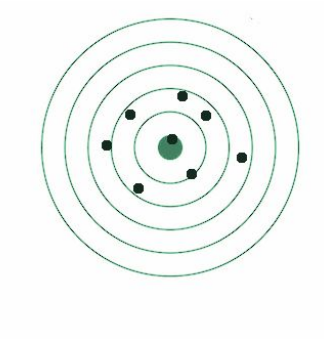
DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Aprendizaje Automático

Clase 5:
Error irreducible
Sesgo y Varianza
Ensambles (Bagging, Random Forest)

Sesgo y Varianza



"Las nociones de sesgo y varianza ayudan a explicar cómo los algoritmos muy simples pueden superar a los más sofisticados y cómo los ensambles pueden superar a los modelos individuales"

[Domingos, Pedro. "**A unified bias-variance decomposition**." Proceedings of 17th international conference on machine learning. Stanford: Morgan Kaufmann, 2000.]

http://homes.cs.washington.edu/~pedrod/bvd.pdf

Tarea del aprendizaje supervisado

Objetivo del aprendizaje supervisado

Estimar la **función determinista f** que determina la relación $X \rightarrow Y$:

Es decir, estimar **f** tal que **Y = f(X)** a través de un modelo $\hat{h}_D$.

Problema: si la función **f** es determinista, ¿cómo puede ser que haya etiquetas contradictorias entonces?

Ej: $x^{(1)} = [1, 3, 14, 4] \rightarrow$ *etiquetado como Perro (es decir* $y^{(1)} = C_1$*)*
$x^{(2)} = [1, 3, 14, 4] \rightarrow$ *etiquetado como Gato (es decir* $y^{(2)} = C_2$*)*

Posibles causas:

- ***Variables no observadas:*** *El proceso* ***"real"*** *utiliza* ***más atributos***, ***no representados*** *en las dimensiones de X*
- ***Errores de medición:*** *Errores en la recolección de datos, como errores de redondeo, de registro, etc,*
- ***Errores de etiquetado:*** *Las etiquetas están mal, alguien (o algo) se* ***equivocó al etiquetar (o simplemente no hay acuerdo)***.
- ***Variación aleatoria en Y:*** *Puede haber una* ***variación aleatoria en la salida*** *que no está relacionada con la entrada.*
- ***Variación aleatoria en X:*** *En algunos casos, las variables de entrada pueden generarse mediante procesos estocásticos, como caminatas aleatorias, que crean una* ***aleatoriedad inherente o ruido.***

Tarea del aprendizaje supervisado

Objetivo del aprendizaje supervisado (revisado)

Estimar la **función determinista f** que determina la relación $X \rightarrow Y$:

Es decir, estimar **f** tal que $\mathbf{Y = f(X) + \varepsilon}$ a través de un modelo $\hat{h}_D$. En donde ε es el **"el error irreducible".**

- ε **es una variable aleatoria** que representa la cantidad de ruido o incertidumbre en la relación entre las variables de entrada y la variable de salida. ε podría depender de X, pero en general suponemos que no.

- Por ejemplo,
  - En regresión se espera que el término de error ε siga una **distribución normal** con **media de cero** y **varianza finita**.
  - En clasificación podemos pensar algo que **cambia las respuestas al azar** (de 0 a 1 o 1 a 0, en caso binario) — por lo tanto sumar ε es un abuso de notación.

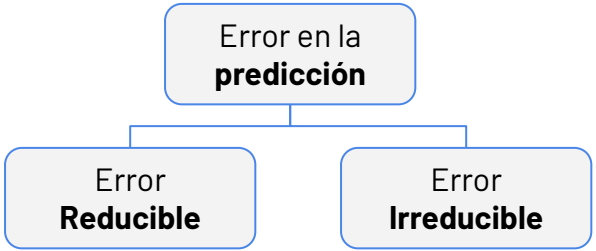
Esto es un "modelo"de la realidad (una serie de suposiciones que simplifica encarar el problema). Para más opciones o justificaciones, ver el ESLR (cap 2, sec 2.6)

Representando el error de generalización

Concentrémonos entonces en el error que sí podemos reducir.

Hasta ahora vimos cómo minimizar el **error en entrenamiento** y confiamos en que eso ayudará para reducir el error de generalización.

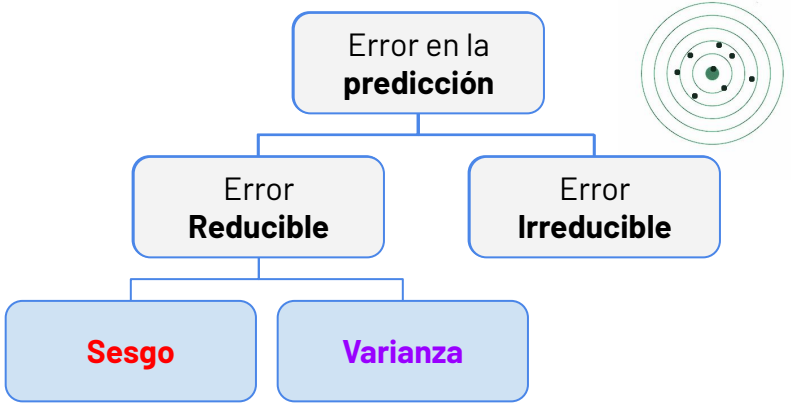
Hoy estudiaremos **el error de generalización** y lo descompondremos para entender de dónde proviene.



Empecemos por caracterizar cuál será el error que esperamos encontrar al clasificar **una instancia** $\mathrm{x}^{(\mathrm{i})}$ al utilizar un modelo construido a partir del **algoritmo** L

$$Error\_esperado(x^{(i)}; L) \ ?$$

Sesgo y Varianza estadísticas



Objetivo: Aprender la relación $\mathbf{Y = f(X) + \varepsilon}$ a través de un modelo $\hat{h}_D(X)$.

$$Error\_esperado(x^{(i)}; L) = \mathrm{E}_{D_n}\left[error\left(y^{(i)}, \hat{h}_{(L,D_n)}(x^{(i)})\right)\right]$$

$$= \mathrm{E}_{D_n}\left[error\left(f(x^{(i)}) + \varepsilon, \hat{h}_{(L,D_n)}(x^{(i)})\right)\right]$$

Para el caso de regresión.
En donde error = MSE.
$\mathrm{MSE(a,b)} = \mathrm{(a-b)}^2$

$$\overset{\text{reg}}{=} \mathrm{E}_{D_n}\left[\left(f(x^{(i)}) + \varepsilon - \hat{h}_{(L,D_n)}(x^{(i)})\right)^2\right]$$

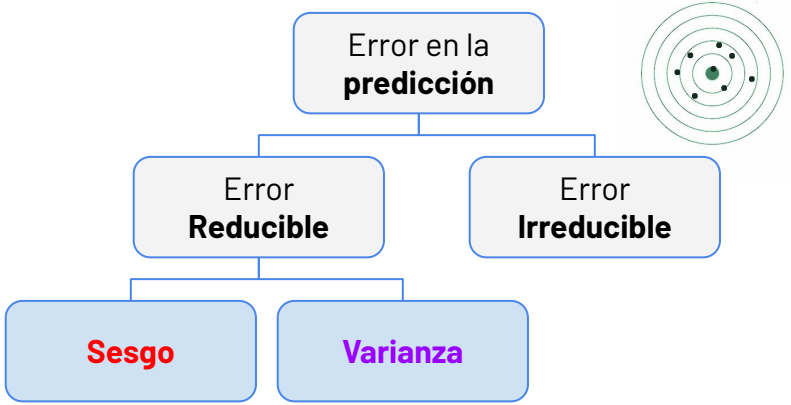
$$= \dots$$

Ejercicio de la práctica "Bias-Variance decomposition"

$$= \left(\text{Sesgo}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right]\right)^2 + \text{Var}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] + \text{Var}(\varepsilon)$$

En donde $\mathbf{E}_{\mathbf{Dn}}$ refiere a la esperanza sobre todos los posibles datasets (muestreados a partir de $\mathbf{P(X,Y)}$ de tamaño n)
$\hat{h}_{(L,Dn)}(x^{(i)})$ refiere a la predicción un modelo entrenado utilizando el algoritmo L sobre los datos D_n)

Sesgo y Varianza estadísticas



Objetivo: Aprender la relación $\mathbf{Y = f(X) + \varepsilon}$ a través de un modelo $\hat{h}_D(X)$.

$$Error\_esperado(x^{(i)}; L) \overset{\text{reg}}{=} \left(\text{Sesgo}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] \right)^2 + \text{Var}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] + \text{Var}(\varepsilon)$$

Sesgo (bias): Dado un algoritmo, cuánto esperamos que una predicción **difiera** del **valor real** (técnicamente, del valor medio real)

$$\text{Sesgo}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] = \text{E}_{D_n}\left[error\left(\hat{h}_{(L,D_n)}(x^{(i)}), f(x^{(i)})\right)\right]$$

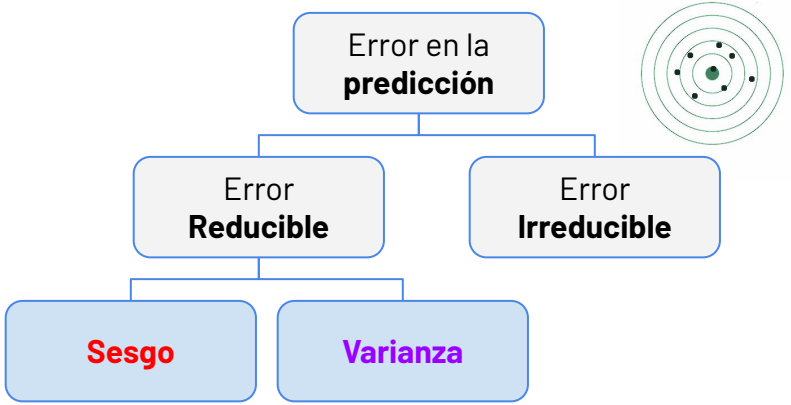
Nota, acá error(a,b) = a-b (interesa el signo).

Varianza: Dado un algoritmo, cuánto esperamos que una predicción **difiera** del **valor más común de dicho algoritmo**.

$$\text{Var}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] = \text{E}_{D_n}\left[error\left(\hat{h}_{(L,D_n)}(x^{(i)}), \text{E}_{D_n}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right]\right)\right]^2$$

En donde $\mathbf{E_{Dn}}$ refiere a la esperanza sobre todos los posibles datasets (muestreados a partir de $\mathbf{P(X,Y)}$ de tamaño $\mathbf{n}$)
$\hat{h}_{(L,Dn)}(x^{(i)})$ refiere a la predicción un modelo entrenado utilizando el algoritmo L sobre los datos D_n)

Sesgo y Varianza estadísticas



Objetivo: Aprender la relación $\mathbf{Y = f(X) + \varepsilon}$ a través de un modelo $\hat{h}_D(X)$.

Simplificando un poco la notación

$\mathbf{E_D = E}$ y $pred^{(i)} \overset{\text{def}}{=} \hat{h}_{(L,D_n)}(x^{(i)})$

$$Error\_esperado(x^{(i)}; L) \overset{\text{reg}}{=} \left(\text{Sesgo}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] \right)^2 + \text{Var}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] + \text{Var}(\varepsilon)$$

Sesgo (bias): Dado un algoritmo, cuánto esperamos que una predicción **difiera** del **valor real** (técnicamente, del valor medio real)

$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

Nota, acá error(a,b) = a-b (interesa el signo).

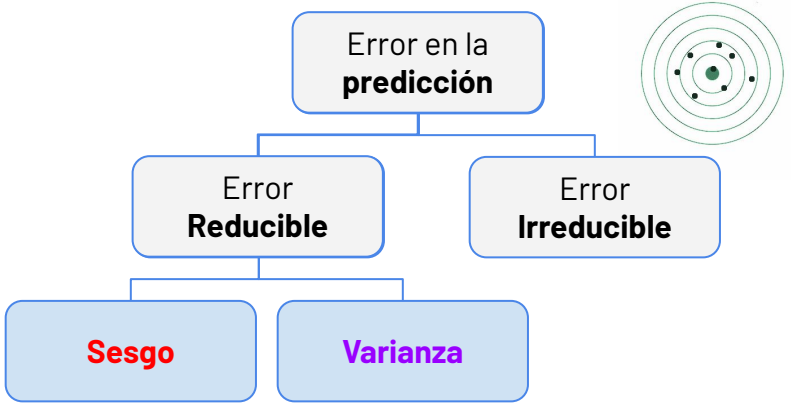
Varianza: Dado un algoritmo, cuánto esperamos que una predicción **difiera** del **valor más común de dicho algoritmo.**

$$\text{Var}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)\right]^2$$

Nota:

Sesgo Estadístico != Sesgo inductivo
Sesgo Estadístico != Sesgo de Equidad (fairness)
Sesgo Estadístico != Bias term (redes / regresión)

Sesgo y Varianza estadísticas



Objetivo: Aprender la relación $\mathbf{Y = f(X) + \varepsilon}$ a través de un modelo $\hat{h}_D(X)$.

$$Error\_esperado(x^{(i)}; L) \overset{\text{reg}}{=} \left(\text{Sesgo}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] \right)^2 + \text{Var}\left[\hat{h}_{(L,D_n)}(x^{(i)})\right] + \text{Var}(\varepsilon)$$

$$Error\_esperado(x^{(i)}) \overset{\text{clf}}{=} (Sesgo \ldots Varianza \ldots \varepsilon)??$$

En clasificación, **pese a que la fórmula no sea la misma,** también puede expresarse la fórmula del error esperado en términos del sesgo y la varianza.
Domingos, Pedro. **"A unified bias-variance decomposition."** Proceedings of 17th international conference on machine learning. Stanford: Morgan Kaufmann, 2000].

(lo tienen que leer para el cuestionario)

Interesa entonces seguir entendiendo estos conceptos y ver cómo podemos manipularlos.

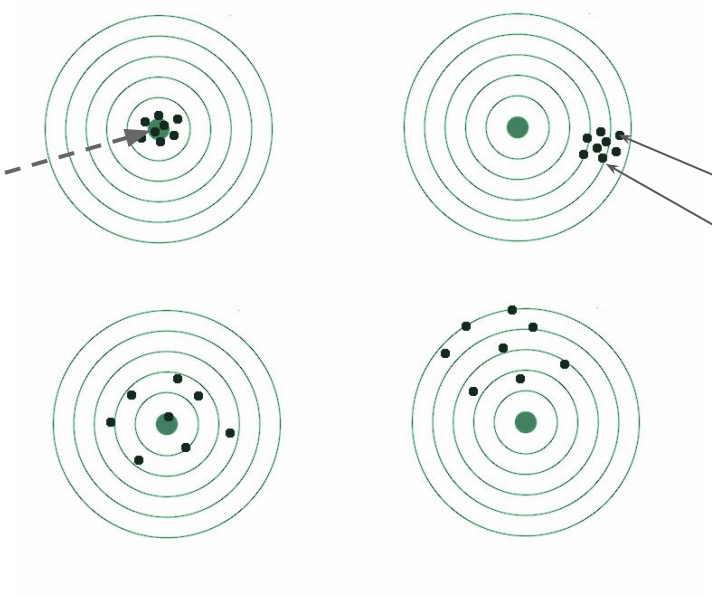
Visualizaciones

Sesgo y Varianza

Una visualización

$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$



$f(x^{(i)})$

Valor "verdadero" de la instancia i (o valor verdadero medio si consideramos ε)

Predicciones para $x^{(i)}$ (al variar el dataset de entrenamiento para L)

$\hat{h}_{D1,\,L}(x^{(i)})$

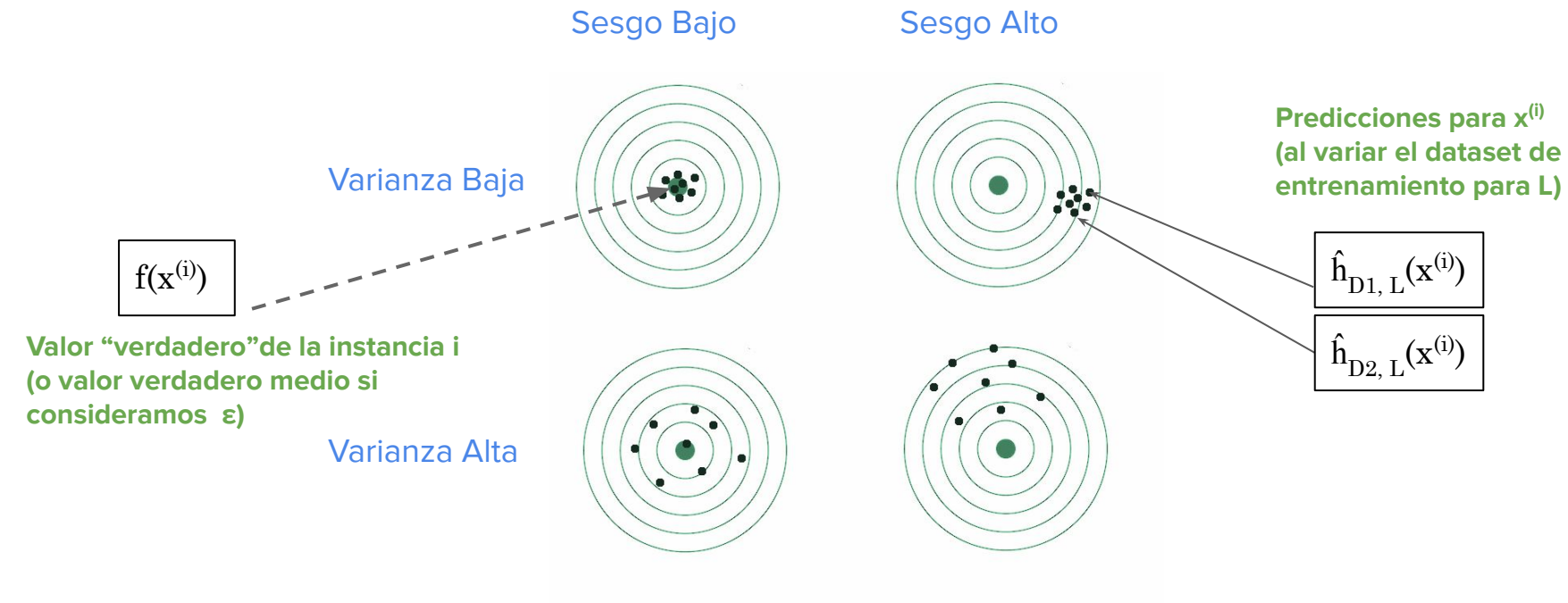
$\hat{h}_{D2,\,L}(x^{(i)})$

Sesgo y Varianza

Una visualización

$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$

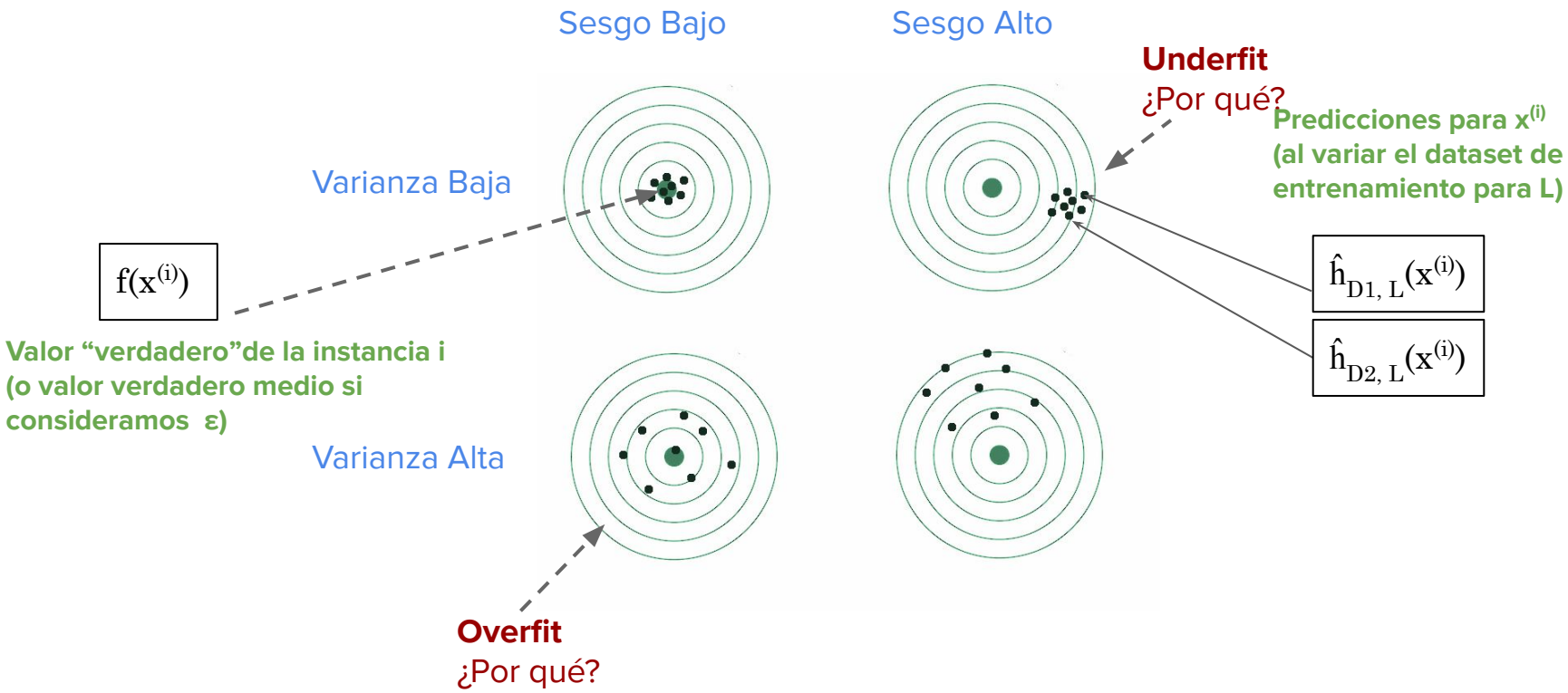


Sesgo y Varianza

Una visualización

$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$

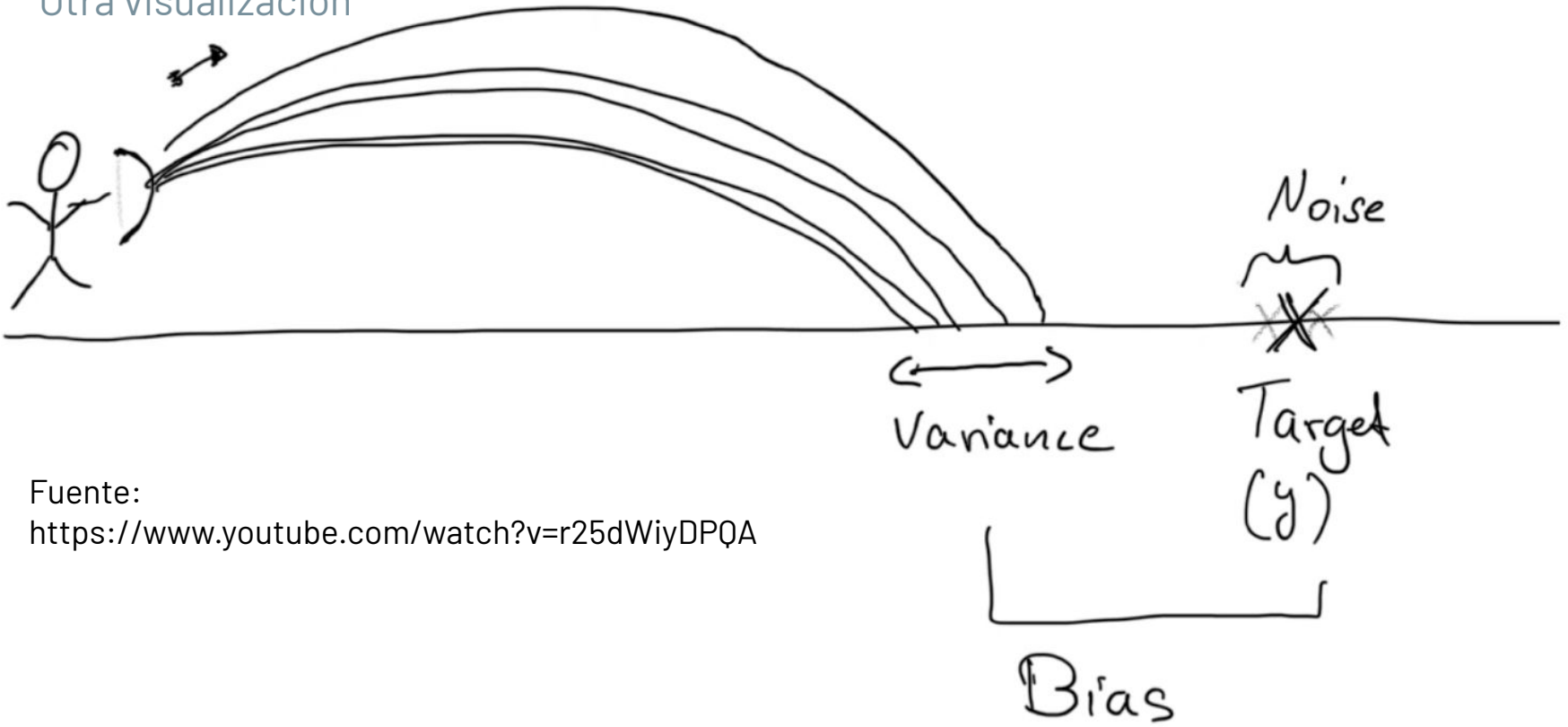


Sesgo y Varianza

Otra visualización

$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$



Fuente:
https://www.youtube.com/watch?v=r25dWiyDPQA

Para pensar

Clasificar los siguientes métodos según la potencialidad de bajo/alto sesgo y baja/alta varianza.

- Árboles de decisión
- KNN
- L/QDA.

¿Depende de los hiperparámetros?

Sesgo y Varianza

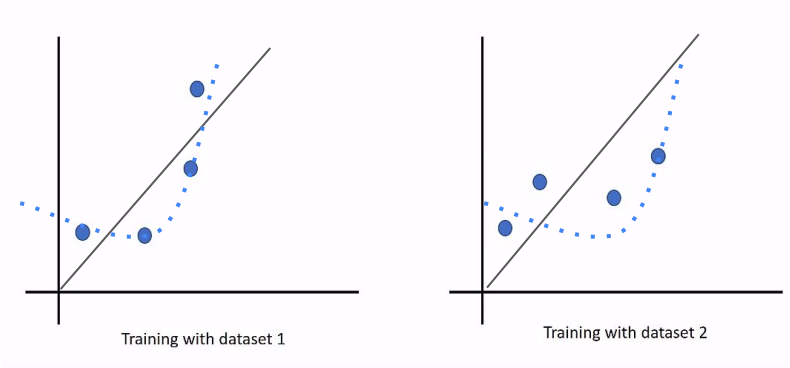
Visualización (regresión)

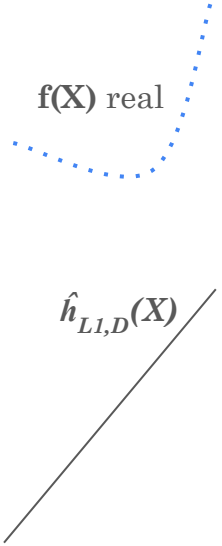
$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$

Algoritmo:
Regresión Lineal

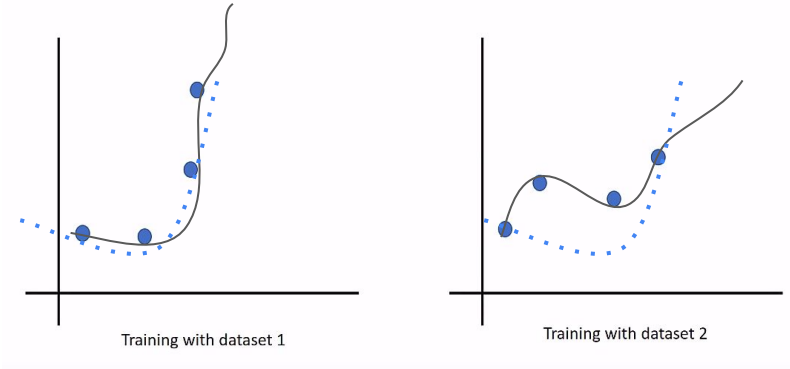
Sesgo: ??
Varianza: ??





Algoritmo:
Regresión Polinómica

Sesgo: **??**
Varianza: ??



$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

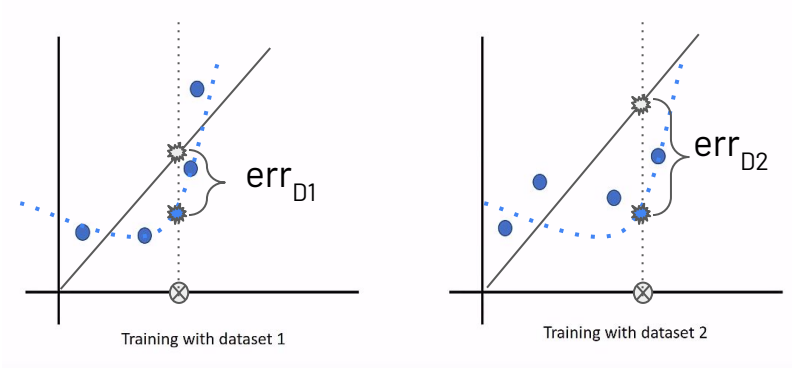
$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$

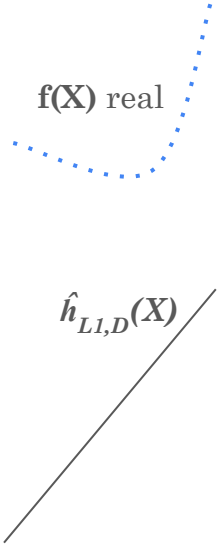
Sesgo y Varianza

Visualización (regresión)

Algoritmo:
Regresión Lineal

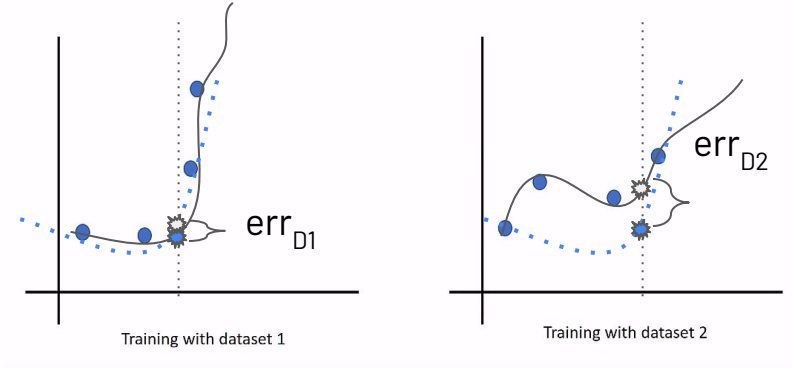
Sesgo: ??
Varianza: ??





Algoritmo:
Regresión Polinómica

Sesgo: ??
Varianza: ??



Sesgo y Varianza

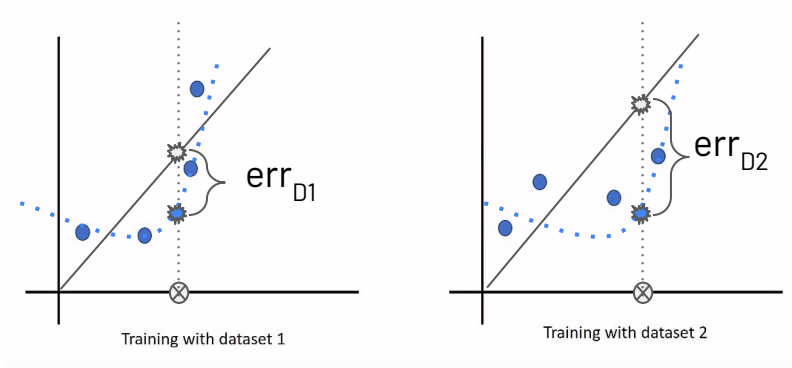
Visualización (regresión)

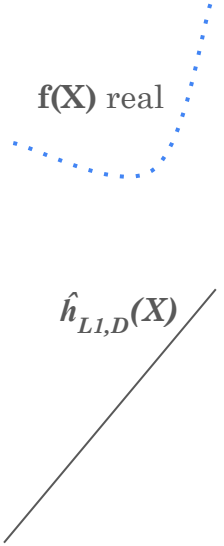
$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$

Algoritmo:
Regresión Lineal

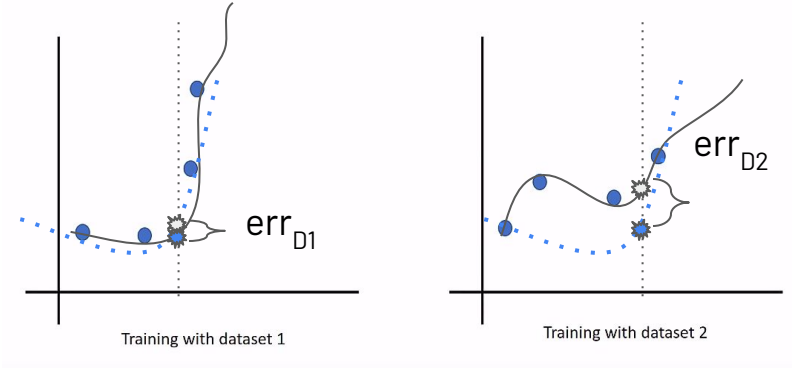
Sesgo: **Alto (Underfit)**
Varianza: ??

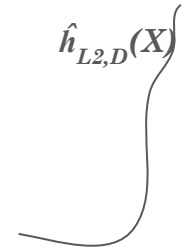




Algoritmo:
Regresión Polinómica

Sesgo: **Bajo**
Varianza: ??





$$\text{Sesgo}\left[pred^{(i)}\right] = \text{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

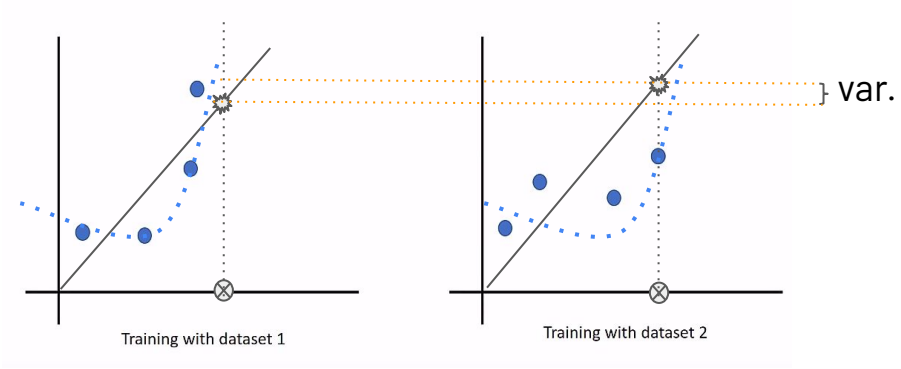
$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error\left(pred^{(i)}, \text{E}[pred^{(i)}]\right)]$$

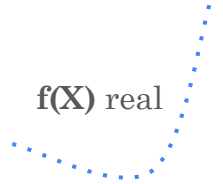
Sesgo y Varianza

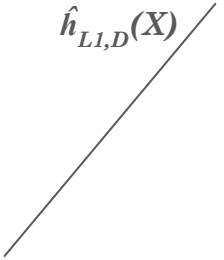
Visualización (regresión)

Algoritmo:
Regresión Lineal

Sesgo: **Alto (Underfit)**
Varianza: ??

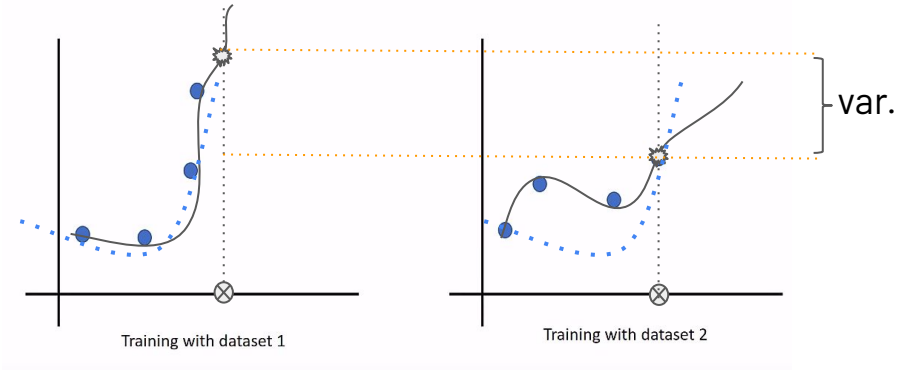


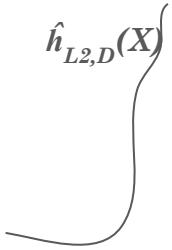




Algoritmo:
Regresión Polinómica

Sesgo: **Bajo**
Varianza: ??





$$\text{Sesgo}\left[pred^{(i)}\right] = \mathrm{E}\left[error\left(pred^{(i)}, f(x^{(i)})\right)\right]$$

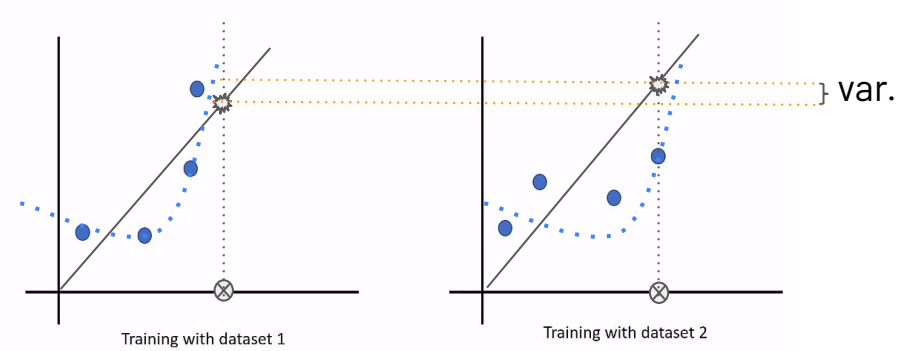
$$\text{Var}\left[pred^{(i)}\right] = \mathrm{E}[error\left(pred^{(i)}, \mathrm{E}[pred^{(i)}]\right)]$$

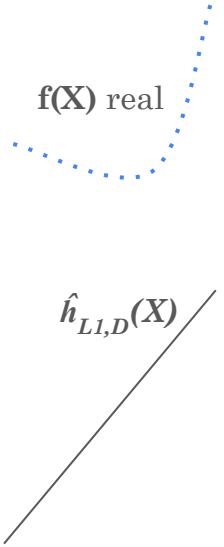
Sesgo y Varianza

Visualización (regresión)

Algoritmo:
Regresión Lineal

Sesgo: **Alto (Underfit)**
Varianza: **Baja**

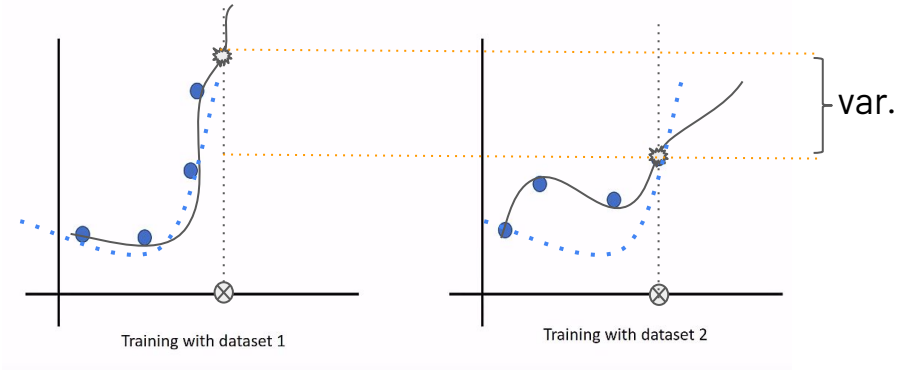


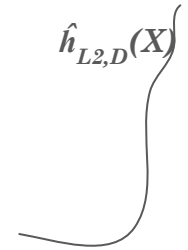


Notar que la varianza **no depende** de **f**.

Algoritmo:
Regresión Polinómica

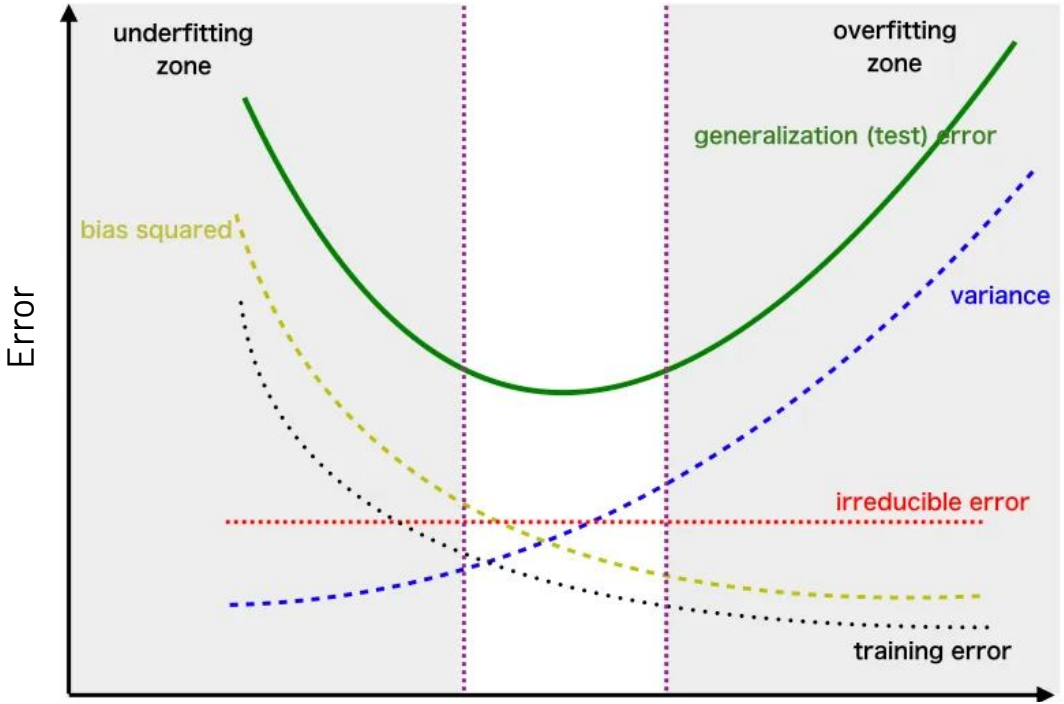
Sesgo: **Bajo**
Varianza: **Alta (Overfit)**





Sesgo y Varianza

Visualización (una más)



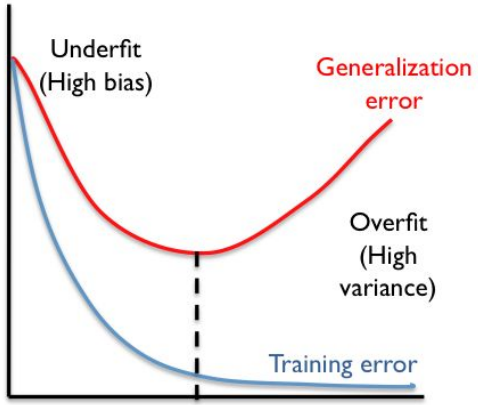
Fuente: https://towardsdatascience.com/the-bias-variance-tradeoff-8818f41e39e9

Herramientas de Diagnóstico

Sesgo y Varianza

Herramientas de Diagnóstico

Herramienta 1: curvas de complejidad



Herramienta 2: curvas de aprendizaje

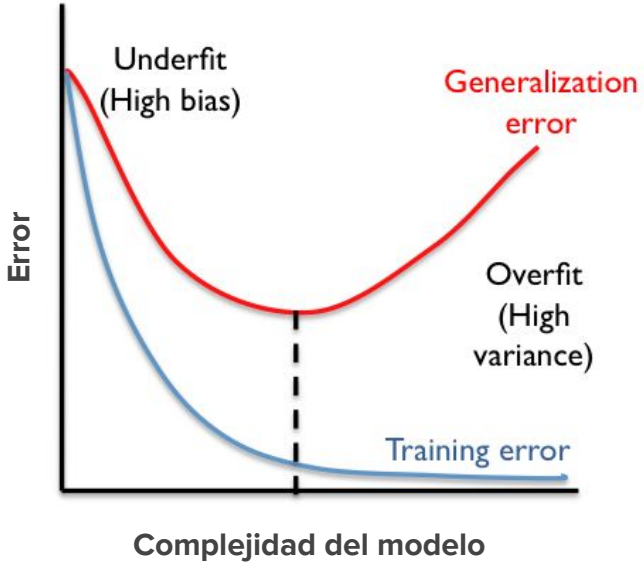
¿Por qué? para poder contestar:

¿Servirá extraer más atributos?
¿Convendrá recolectar más datos?
¿Pruebo con otros hiperparámetros, cuáles?
¿Cambio de algoritmo, a uno más simple?

Curvas de complejidad

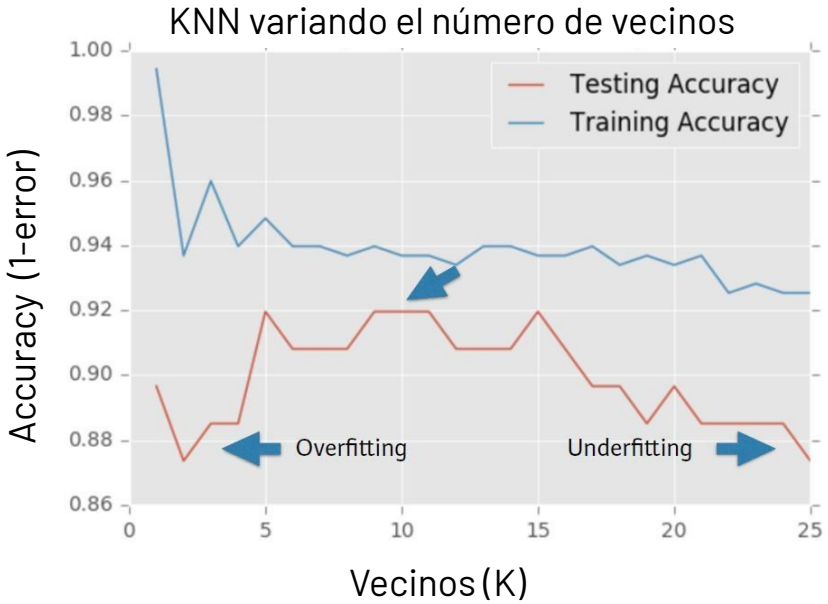
Herramientas de diagnóstico

Curvas de Complejidad del Modelo



Procedimiento:
Medir el error de entrenamiento y validación a medida que variamos hiperparámetros del algoritmo.

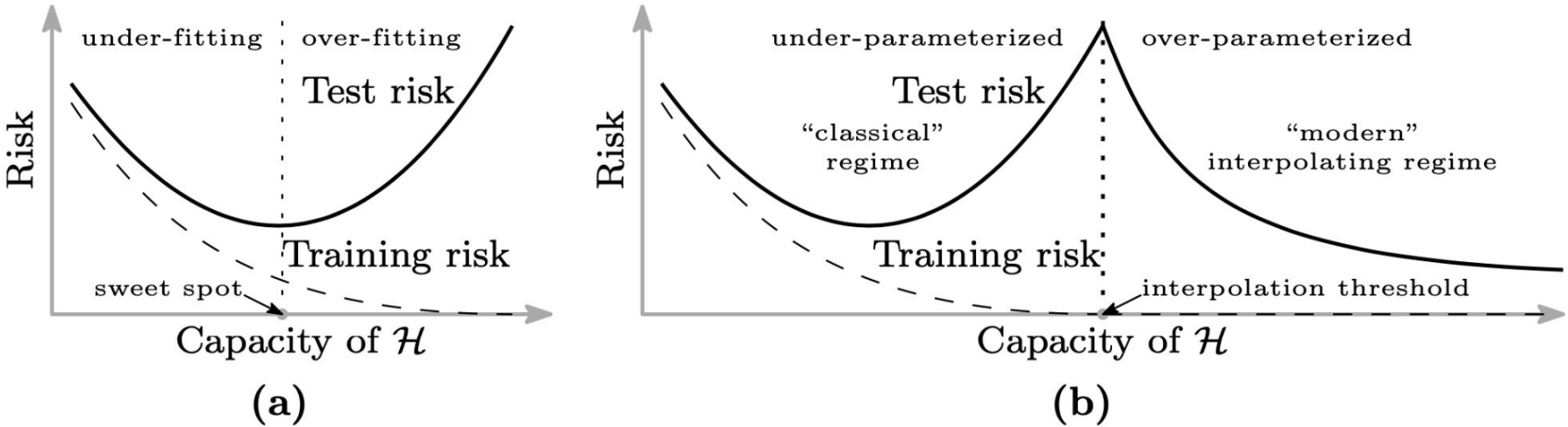
Ejemplo:



Herramientas de diagnóstico

Curvas de Complejidad del Modelo

[Belkin, M., Hsu, D., Ma, S., & Mandal, S. (**2019**). **Reconciling modern machine-learning practice and the classical bias–variance trade-off**. *Proceedings of the National Academy of Sciences*, *116*(32), 15849-15854.]

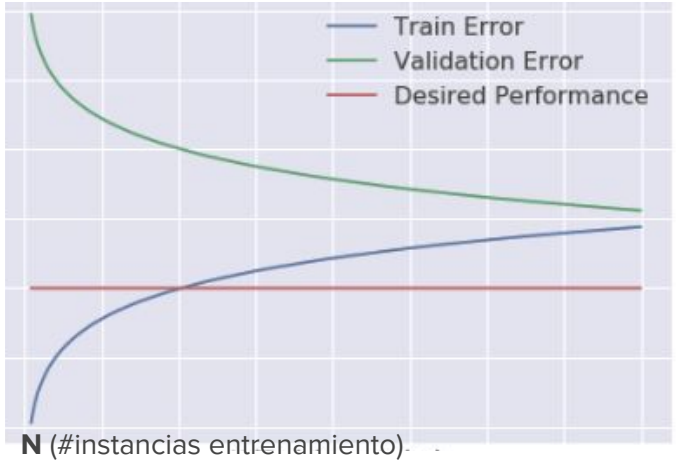


Si les interesa este tema, ver también la Sección 10.8 ISLR

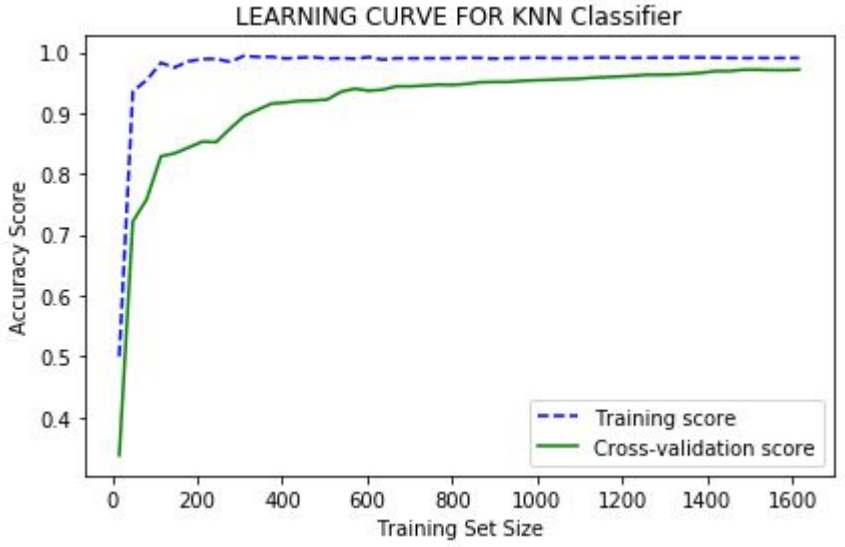
Curvas de aprendizaje

Herramientas de diagnóstico

Curvas de aprendizaje

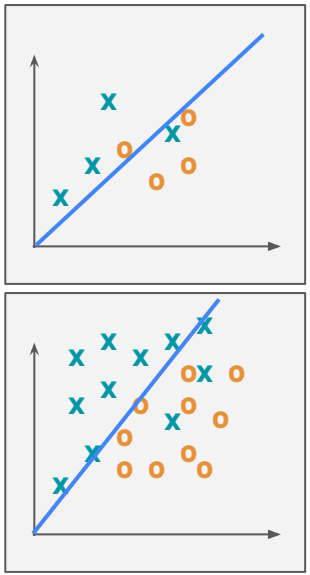


Procedimiento:
Medir el error de entrenamiento y validación a medida que cambiamos la cantidad de datos de entrenamiento.
Atención 1: Manteniendo siempre el mismo conjunto de validación.
Atención 2: Incrementando el train set de manera acumulativa.



Sesgo y Varianza

Herramientas de Diagnóstico



Ejemplo 1
LDA por ejemplo

Alto sesgo. No captura patrones complejos.

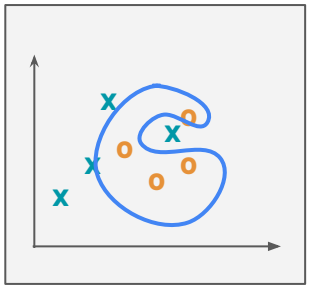
Baja varianza. Poca variación aunque cambiemos los datos.

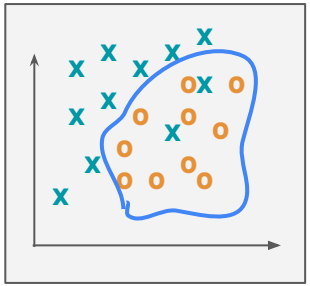
¿Servirá extraer más atributos?
¿Convendrá recolectar más datos?

Curvas de aprendizaje

Sesgo y Varianza

Herramientas de Diagnóstico





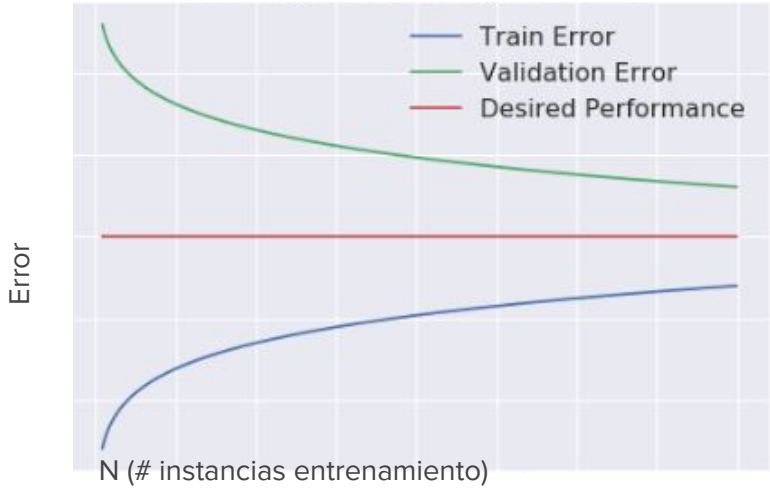
Ejemplo 2
Algoritmo más complejo

Bajo sesgo. En promedio encontrará el patrón adecuado.

Alta varianza. Mucha variación ante cambios en los datos.

¿Servirá extraer más atributos?
¿Convendrá recolectar más datos?

Curvas de aprendizaje



Sesgo y Varianza

Herramientas de Diagnóstico

Opciones para disminuir la varianza

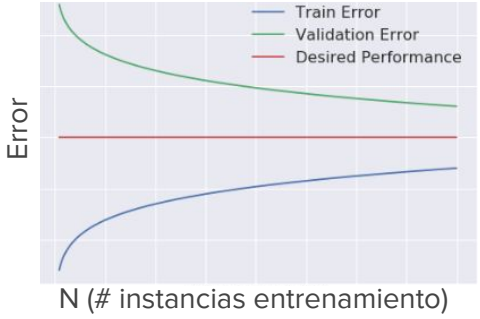
- Seleccionar modelos simples
- Reducción dimensional
- Regularización (pruning, lasso, ridge, etc)
- Usar algunos **ensambles**

Opciones para disminuir el bias

- Seleccionar modelos más complejos
- Extraer más features
- Usar otros **ensambles**

Curvas de aprendizaje

Alto Sesgo

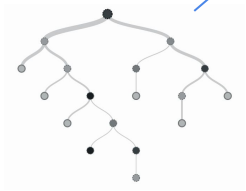


Alta Varianza

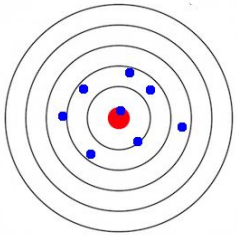
Sesgo y Varianza

Un método: Ensambles

Regla de pulgar:



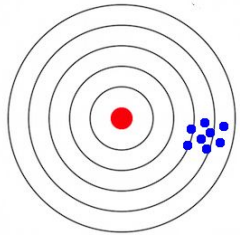
Overfit



“Bagging” Methods

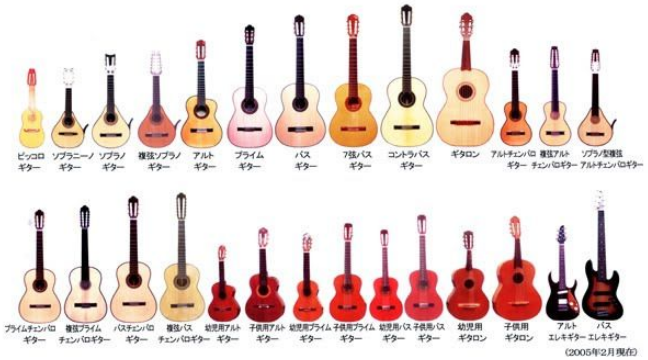
Ensambles

Underfit



“Boosting” Methods

Ensambles



Bagging

Random Forest

Bagging

$$\text{Var}\left[pred^{(i)}\right] = \text{E}[error(pred^{(i)}, \text{E}[pred^{(i)}])]$$

Ensambles

Bagging

Dado un conjunto de $\mathbf{B}$ variables aleatorias i.i.d. (independientes e idénticamente distribuidas) $\mathbf{Z_1, \ldots, Z_B}$, cada una con varianza $\boldsymbol{\sigma^2}$: La varianza de la media $\bar{\mathbf{Z}}$ de las observaciones está dada por

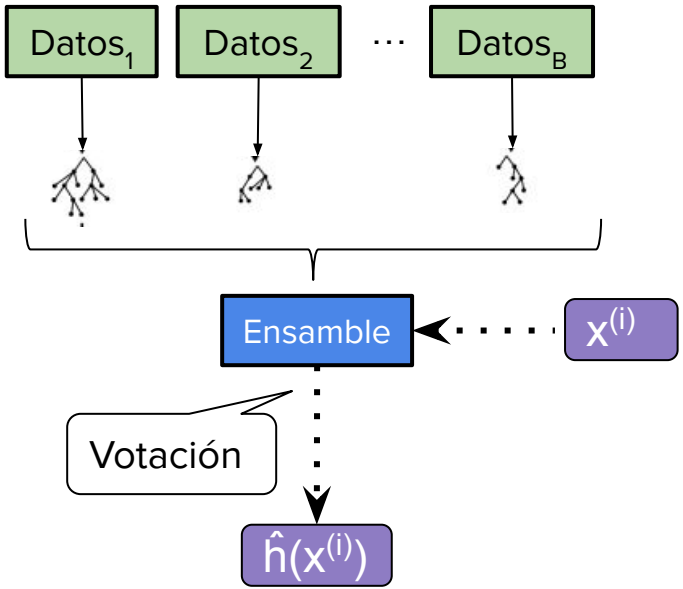
$$\text{Var}(\bar{\mathbf{Z}}) = \boldsymbol{\sigma^2/B}.$$

Surge la idea de **promediar** estimadores para **reducir** su varianza**.**

En otros términos, hacer que $\mathbf{pred^{(i)}}$ tienda a $\mathbf{E[pred^{(i)}]}$

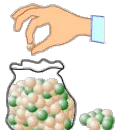
¿Cómo?

- Tomar muchas training sets distintos
- Construir un modelo predictivo distinto por cada set
- Promediar las predicciones resultantes
- $\hat{\mathbf{h}}(\mathbf{x}) = \mathbf{combinar}(\hat{\mathbf{h}}_{\mathbf{D\_1}}(\mathbf{x}), \hat{\mathbf{h}}_{\mathbf{D\_2}}(\mathbf{x}), \ldots., \hat{\mathbf{h}}_{\mathbf{D\_B}}(\mathbf{x}))$
- ¿Es práctico? (¿de dónde sacamos B datasets?)



Ensambles

Bagging (**B**ootstrap **Agg**regat**ing**)



Método de Bootstrap (1979).

Dado un dataset, crear otros del mismo tamaño con instancias (filas) elegidas al azar (con reposición)

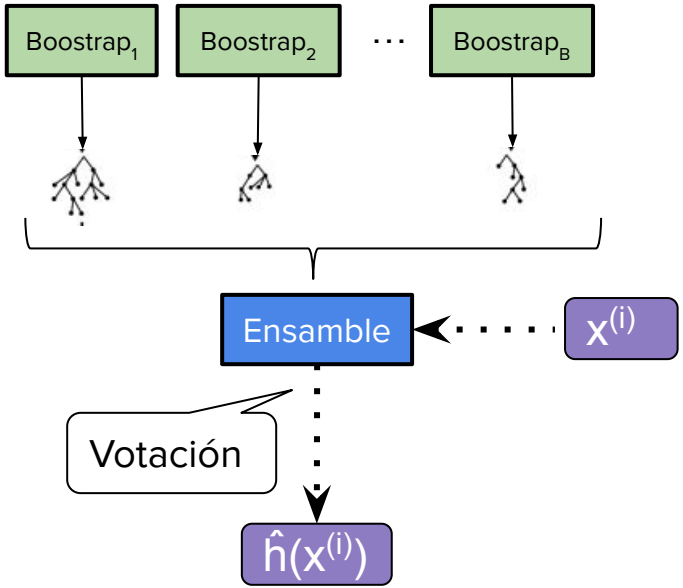
Data: $x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}, x^{(7)}, x^{(8)}$

Bootstrap$_1$: $x^{(7)}, x^{(6)}, x^{(2)}, x^{(6)}, x^{(5)}, x^{(2)}, x^{(1)}, x^{(1)}$

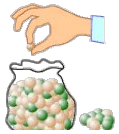
Bootstrap$_2$: $x^{(2)}, x^{(6)}, x^{(5)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(1)}$

Bootstrap$_3$: $x^{(3)}, x^{(3)}, x^{(1)}, x^{(5)}, x^{(7)}, x^{(6)}, x^{(2)}, x^{(8)}$

Bagging



"No todos los estimadores pueden mejorarse modificando los datos de esta manera. Parece que los estimadores altamente no lineales, como los árboles, son los que más se benefician. "(Elements of Statistical Learning)



Ensambles

Bagging (**B**ootstrap **Agg**regat**ing**)

Método de Bootstrap (1979).

Dado un dataset, crear otros del mismo tamaño con instancias (filas) elegidas al azar (con reposición)

Data: $x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}, x^{(7)}, x^{(8)}$

Bootstrap$_1$: $x^{(7)}, x^{(6)}, x^{(2)}, x^{(6)}, x^{(5)}, x^{(2)}, x^{(1)}, x^{(1)}$

Bootstrap$_2$: $x^{(2)}, x^{(6)}, x^{(5)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(1)}$

Bootstrap$_3$: $x^{(3)}, x^{(3)}, x^{(1)}, x^{(5)}, x^{(7)}, x^{(6)}, x^{(2)}, x^{(8)}$

¿Qué tan parecidos son entre sí?

P(elegir el mismo elem en dos dataset) =

$$1 - \left(1 - \frac{1}{n}\right)^n \approx 0.632$$

Problema con Bagging: Los árboles están muy correlacionados ¿Por qué? (2 motivos)

¿Y cómo afecta este problema?

Dado un conjunto de $\mathbf{B}$ variables aleatorias i.d. (idénticamente distribuidas, pero no necesariamente independientes) $\mathbf{Z_1, \ldots, Z_n}$, cada una con varianza $\boldsymbol{\sigma^2}$ con correlación entre pares positiva ρ:

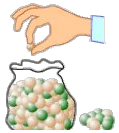
La varianza de la media $\mathbf{Z}$ de las observaciones está dada por:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

A medida que B crece, el segundo término desaparece, pero no así el primero.

Ensambles

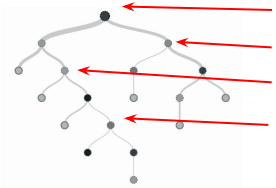
Random Forest



Random Forest

- Bagging + Features al azar **por nodo.**
- **Para cada nodo**, elegimos al azar **m** features para considerar en el "atributo que mejor separe".
- En general funciona $m \approx sqrt(p)$ (p = #features), también $m \approx log_2(p)$ aunque se puede ir tan bajo como $m = 1$

(Bootstrap)



(m=3)

Mejor corte entre: x_2, x_4, x_{16}
Mejor corte entre: x_3, x_5, x_6
Mejor corte entre: x_1, x_{32}, x_6
Mejor corte entre: x_{32}, x_{41}, x_{63}

Algoritmos de la familia "Bagging"

Bagging [Breiman 1994]
Subsets basado en filas al azar (con reposición)

Random Subspaces (Feature Bagging) [Ho 1998]
Subsets basados en columnas al azar

Pasting [Breiman 1999]
Subsets basado en filas al azar. Los votos de los clasificadores se ponderan según su capacidad predictiva en un conjunto de validación.

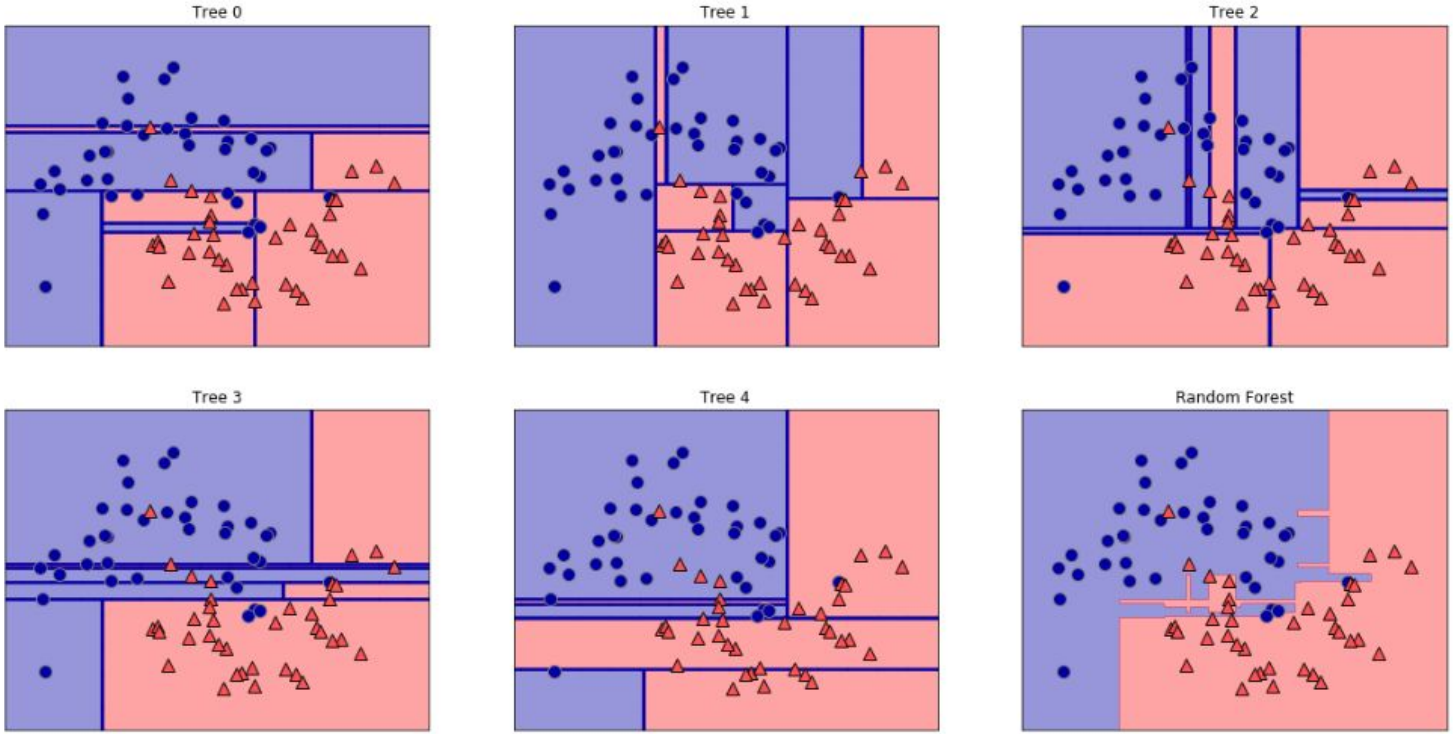
Random Forest [Breiman 2001]
[Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.]

Extremely Randomized Trees [Geurts 2006]
Random forest + aleatorización en el mejor corte

Random Patches [Louppe 2012]
Subsets basados en filas y columnas al azar.

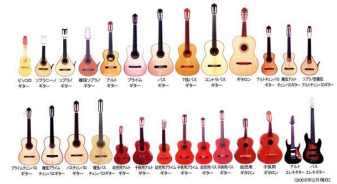
Ensambles

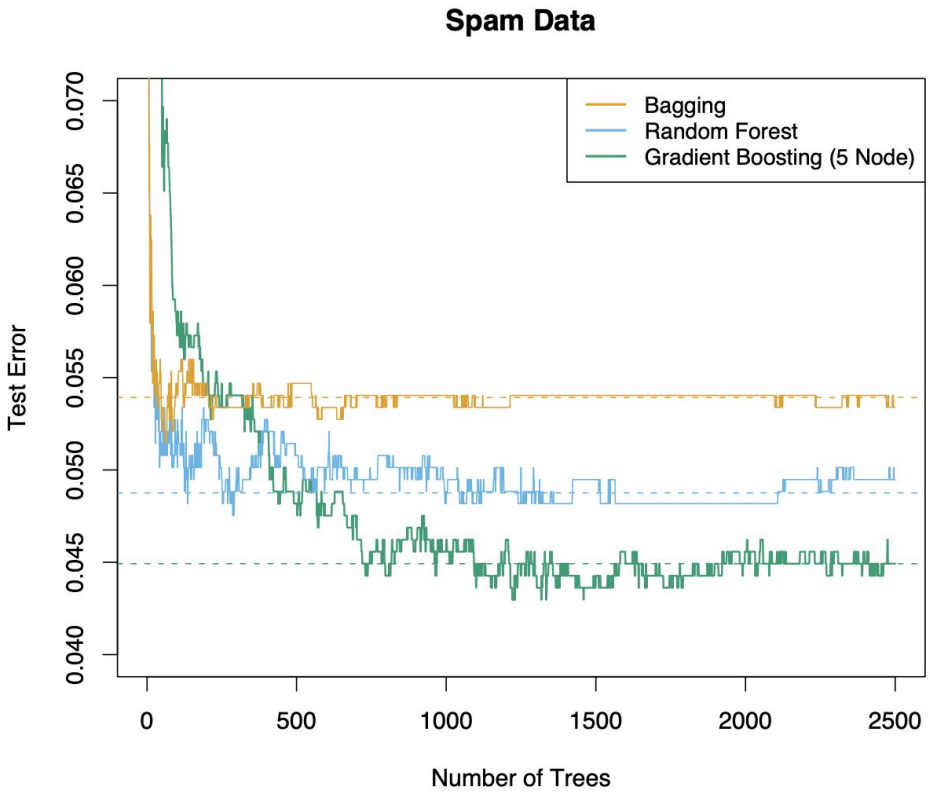
Random Forest



Ensambles

Random Forest





[Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). ***The elements of statistical learning**: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.]

FIGURE 15.1. *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting, 5-node trees were used, and the number of trees were chosen by 10-fold cross-validation (2500 trees). Each "step" in the figure corresponds to a change in a single misclassification (in a test set of 1536).*

Ensambles

OOB Error

¿Podremos aprovechar las características de **Bagging** para obtener estimaciones realistas de la performance sin hacer cross-validation?

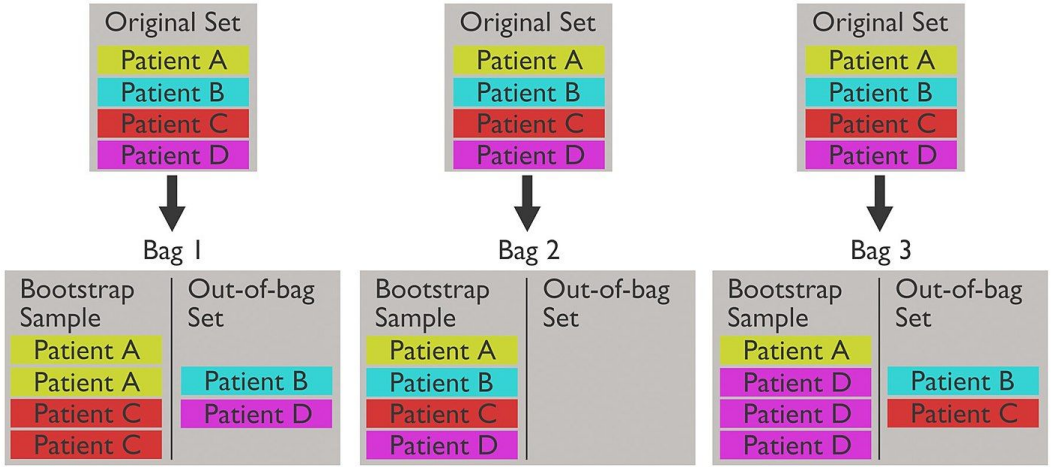
Out-of-Bag Error:

Cómo técnica para obtener estimaciones de qué tan bien generalizan los modelos.Idea: para cada instancia, utilizar los árboles que no contienen a $x^{(i)}$ en su conjunto de entrenamiento.

La ventaja del método OOB es que requiere menos computación y permite probar el modelo a medida que se entrena.

No reemplaza a Cross Validation:

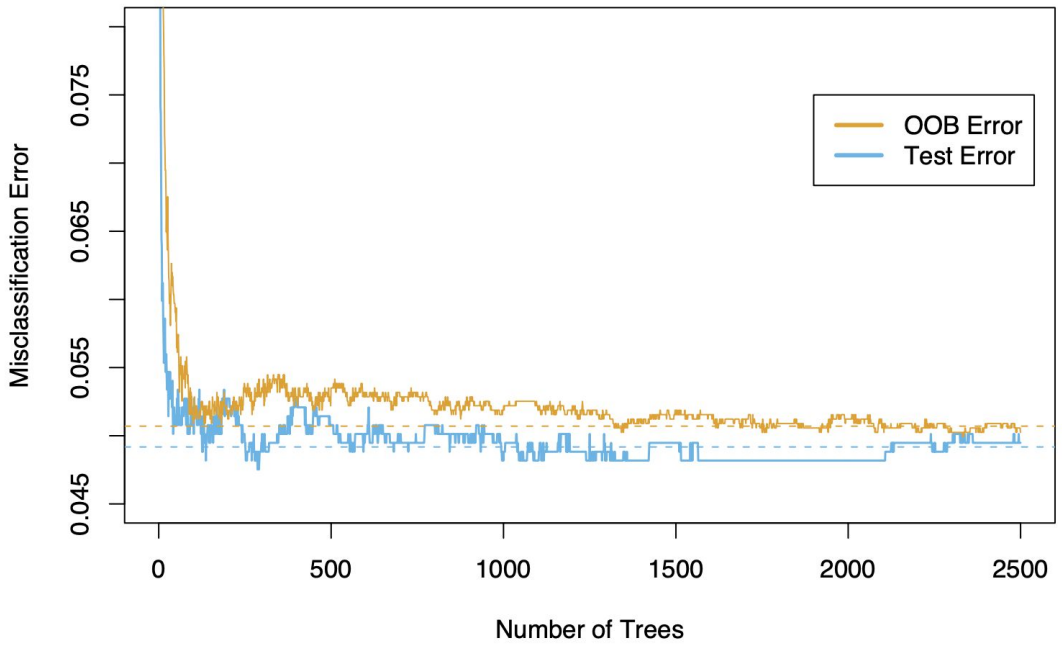
[Janitza, S., & Hornung, R. (2018). **On the overestimation of random forest's out-of-bag error**. PloS one, 13(8), e0201904.



https://en.wikipedia.org/wiki/Out-of-bag_error

Ensambles

OOB Error



[Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). ***The elements of statistical learning**: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.]

FIGURE 15.4. OOB *error computed on the* `spam` *training data, compared to the test error computed on the test set.*

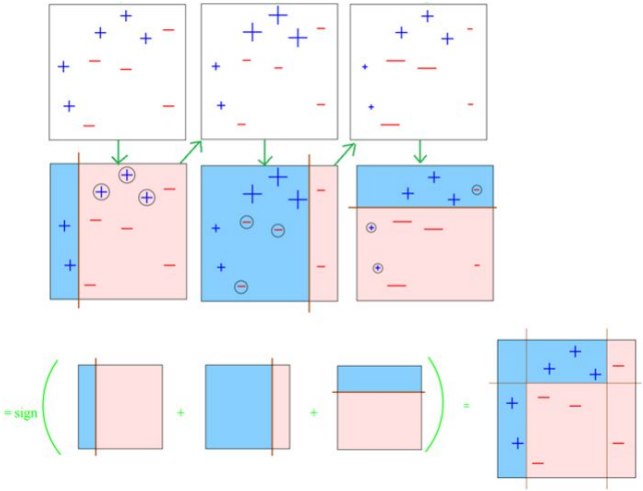
Boosting

Ensambles

Boosting

Estimadores son construidos de manera **secuencial**:

Idea, combinar clasificadores "débiles" (alto sesgo) de manera que cada clasificador se convierta en un "experto" en los errores que cometen los clasificadores anteriores.



Algoritmos de la familia "Boosting"

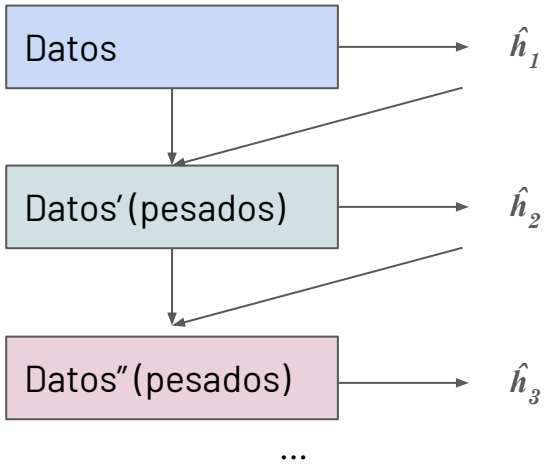
Ada Boost [Yoav 1997]
Cada instancia tiene un peso determinado según si el algoritmo pudo o no predecir bien su valor en árboles anteriores.

Gradient Boosting [Friedman 1999]
Generalización del anterior a cualquier función de costo diferenciable

eXtreme Gradient Boosting (XGBoost) [Chen 2016]
Implementación eficiente de Gradient Boosting + regularización en los nuevos árboles
(Algoritmo ganador en competencias de Kaggle)

Ensambles

Boosting: AdaBoost



$$\hat{h}(x) = \text{combinación pesada } \hat{h}_t$$

AdaBoost [Yoav 1997]

Predictores en secuencia (malo para la paralelización), de tal manera que el segundo ajuste bien lo que el primero no ajustó, que el tercero ajuste un poco mejor lo que el segundo no pudo ajustar y así sucesivamente.

[Yoav 1997] Freund, Yoav; Schapire, Robert E (1997). "A decision-theoretic generalization of on-line learning and an application to boosting". Journal of Computer and System Sciences.

Ensambles

Boosting: AdaBoost

Algorithm 1: Algoritmo AdaBoost

Data: $(x^{(1)}; y^{(1)}), ..., (x^{(n)}; y^{(n)}), x^{(i)} \in X, y^{(i)} \in Y = \{-1, 1\}$

Inicializar: $D_1[i] = \frac{1}{n}$ para i = 0..n

for $t = 1$ *to* T **do**

1) Entrenar un clasificador $h_t : X \to \{-1, 1\}$ débil tomando en cuenta los pesos D_t;
2) Computar el error ponderado $\epsilon_t = \sum_{i:h_t(x^{(i)}) \neq y^{(i)}} D_t[i]$;
3) Elegir el ponderador $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
4) Actualizar $D_{t+1}[i] = \frac{D_t[i]}{Z_t} e^{(-\alpha_t y^{(i)} h_t(x^{(i)}))}$ para todo i, donde Z_t es una constante de normalización que logra que $sum(D_{t+1}) = 1$;

Output: $\mathrm{h(x)} = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

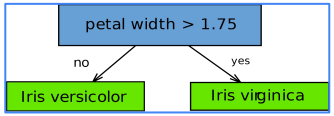
Vector de pesos para cada observación

Sumamos sólo los pesos de las instancias mal clasificadas

Si el error es grande, α chico.

El nuevo peso será grande si el predictor anterior fue muy bueno en general pero erró para esa instancia

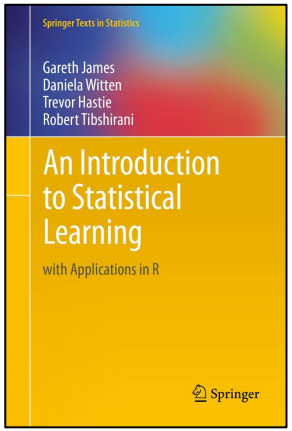
Notar que este algoritmo es una meta heurística. Los h pueden ser árboles, LDA, SVM, etc). Se suelen utilizar árboles "decision stump"



TAREA

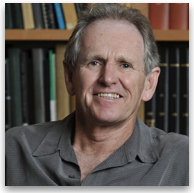
- Leer **sección 8.2.1 Bagging y 8.2.2 Random Forest** del ISLR
- Leer (al menos) **la introducción del paper y las definiciones de la sección 2 del paper:**
  Domingos, Pedro A unified bias-variance decomposition (y tanto como puedan de los teoremas que se plantean)
- Opcional (para la gente valiente, leer lo que puedan de...)
  - **Capítulo 15 Random Forests** del ESLI
  - **Capítulo 10 Boosting and Additive Trees** del ESLI
- **Completar el cuestionario (no hay notebook esta semana, sí cuestionario y guía de ejercicios)**

ISLR



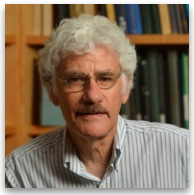
Gareth James

Daniela Witten



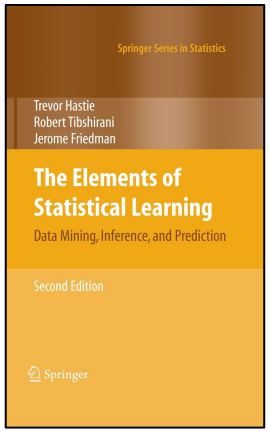
Trevor Hastie

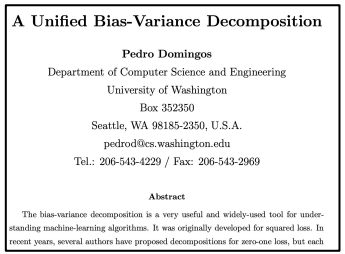
Robert Tibshirani



Jerome Friedman

ESLI




A Unified Bias-Variance Decomposition

Pedro Domingos
Department of Computer Science and Engineering
University of Washington
Box 352350
Seattle, WA 98185-2350, U.S.A.
pedrod@cs.washington.edu
Tel.: 206-543-4229 / Fax: 206-543-2969

Abstract

The bias-variance decomposition is a very useful and widely-used tool for understanding machine-learning algorithms. It was originally developed for squared loss. In recent years, several authors have proposed decompositions for zero-one loss, but each

Pedro Domingos