2023/

ITBA

Maestría en Management & Analytics

Clase 4 - Regularización

En esta clase vamos a profundizar los conceptos de **underfitting** y **overfitting**.

Además vamos a presentar algunas técnicas de **regularización** que sirven para mitigar el overfitting.

Sesgo y varianza.

Dado el modelo:

$$y = f(X) + \epsilon$$

Donde epsilon es un término de error aleatorio con distribución:

$$\epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$$

Podemos obtener una estimación de *f* tal que:

$$\hat{y} = \hat{f}(X)$$

Sesgo y varianza.

La esperanza del error de predicción al cuadrado será:

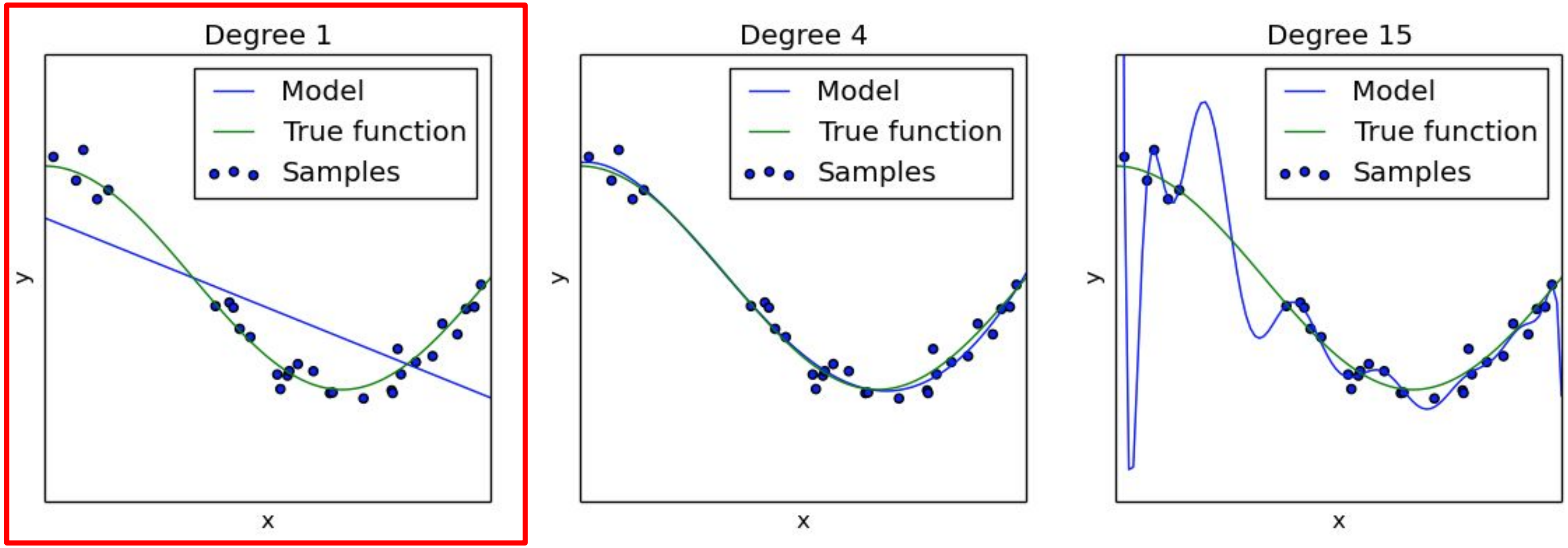
$$Err(x) = E\left[\left(y - \hat{f}(x)\right)^2\right]$$

Podemos descomponer la esperanza del error al cuadrado de la siguiente manera:

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_\epsilon^2$$

$$Err(x) = Sesgo^2 + Varianza + error\ irreducible$$

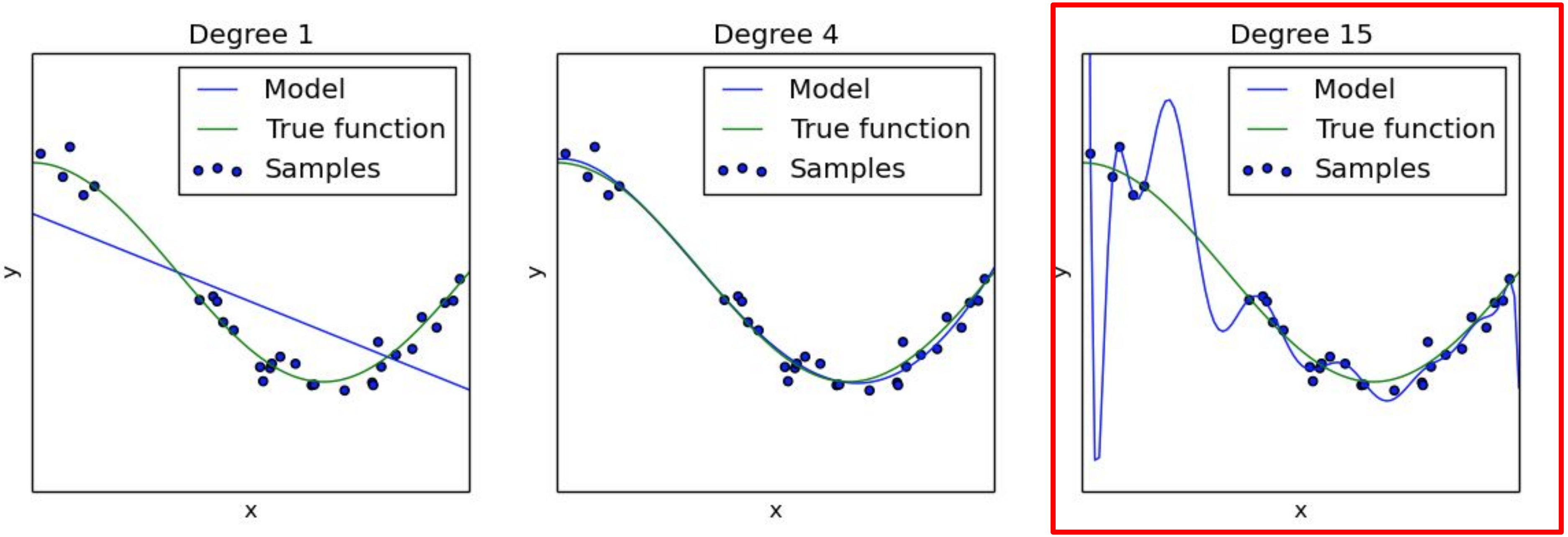
Sesgo y varianza.



Por **sesgo** nos referimos al error que se introduce al aproximar un problema de la vida real, que puede ser muy complejo, con un modelo muy simple (es decir que tiene poca capacidad de ajuste). En ese caso el modelo padecerá un sesgo sistémico, independientemente de la cantidad de observaciones que tenga el dataset. En este caso estamos ante la presencia de **underfitting**.

https://scikit-learn.org/0.15/auto_examples/plot_underfitting_overfitting.html

Sesgo y varianza.

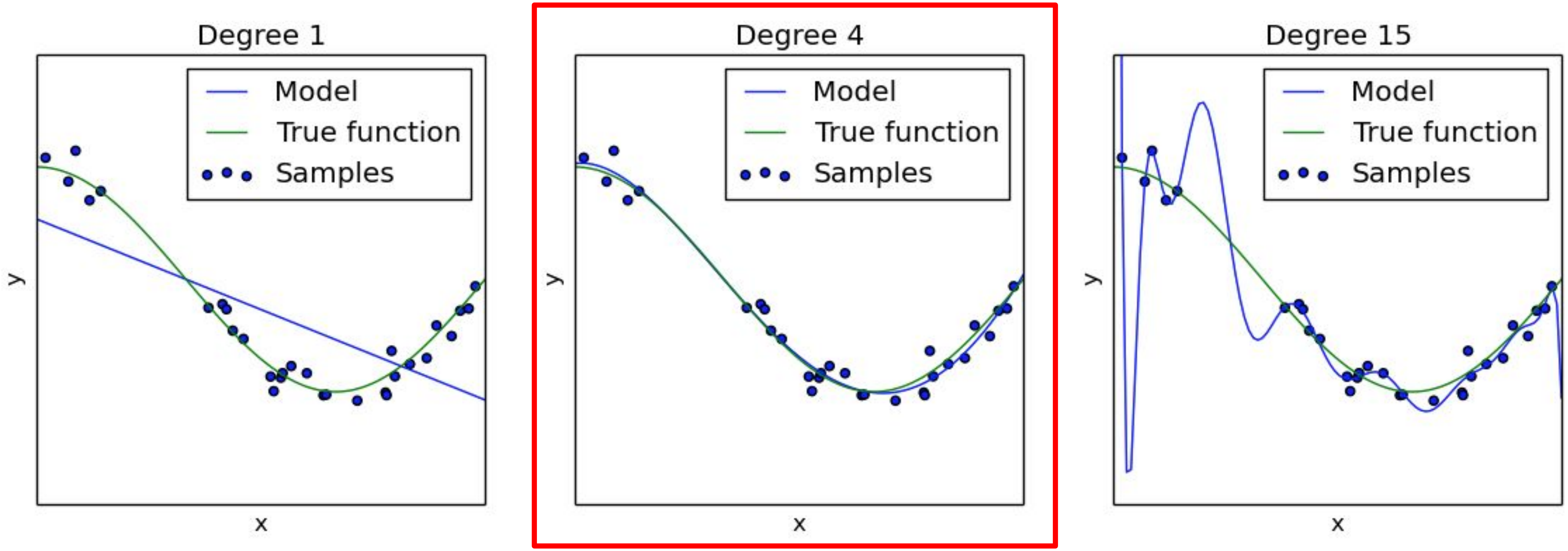


Por **varianza** nos referimos cuánto variaría la estimación de *f* si entrenáramos al modelo con un dataset diferente. Si una estimación varía mucho ante variaciones en el dataset de entrenamiento, entonces se entiende que está ajustando también al ruido del dataset (a las variaciones aleatorias de epsilon) y por lo tanto estamos ante la presencia de **overfitting**.

https://scikit-learn.org/0.15/auto_examples/plot_underfitting_overfitting.html

ITBA

Sesgo y varianza.



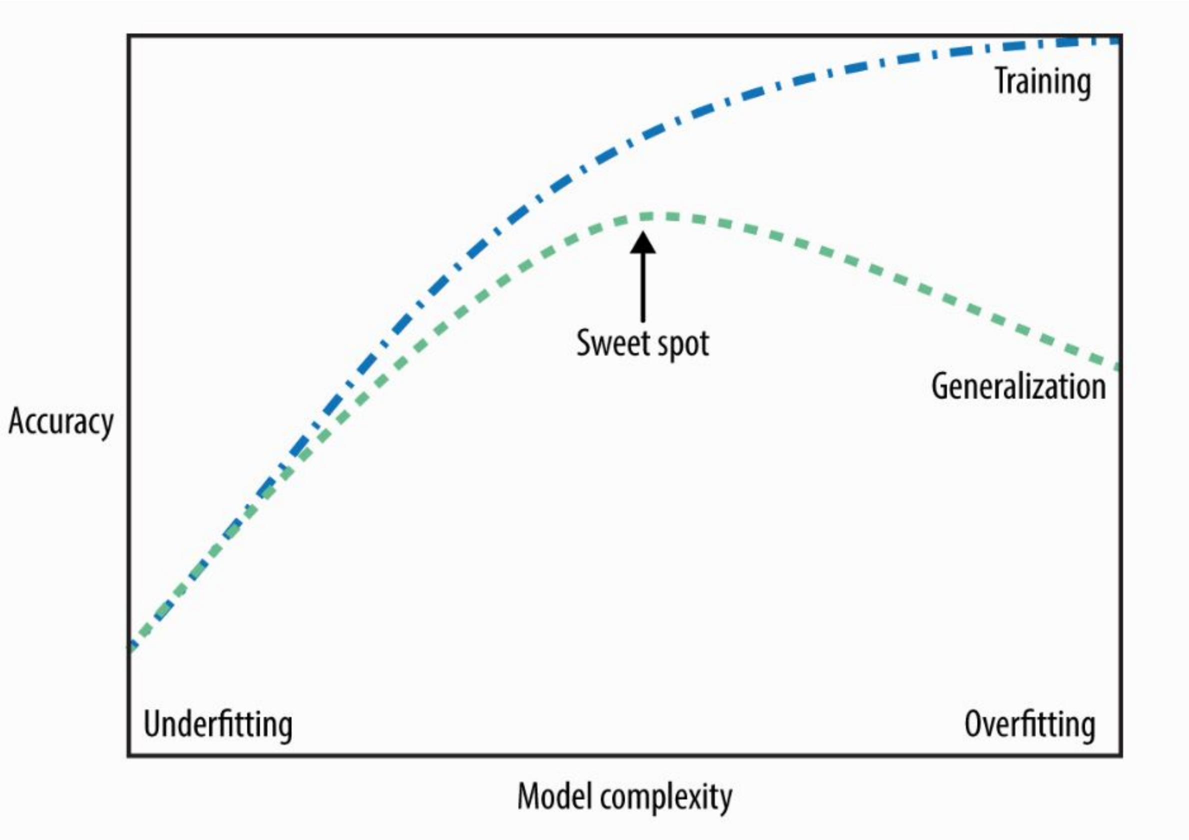
Por lo tanto el modelo debe tener el nivel de complejidad adecuado para poder ajustarse a la dinámica subyacente de los datos (la señal), sin ajustarse al ruido. Esto le permitirá poder realizar predicciones adecuadas ante datos que no pertenecen al dataset de entrenamiento.

https://scikit-learn.org/0.15/auto_examples/plot_underfitting_overfitting.html

Underfitting y overfitting

Nuestro objetivo es que el modelo tenga buena capacidad de **generalización.**

- Para modelos con **alto sesgo**, la performance del modelo con el set de testeo es similar a la performance con el set de entrenamiento, y ambas son bajas.
- Para modelos con **alta varianza**, la performance del modelo el set de testeo es mucho peor que la performance el set de entrenamiento.
- Buscamos que **optimizar la performance** del modelo en el **set de testeo.**

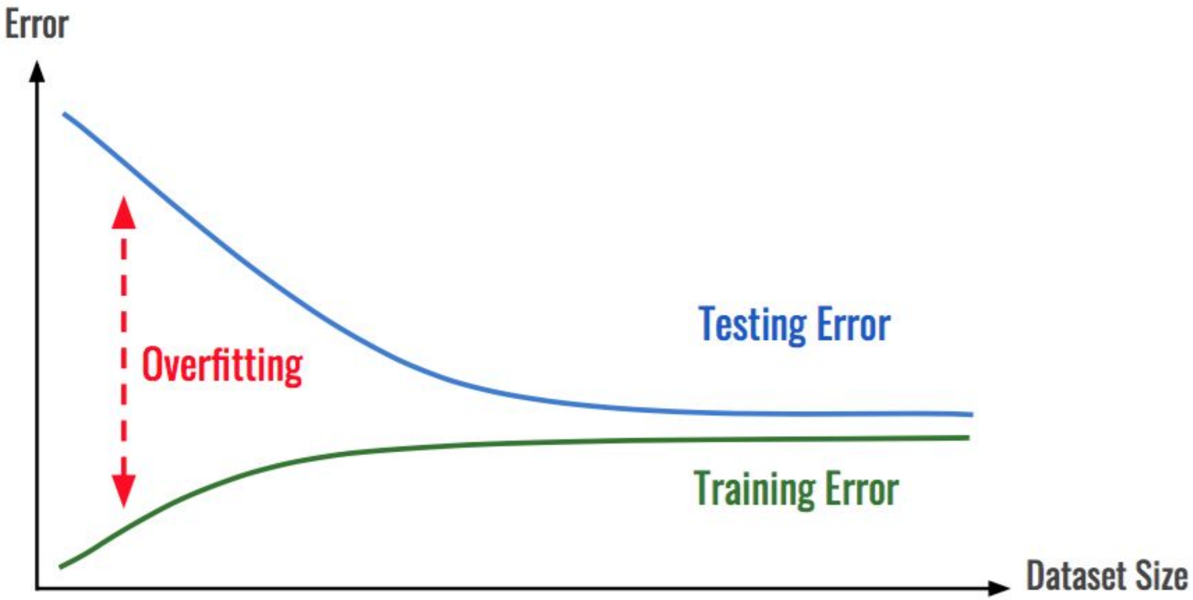


Underfitting y overfitting

El overfitting también depende del **tamaño del dataset de entrenamiento**.

Un plot del score de entrenamiento/testeo con respecto al tamaño del set de entrenamiento se conoce como **learning curve**.

A igualdad de complejidad del modelo, el error el set de entrenamiento y testeo convergen a medida que aumenta el número de muestras de entrenamiento.



Una vez que tenemos datos suficientes para que el modelo haya convergido, agregar más muestras de entrenamiento no ayudará a mejorar el score.

La única forma de incrementar la performance del modelo en este caso es usar otro modelo, generalmente más complejo.

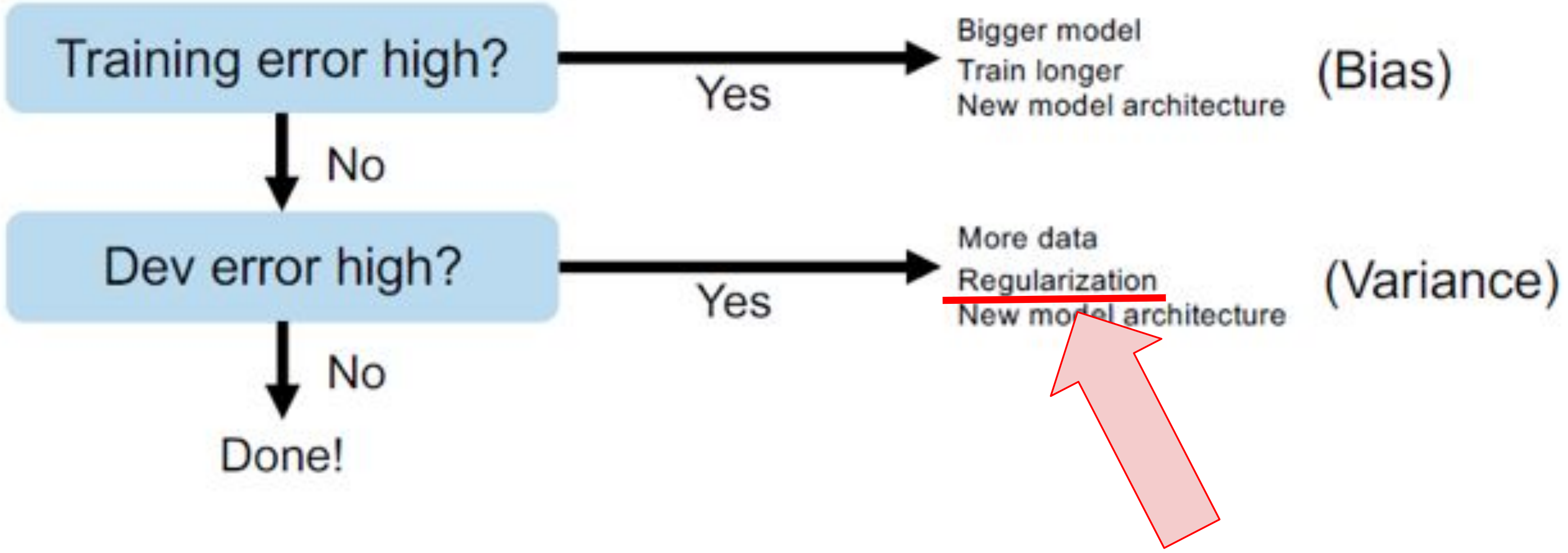
ITBA

Underfitting y overfitting

El dilema del **trade-off entre sesgo y varianza** concierne principalmente a los modelos de **machine learning tradicional**.

Actualmente, en proyectos de **deep learning** a menudo tenemos acceso a datos abundantes y podemos usar redes neuronales muy grandes (aprendizaje profundo). Por lo tanto, hay menos trade-off, ya que ahora hay más opciones para reducir el sesgo sin dañar la varianza, y viceversa.

Receta básica de machine learning (presentada por Andrew Ng):



Regularización

Las técnicas de **regularización** buscan **combatir el overfitting** de diferentes maneras.

En esta clase vamos a ver 2 técnicas muy difundidas:

- Regularización **L2**, a la que le corresponde la **regresión Ridge**
- Regularización **L1**, a la que le corresponde la **regresión Lasso**

Ambas son técnicas de "*shrinkage*", es decir que buscan reducir el tamaño de los parámetros estimados agregando una penalidad al problema de optimización que estima dichos valores.

Veamos a estas técnicas en detalle.

Regularización

Recordemos que la **regresión lineal** estima los parámetros minimizando la siguiente función, es decir el **error cuadrático**:

$$CF = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

En machine learning, la función que se busca minimizar para estimar los parámetros se llama función de costo (CF, por sus siglas en inglés).

Regularización

Desarrollando la expresión del modelo lineal obtenemos:

$$CF = \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2$$

Regularización

Las técnicas de **regularización** agregan una **“penalidad”** a esa función de costo:

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha f(\boldsymbol{w})$$

Regularización

Las técnicas de **regularización** agregan una **“penalidad”** a esa función de costo:

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha \boxed{f(\mathbf{w})}$$

$f(\mathbf{w})$ es una función de los parámetros del modelo. En particular, veremos que en las técnicas de regularización que vamos a ver hoy, se trata de normas del vector de parámetros. Es decir que a mayor tamaño de los parámetros, mayor penalidad.

Regularización

Las técnicas de **regularización** agregan una **“penalidad”** a esa función de costo:

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \boxed{\alpha} f(\boldsymbol{w})$$

α es un parámetro que “regula” la fuerza de la penalización: cuanto más grande es, mayor es la penalización. En algunos textos a este parámetro se lo denomina lambda.

Regularización L2 o regresión Ridge

Cuando en la penalidad utilizamos la **norma L2 o euclídea**, obtenemos la **regularización L2** que corresponde a la **regresión Ridge**. Su función de costo es:

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha \sum_{j=1}^{p} w_j^2$$

$$L2: \|\mathbf{w}\|_2^2 = \sum_{j=1}^{p} w_j^2$$

Regularización L2 o regresión Ridge

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha \sum_{j=1}^{p} w_j^2$$

- Como en el caso de la regresión lineal, se busca **minimizar la CF**
- Sin embargo, existe un **término de penalización**, que es menor cuando los pesos se acercan a cero, por lo tanto tiene el efecto de achicar los mismos hacia cero (tanto si son negativos como positivos)
- Como vimos el término **alpha** regula el efecto de la penalización. A mayor alpha, mayor penalización y por lo tanto la estimación resulta en pesos más chicos. Alpha es un **hiperparámetro**. Tenemos que definirlo previamente a entrenar el modelo ya que el modelo no lo puede aprender. ¿Cómo se elige el valor óptimo de alpha? La clase que viene va a estar dedicara a este tema.

Regularización L2 o regresión Ridge

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha \sum_{j=1}^{p} w_j^2$$

- Un aspecto metodológico importante es que para aplicar estas técnicas de regularización, tenemos que **normalizar los datos**. Esto se debe a que, al penalizar los parámetros por su tamaño, la escala en la que están medidas las variables se vuelve absolutamente relevante.
- Con el modelo de regresión lineal sin regularización, el valor de los parámetros compensaba por las unidades de medida de las variables, por lo que no afectaba al resultado del modelo. Al aplicar regularización, no queremos que efectos de escala afecten la importancia relativa de los parámetros. Por este motivo, estandarizamos a todas las variables de modo tal que de llevar a **todas las variables a una escala común**.

Regularización L1 o regresión Lasso

Cuando en la penalidad utilizamos la **norma L1**, obtenemos la **regularización L1** que corresponde a la **regresión Lasso**. Su función de costo es:

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha \sum_{j=1}^{p}\left|w_j\right|$$

$$L1: \|\boldsymbol{w}\|_1 = \sum_{j=1}^{p}\left|w_j\right|$$

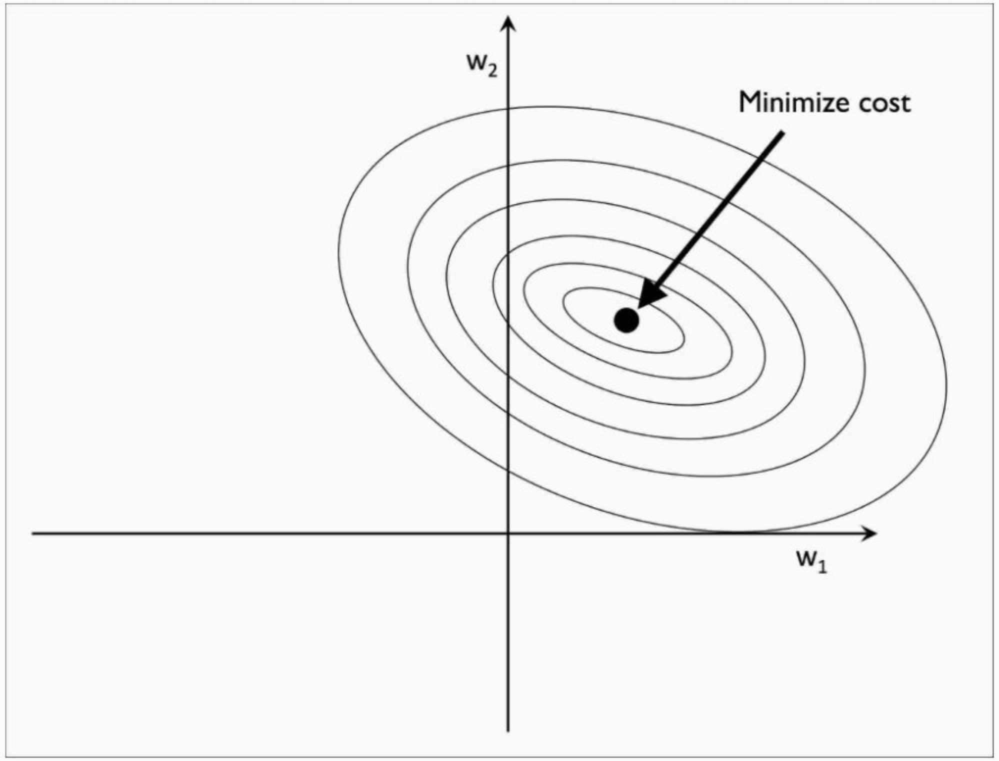
Regularización L1 o regresión Lasso

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \alpha \sum_{j=1}^{p} |w_j|$$

- 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 α sea lo suficientemente grande.
- Por lo tanto, el Lasso hace selección de variables.

Interpretación geométrica

Podemos representar geométricamente los contornos de la minimización del error cuadrático de un modelo lineal con 2 variables del siguiente modo:



Interpretación geométrica

Adicionalmente, se puede demostrar que la estimación de los coeficientes de Ridge y Lasso, resuelven los siguientes problemas, respectivamente:

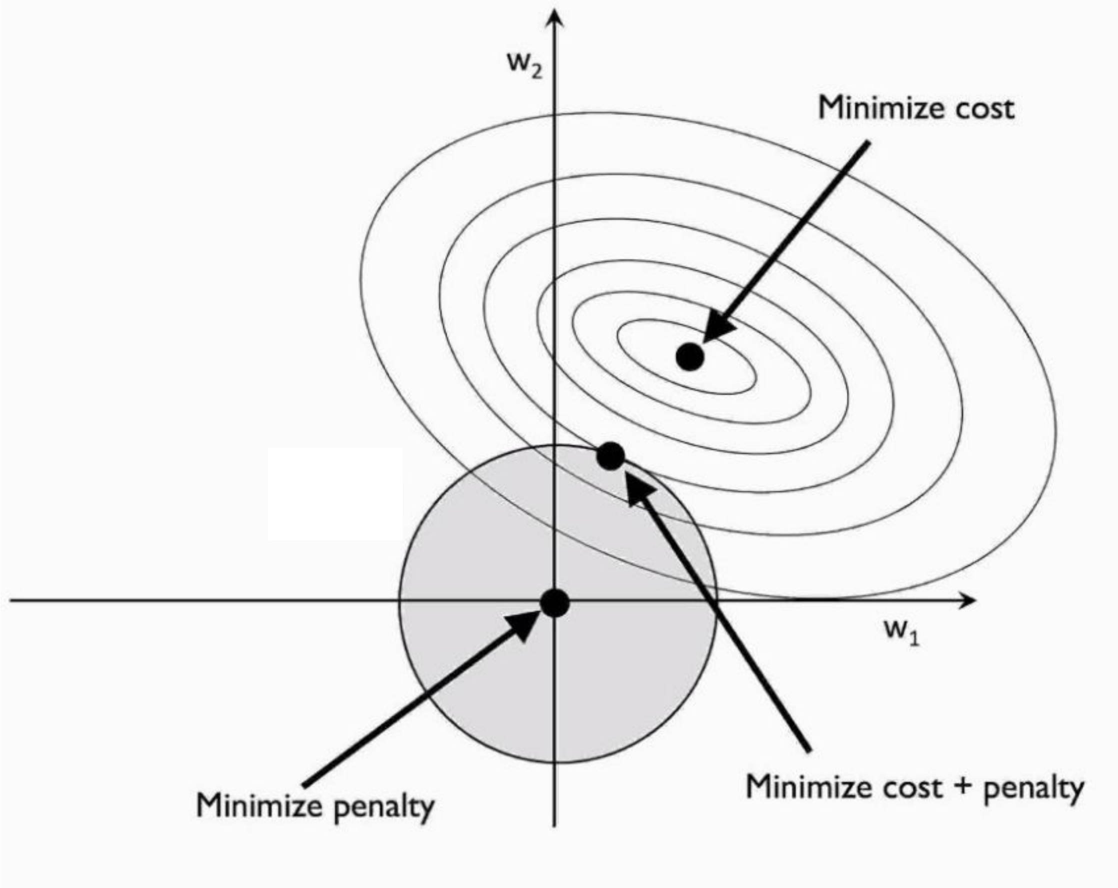
$$Minimizar_w \left\{ \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2 \right\} \quad s.a. \ \sum_{j=1}^{p} w_j^2 \le s$$

$$Minimizar_w \left\{ \sum_{i=1}^{N} \left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij} \right)^2 \right\} \quad s.a. \ \sum_{j=1}^{p} |w_j| \le s$$

Es decir que por cada valor de α , existe un s tal que se obtengan las mismas estimaciones

Interpretación geométrica

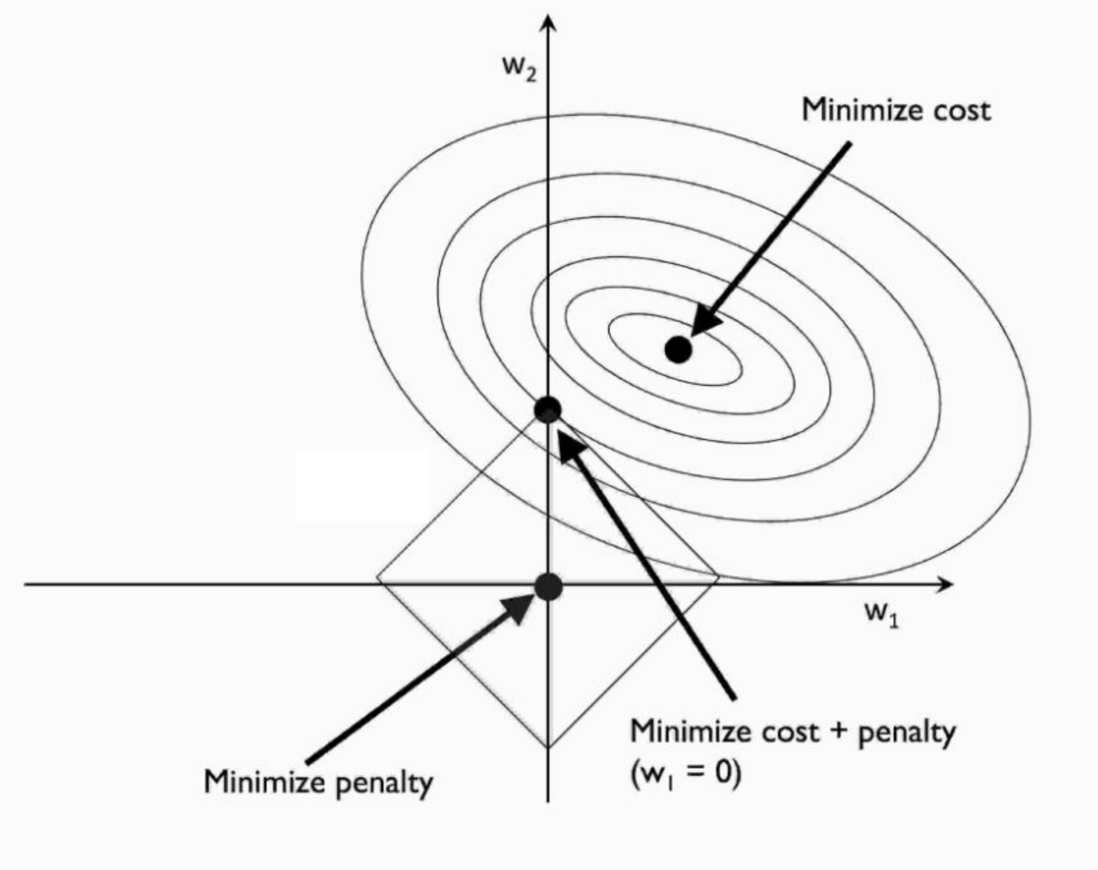
En el caso de la regresión Ridge, tenemos:



$$w_1^2 + w_2^2 \leq s$$

Interpretación geométrica

En el caso de la regresión Lasso, tenemos:



$$|w_1| + |w_2| \leq s$$

ElasticNet

- Lasso tiene la ventaja de que realiza selección de variables al llevar algunos coeficientes a cero.
- Sin embargo puede suceder que las soluciones de Lasso sean muy dependientes del dataset, lo que puede generar problemas.
- Una forma de resolver este problema es combinando las regularizaciones L1 y L2. Esto es lo que hace la técnica **ElasticNet**.

ElasticNet

$$CF = \sum_{i=1}^{N}\left(y_i - w_0 - \sum_{j=1}^{p} w_j x_{ij}\right)^2 + \lambda\left(\alpha\,\|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2\right)$$

$$con \quad \alpha \in [0,1]$$

- ElasticNet puede obtener muy buenos resultados, pero hay que tener en cuenta que ahora tenemos que **optimizar 2 hiperparámetros**.
- En este caso, α no es el mismo parámetro que en L1 y L2.

Recapitulando:

- Presentamos el **trade-off entre sesgo y varianza**.
- Presentamos una **receta básica** de Andrew Ng para proyectos de **Machine Learning**.
- Presentamos el concepto de **regularización** y las técnicas Ridge, Lasso y ElasticNet.

GRACIAS