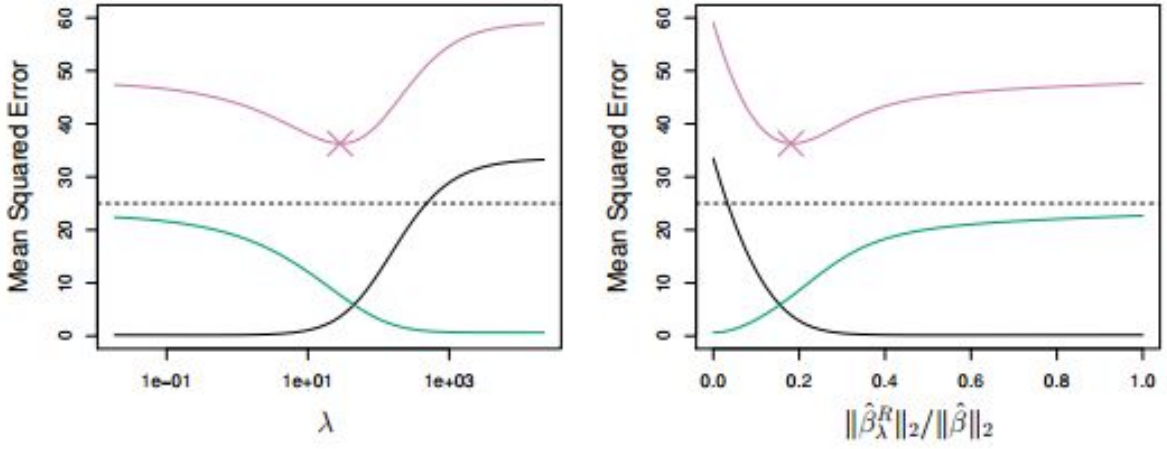
- En Regresión **Ridge**, en cambio, tanto la estimación de los coeficientes como la predicción son **sensibles a la escala**.
- Recordemos el problema de optimización que resuelve Ridge:

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

- Si una variable se encuentra en una escala que le da un valor absoluto mayor, esto **va a afectar el cálculo de la suma de cuadrados** del vector de coeficientes.
- Por esta razón **es importante estandarizar(dividir por el desvío estándar)** todos los regresores antes de ejecutar una regresión Ridge. Así ya no están en unidades físicas sino en unidades de su propio desvío estándar.

$$\tilde{x}_{ij}=\frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\overline{x}_j)^2}}$$

El tradeoff entre Bias-Variance



Aquí se pueden ver n=50 simulaciones, p=45 predictores, todos con coeficientes no nulos.

El gráfico expresa el sesgo cuadrado (negro), varianza (verde) y el MSE del test (violeta), para una regresión ridge en los datos simulados, como una función d $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

La línea punteada, indica el MSE mínimo

- La regresión Ridge tiene una clara desventaja: incluye todos los predictores p en el modelo final, a diferencia de aquellos modelos que eligen un conjunto de variables.
- La regresión Lasso es un alternativa relativamente nueva a Ridge, que corrige esta desventaja. Los coeficientes $\hat{\beta}_\lambda^L$, minimizan el número de variables

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS} + \lambda\sum_{j=1}^{p}|\beta_j|.$$

- Lasso utiliza penaliza con l1, y no con l2. La norma de l1 de un un vector de coeficientes β está dada por $\|\beta\|_1 = \sum|\beta_j|$.

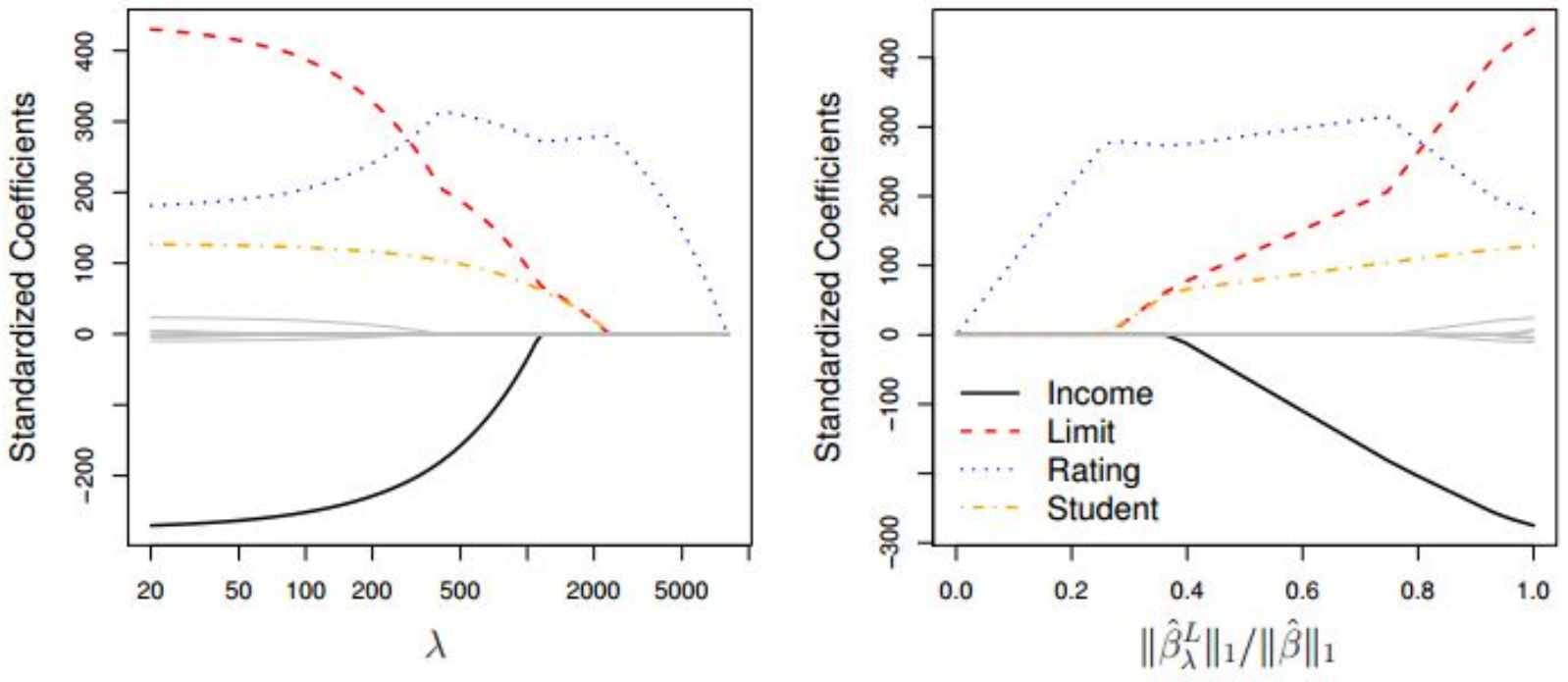
- Como en la regresión ridge, lasso “achica” los coeficiente estimados hacia el zero.

- Sin embargo, en el caso de Lasso, el l1 fuerza los coeficientes a valer exactamente cero, en el caso de que λ sea lo suficientemente grande.

- Por lo tanto, como en la selección de subsets, el lasso n selecciona variables

- Entonces, decimos que Lasso genera modelos dispersos, es decir, modelos con una selección de variables

- Al igual que en Ridge, la elección de un buen valor λ es crítico en Lasso; nuevamente, cross-validation es el método para su elección

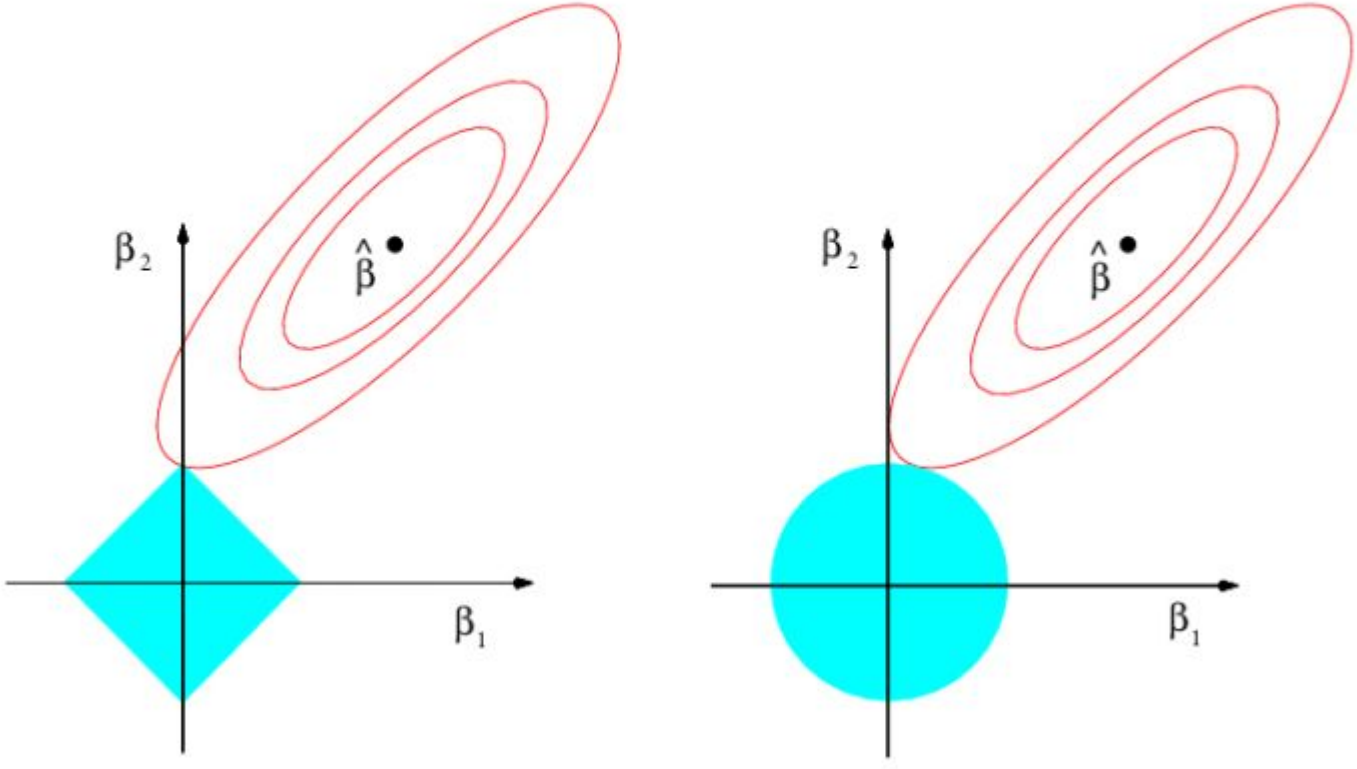

Standardized Coefficients
400
300
200
100
0
−200
20
50
100
200
500
2000
5000
λ
Standardized Coefficients
400
300
200
100
0
−100
−300
0.0
0.2
0.4
0.6
0.8
1.0
$\|\hat{\beta}_\lambda^L\|_1/\|\hat{\beta}\|_1$
Income
Limit
Rating
Student

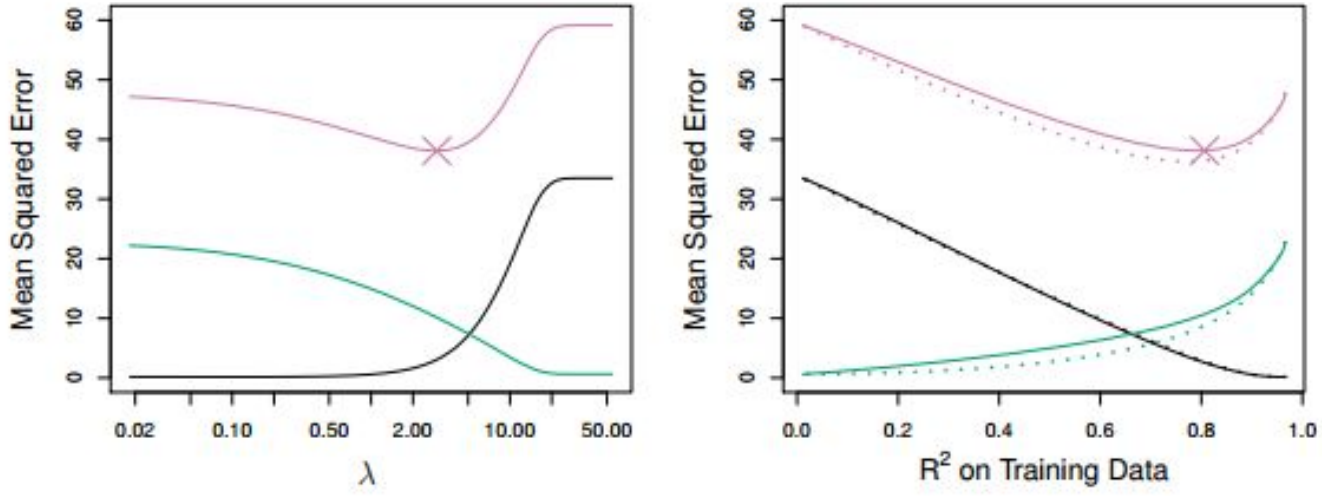
- ¿Por qué Lasso, a diferencia de Ridge, resulta en coeficientes estimados exactamente igual a cero?
- Uno puede mostrar que la estimación de coeficientes de las regresión Lasso y Ridge resuelve estos problemas, respectivamente.

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

$$\underset{\beta}{\text{minimize}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s,$$

Lasso, graficamente



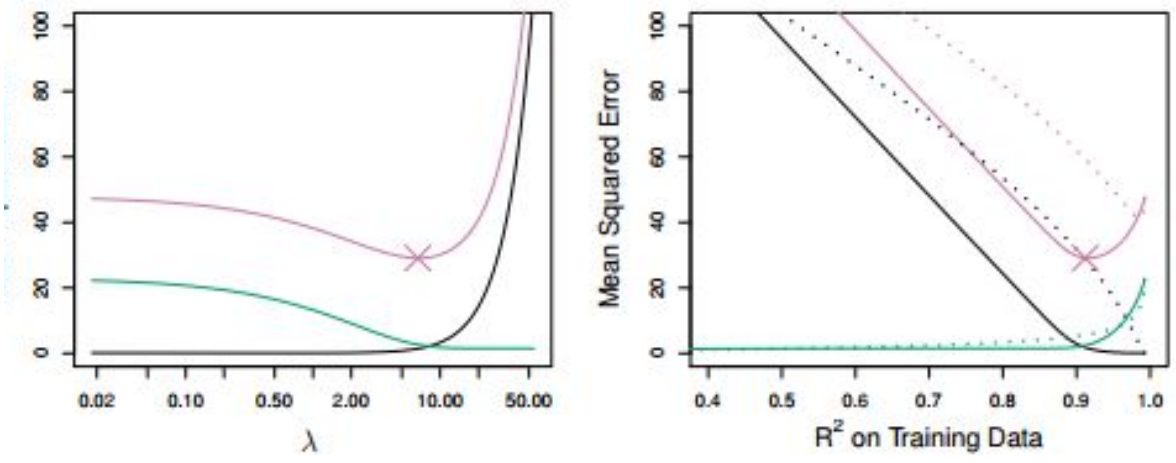


Izquierda: Sesgo cuadrado (negro), la varianza (verde) y el MSE del test (violeta)
Derecha: Comparación del Sesgo cuadrado, la varianza y el MSE del test, entre Lasso (llena) y Ridge (punteada).

Las cruces indican el mínimo MSE

Estos datos se generaron haciendo que todos los coeficientes fueran diferentes a cero.

En este caso, los dos modelos tienden a performar prácticamente igual. Ridge tiene una menor varianza y por eso parece mejorar respecto a Lasso



Izquierda: Sesgo cuadrado (negro), la varianza (verde) y el MSE del test (violeta). Los datos simulados, son similares a los anteriores, pero en este caso solo dos predictores estan relacionados con la respuesta.
Derecha: Comparación del Sesgo cuadrado, la varianza y el MSE del test, entre Lasso (llena) y Ridge (punteada).

Las cruces indican el mínimo MSE

Estos datos se generaron haciendo que solamente dos coeficientes fueran diferentes a cero.

De esta forma, vemos cómo Lasso mejora claramente la performance con respecto a Ridge, tanto en lo referido a variancia como a MSE.

- Los últimos dos ejemplos demuestran que ningún método va a dominar por completo al otro.

- En general, se esperaría que Lasso performe mejor, cuando la cantidad de predictores asociados a la respuesta, es baja.

- Sin embargo, el número de predictores asociados a una respuesta, nunca es conocido a priori en un caso real.

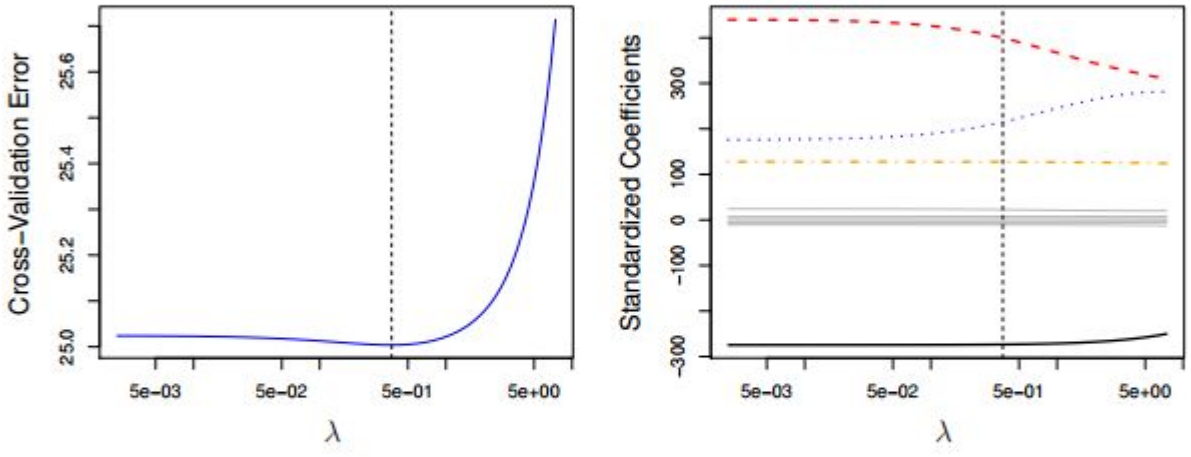
- Una buena práctica para elegir entre uno o el otro es a través de cross-validation.

- Se necesita un método para poder ajustar el hiperparámetro λ o s, respectivamente.

- **Cross-validation** es una manera simple de atacar este problema. Se elige un rango de valores que puede tomar el hiperparámetro, y luego se computan los errores que devuelve cross-validation, para cada caso.

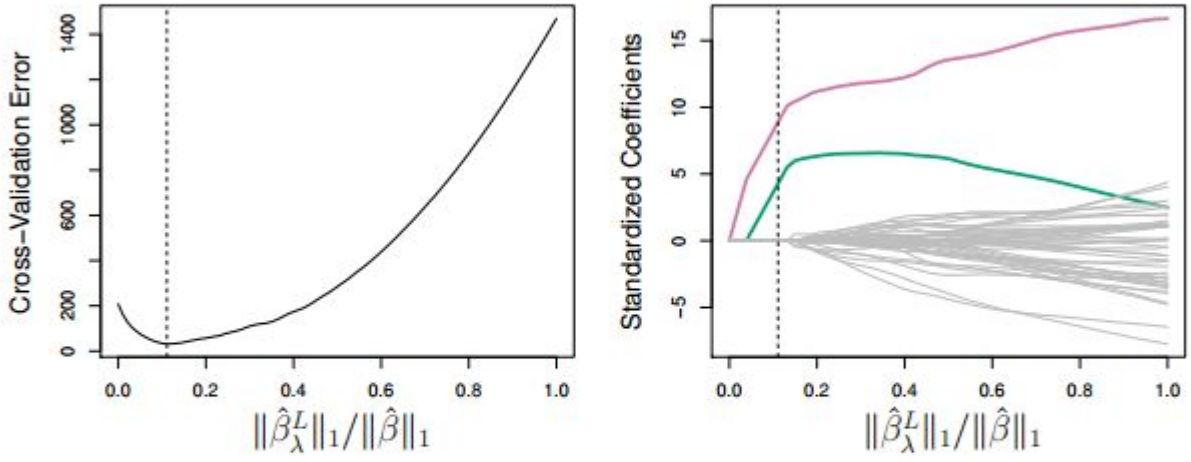
- Se elige el hiperparámetro asociado al menor error computado.

- Finalmente, “re-fiteamos” el modelo con el hiperparámetro elegido.



Izquierda: El error que resulta de cross validation, al aplicar una regresión Ridge al dataset *credit*, para un rango de valores de **λ**

Derecha: El coeficiente estimado en función de **λ**. la línea vertical punteada, indica el **λ** elegido por cross-validation.



Izquierda: Los MSE de un cross-validation de 10 particiones, para Lasso, aplicado EN UN DATASET SIMULADO

Derecha: Los coeficientes Lasso correspondientes a cada partición del cross-validation.

La línea vertical punteada, indica el caso en el que el cross-validation dió el menor error.

$$\lambda_1||\beta||_1 + \lambda_2||\beta||_2^2 = \lambda(||\beta||_1 + \alpha||\beta||_2^2)$$

- ElasticNet combina (linealmente) lo mejor de ambos mundos.
- El par´ametro λ regula la complejidad del modelo. El par´ametro α regula la importancia relativa de Lasso vs. Ridge.
- Es posible obtener soluciones parsimoniosas y bien condicionadas.
- No free lunch!: ahora hay que calibrar dos hiperparámetros.

Cross Validation

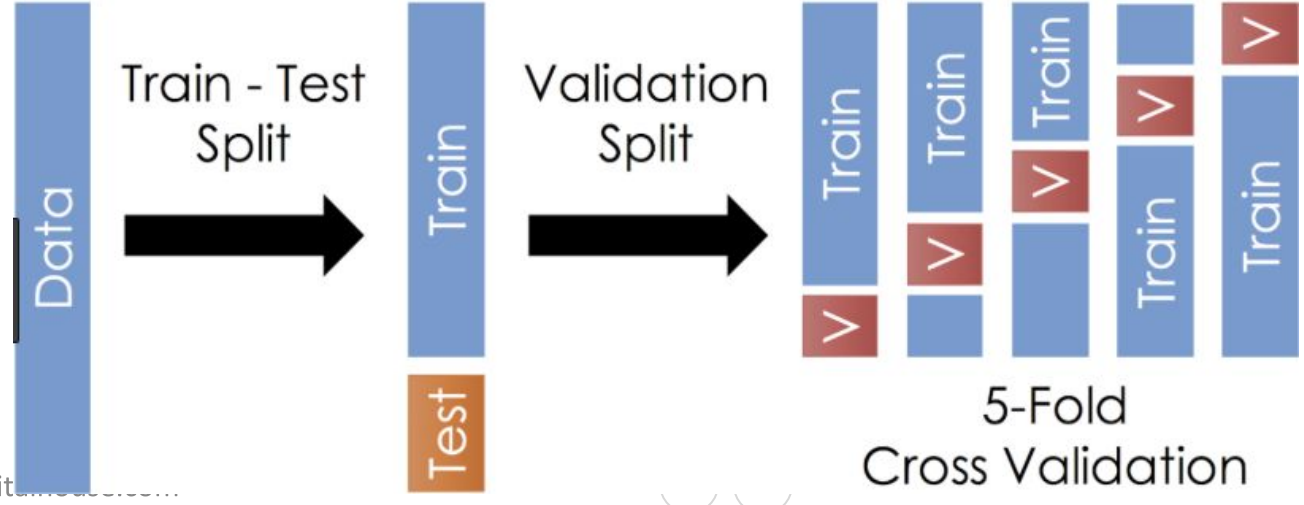
- Alpha es un parámetro desconocido (técnicamente es un hiperparámetro). Pero ¿cómo lo estimamos?

- Se hace a través de “cross-validation” (validación cruzada)

- ¿Qué es?
  - Es un método general para evaluar un determinado modelo predictivo (sean regresiones lineales, logísticas, árboles de clasificación, etc.)
  - Es similar a la particición en test y training set… solamente, que repetida varias veces.
  - Muy útil cuando no hay suficientes datos para generar un test-set grande

- ¿Para qué sirve? (algunos usos habituales):
  - Estimar los “hiperparámetros” de un modelo (por ejemplo, alfa en el caso de las técnicas de regularización)
  - Generar estimaciones del error de generalización

- ¿Cómo funciona?
  - Hacemos el split train/validación y test
  - Empezamos dividiendo el dataset de train/validación en k grupos (generalmente, 5 o 10 suele ser la medida convencional) del mismo tamaño.
  - En la primera iteración, el primer grupo generado pasa a ser un test test; el resto, pasa a ser el training set
    - Entrenamos un modelo sobre el training data
    - Hacemos las predicciones sobre el test set y calculamos el error sobre este test-set
  - Repetimos k veces, variando el test set en cada iteración.
  - Al final, promediamos los errores en cada una de las iteraciones



Práctica Guiada

Validación cruzada del hiper parámetro de regularización

Conclusión

- La regularización nos ayuda a evitar el sobreajuste limitando la complejidad del modelo
- Matemáticamente lo logra penalizando la complejidad dentro de la función de costo
- Modelos con regularización suelen tener mayor poder de generalización
- Para determinar el valor de los hiper-parámetros usados para regularizar, usamos validación cruzada