

Ensemble Learning

Dra Ana Georgina Flesia

Diplomatura en Ciencia de Datos 2021
FaMAF-UNC
Oficina 370
georgina.flesia@unc.edu.ar

2021

Contenidos:

- ▶ Modelos de aprendizaje
- ▶ Majority voting
- ▶ Bagging
- ▶ Boosting
- ▶ Random Forests

Modelo de Perdida (Error)

- ▶ Pérdida cuadrática en el caso de prueba y_i es

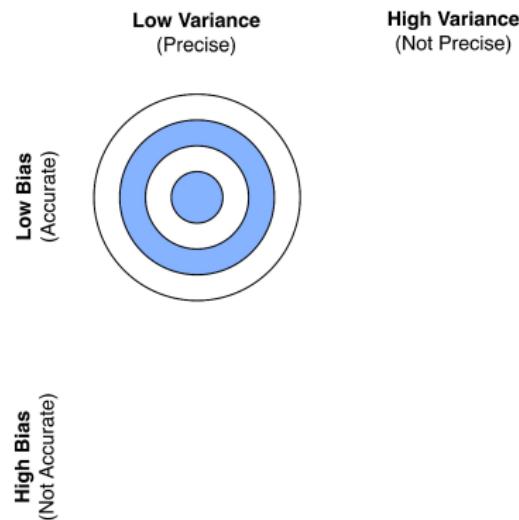
$$S = (y_i - \hat{y}_i)^2$$

- ▶ Error cuadrático medio

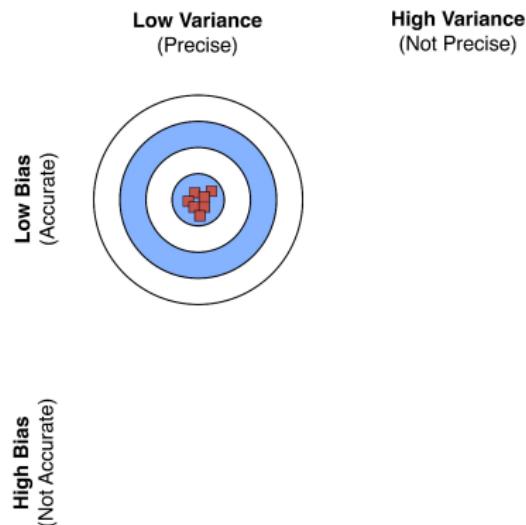
$$\begin{aligned} E[S] &= E[(y - \hat{y})^2] \\ &= (y - E[\hat{y}])^2 + E[(E[\hat{y}] - \hat{y})^2] \\ &= [\text{Bias}]^2 + \text{Variance} \end{aligned}$$

- ▶ Sobre un conjunto de test $D = \{y_1, \dots, y_n\}$, las esperanzas se estiman como los promedios sobre la muestra de test.
 - Desvío (Bias) es el error promedio cuadrado dado por el diferencia con el modelo
 - Varianza es una medida de la variación producida por la aleatoriedad de la muestra.

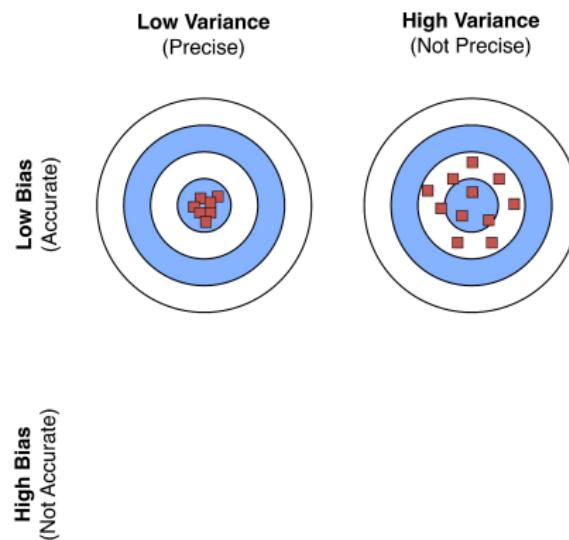
Ensemble methods



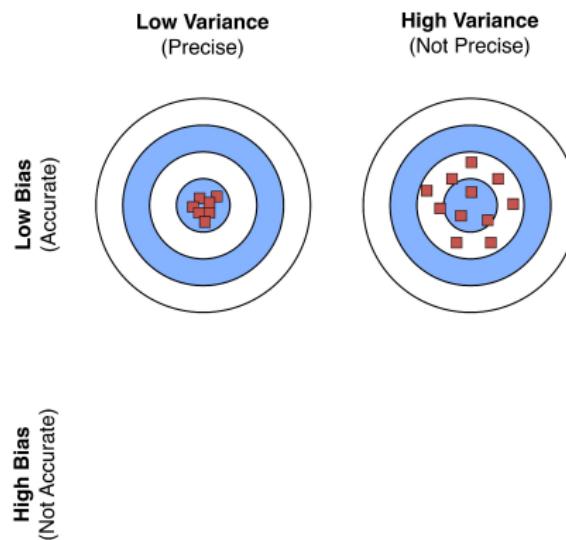
Ensemble methods



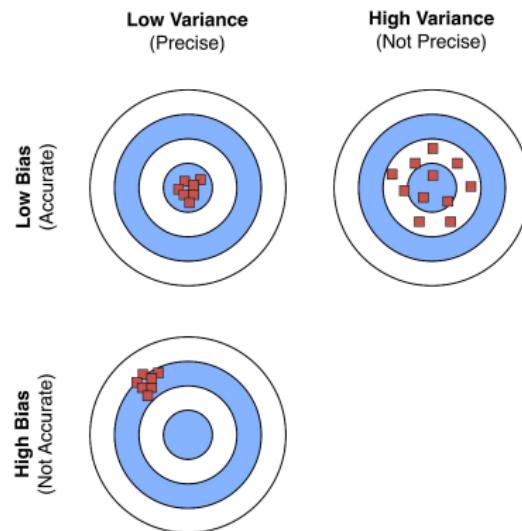
Ensemble methods



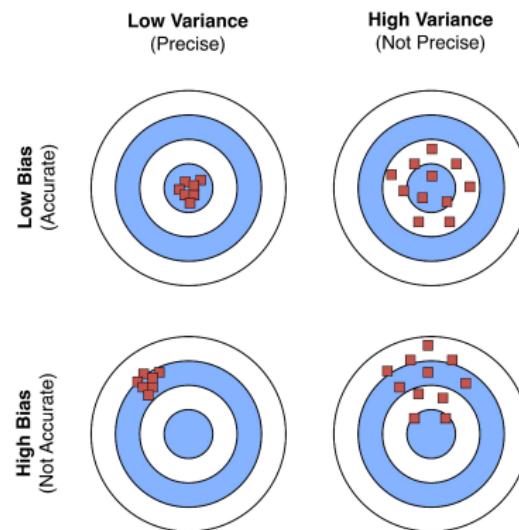
Ensemble methods



Ensemble methods



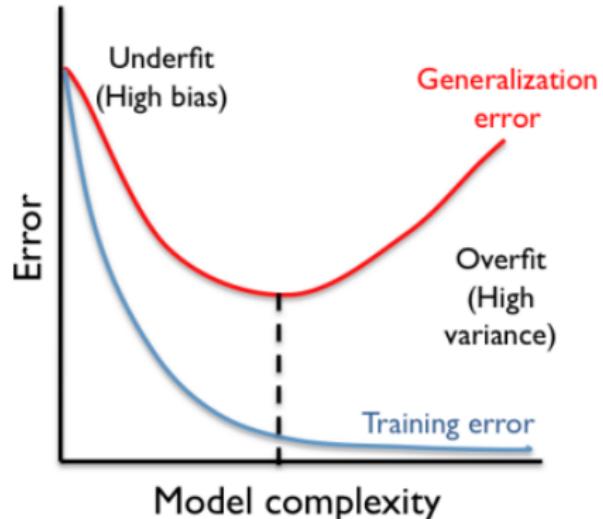
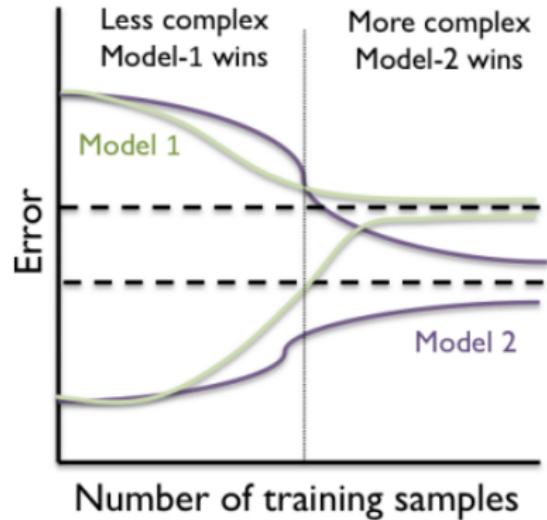
Ensemble methods



Capacidad del Modelo

- ▶ Underfitting: los errores de entrenamiento y testeo son ambos grandes. Sesgo alto.
- ▶ Overfitting: error al entrenar es pequeño, pero al testear aumenta. Varianza alta.
- ▶ Si el espacio de hipótesis que se estudia es grande hay mas tendencia a sobreajustar.

Sobre-entrenamiento y Sub-entrenamiento



Descomposición en sesgo y varianza

- ▶ La descomposición de la pérdida en sesgo y varianza nos ayuda a comprender algoritmos de aprendizaje, los conceptos se relacionan con el ajuste y el sobreajuste
- ▶ Sabemos que el promedio de un estimador reduce su varianza

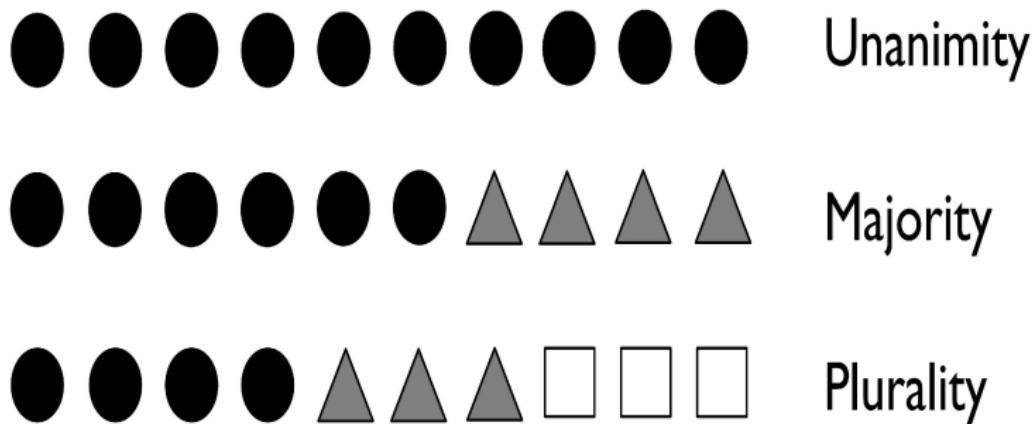
$$Var(\bar{y}) = \frac{Var(\hat{y})}{N}$$

- ▶ podemos promediar modelos? reduciría esto la varianza del modelo resultante?

Contenidos:

- ▶ Modelos de aprendizaje
- ▶ Majority voting
- ▶ Bagging
- ▶ Boosting
- ▶ Random Forests

Majority voting

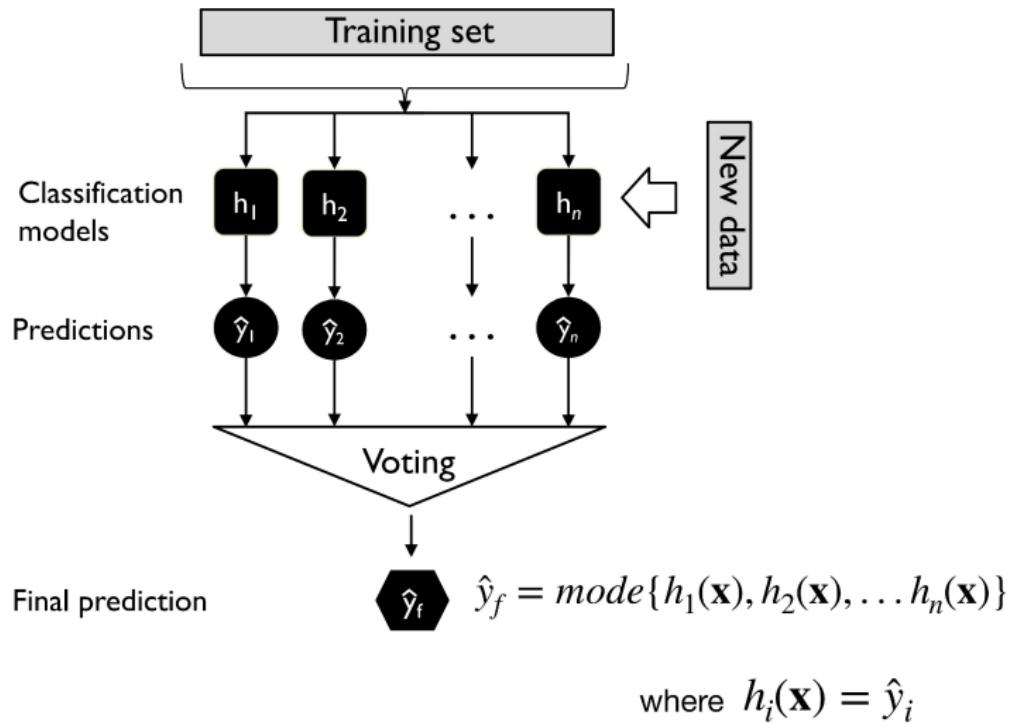


Clasificador Majority vote

Si se tienen n clasificadores diferentes $\{h_1, \dots, h_n\}$, el clasificador del voto de la mayoría es la moda

$$\hat{y}_f = \text{mode} \{h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_n(\mathbf{x})\}$$

Clasificador Majority vote



Clasificador Majority vote caso binario

Si se asumen las siguientes condiciones

- ▶ se tienen n clasificadores independientes $\{h_1, \dots, h_n\}$ con una tasa de error base ϵ
- ▶ independientes significa que los errores no están correlacionados
- ▶ realizan una tarea de clasificación binaria
- ▶ la tasa de error es mejor que adivinar al azar (es decir, inferior a 0,5 para clasificación binaria)

$$\forall \epsilon_i \in \{\epsilon_1, \epsilon_2, \dots, \epsilon_n\}, \epsilon_i < 0,5$$

Clasificador Majority vote caso binario

La probabilidad de que cometamos una predicción errónea con el ensemble si k clasificadores predicen la misma etiqueta de clase es

$$P(k) = \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k} \quad k > \lceil n/2 \rceil$$

donde $\binom{n_k}{k}$ el el coeficiente binomial

$$\binom{n}{k} = \frac{n!}{(n - k)!k!}$$

Error de ensemble:

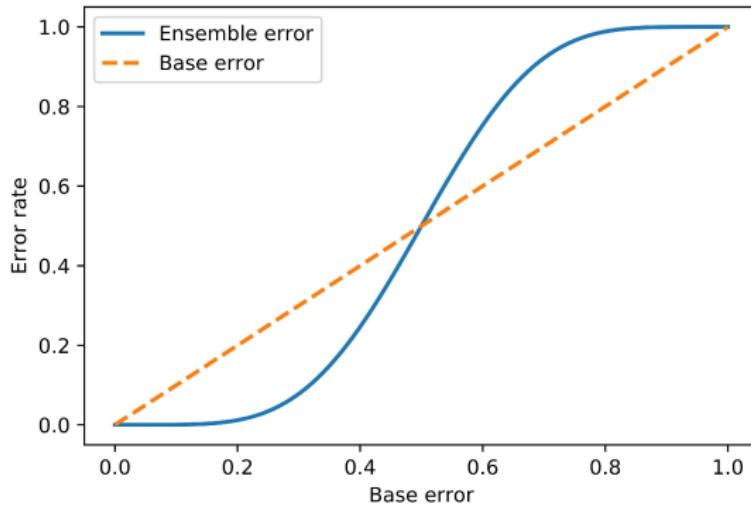
$$\epsilon_{ens} = \sum_k^n \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k}$$

$$\epsilon_{ens} = \sum_{k=6}^{11} \binom{11}{k} 0,25^k (1 - 0,25)^{11-k} = 0,034$$

si $n = 11$ y $\epsilon = 0,25$

Clasificador Majority vote caso binario

$$\epsilon_{ens} = \sum_k \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k}$$



Clasificador Soft Majority vote

Sean $p_{i,j}$ la probabilidad de pertenencia a la clase predicha de la etiqueta j y el clasificador i , y w_i un peso opcional.

$$\hat{y} = \arg \max_j \sum_{i=1}^n w_i p_{i,j}$$

es el clasificador soft majority vote.

Clasificador Soft Majority vote

El siguiente ejemplo binario

$$\hat{y} = \arg \max_j \sum_{i=1}^n w_i p_{i,j}$$

$$j \in \{0, 1\} \quad h_i (i \in \{1, 2, 3\})$$

$$h_1(\mathbf{x}) \rightarrow [0, 9, 0, 1]$$

$$h_2(\mathbf{x}) \rightarrow [0, 8, 0, 2]$$

$$h_3(\mathbf{x}) \rightarrow [0, 4, 0, 6]$$

$$p(j = 0 | \mathbf{x}) = 0,2 \cdot 0,9 + 0,2 \cdot 0,8 + 0,6 \cdot 0,4 = 0,58$$

$$p(j = 1 | \mathbf{x}) = 0,2 \cdot 0,1 + 0,2 \cdot 0,2 + 0,6 \cdot 0,6 = 0,42$$

$$\hat{y} = \arg \max_j \{p(j = 0 | \mathbf{x}), p(j = 1 | \mathbf{x})\}$$

Contenidos:

- ▶ Modelos de aprendizaje
- ▶ Majority voting
- ▶ Bagging
- ▶ Boosting
- ▶ Random Forests

Métodos de Evaluación

- ▶ Bagging (bootstrap aggregating) se basa en un concepto similar al voto mayoritario pero utiliza el mismo algoritmo de aprendizaje (generalmente un algoritmo de árbol de decisión) para ajustar modelos en diferentes subconjuntos de datos de entrenamiento (muestras de bootstrap).
- ▶ Bagging puede mejorar la precisión de los modelos inestables que tienden a sobreajustarse.

Algoritmo

Algorithm 1 Bagging

- 1: Let n be the number of bootstrap samples
 - 2:
 - 3: **for** $i=1$ to n **do**
 - 4: Draw bootstrap sample of size m , \mathcal{D}_i
 - 5: Train base classifier h_i on \mathcal{D}_i
 - 6: $\hat{y} = mode\{h_1(\mathbf{x}), \dots, h_n(\mathbf{x})\}$
-

Muestreo Bootstrap

Original Dataset

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------

Bootstrap 1

x_8	x_6	x_2	x_9	x_5	x_8	x_1	x_4	x_8	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_{10}
-------	-------	----------

Bootstrap 2

x_{10}	x_1	x_3	x_5	x_1	x_7	x_4	x_2	x_1	x_8
----------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_6	x_9
-------	-------

Bootstrap 3

x_6	x_5	x_4	x_1	x_2	x_4	x_2	x_6	x_9	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_8	x_{10}
-------	-------	-------	----------

Training Sets

Muestreo Bootstrap

Si tomamos muestras de una distribución uniforme, podemos calcular la probabilidad de que un ejemplo específico de un conjunto de datos de tamaño n no sea elegido como parte de la muestra bootstrap

$$P(\text{not chosen}) = \left(1 - \frac{1}{n}\right)^n$$

lo cual es asintoticamente equivalente a

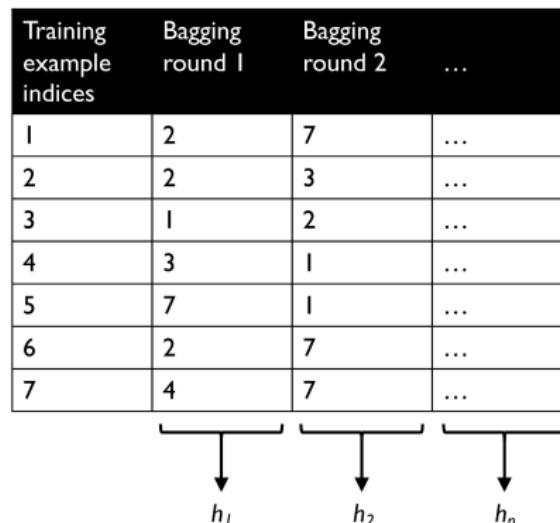
$$\frac{1}{e} \approx 0,368 \quad \text{as} \quad n \rightarrow \infty$$

Muestreo Bootstrap

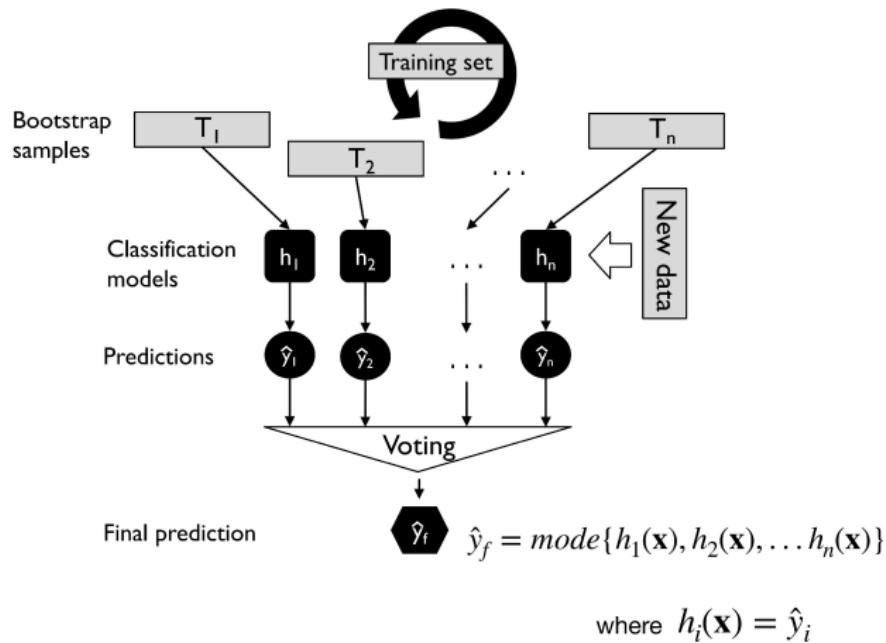
$$P(\text{ not chosen }) = \left(1 - \frac{1}{n}\right)^n$$
$$\frac{1}{e} \approx 0,368, \quad n \rightarrow \infty$$
$$P(\text{ chosen }) = 1 - \left(1 - \frac{1}{n}\right)^n \approx 0,632$$

Por lo cual la proporción de elementos no repetidos de la muestra converge a 63

Muestreo Bootstrap



Bagging Classifier



Porque funciona?

- ▶ Cada árbol individual tiene gran variabilidad , y una tendencia a al overfitting.
- ▶ Un modelo de bagging tiene menor varianza que cada árbol individual y por lo tanto menor overfitting

Contenidos:

- ▶ Modelos de aprendizaje
- ▶ Majority voting
- ▶ Bagging
- ▶ Boosting
- ▶ Random Forests

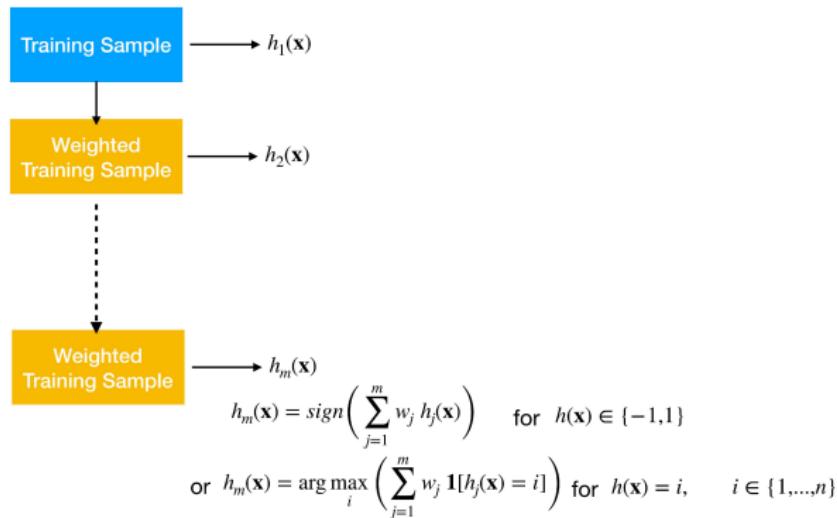
Boosting

- ▶ Hay dos categorías de boosting: Adaptive boosting y gradient boosting.
- ▶ Adaptive and gradient boosting se basan en el mismo concepto de mejoramiento de clasificadores débiles, (como decision tree stumps) en clasificadores fuertes.
- ▶ Boosting es un proceso iterativo donde el conjunto de entrenamiento se pesa en cada iteración de acuerdo a los errores que hace el clasificador.
- ▶ Adaptive and gradient boosting, difieren en como se eligen los pesos y como se combinan los clasificadores débiles.

Adaptive Boosting

- ▶ AdaBoost es un algoritmo clásico , definido por Freund and Schapire in 1997.
- ▶ XGBoost, es una implementación muy eficiente de los algoritmos de gradient boosting .

Boosting general



AdaBoost

- ▶ Se inicializa el vector de pesos con pesos uniformes
- ▶ Loop:
 - Se aplica el clasificador débil a los ejemplos pesados(en vez de usar el training sample original, se sacan muestras bootstrap con probabilidad pesada)
 - Se incrementan los pesos de los datos mal clasificados
- ▶ Se aplica un voto pesado a los clasificadores entrenados

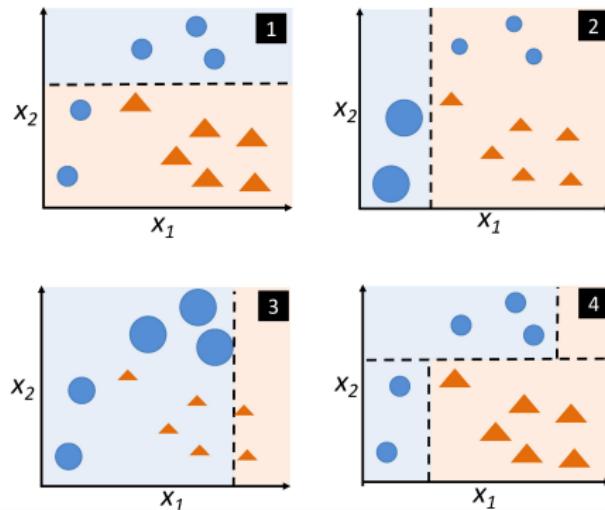
AdaBoost

Algorithm 1 AdaBoost

- 1: Initialize k : the number of AdaBoost rounds
 - 2: Initialize \mathcal{D} : the training dataset, $\mathcal{D} = \{\langle \mathbf{x}^{[1]}, y^{[1]} \rangle, \dots, \langle \mathbf{x}^{[n]}, y^{[n]} \rangle\}$
 - 3: Initialize $w_1(i) = 1/n, \quad i = 1, \dots, n, \quad \mathbf{w}_1 \in \mathbb{R}^n$
 - 4:
 - 5: **for** $r=1$ to k **do**
 - 6: For all $i : \mathbf{w}_r(i) := w_r(i) / \sum_i w_r(i)$ [normalize weights]
 - 7: $h_r := FitWeakLearner(\mathcal{D}, \mathbf{w}_r)$
 - 8: $\epsilon_r := \sum_i w_r(i) \mathbf{1}(h_r(i) \neq y_i)$ [compute error]
 - 9: if $\epsilon_r > 1/2$ then stop
 - 10: $\alpha_r := \frac{1}{2} \log[(1 - \epsilon_r)/\epsilon_r]$ [small if error is large and vice versa]
 - 11: $w_{r+1}(i) := w_r(i) \times \begin{cases} e^{-\alpha_r} & \text{if } h_r(\mathbf{x}^{[i]}) = y^{[i]} \\ e^{\alpha_r} & \text{if } h_r(\mathbf{x}^{[i]}) \neq y^{[i]} \end{cases}$
 - 12: Predict: $h_k(\mathbf{x}) = \arg \max_j \sum_r \alpha_r \mathbf{1}[h_r(\mathbf{x}) = j]$
 - 13:
-

AdaBoost

A decision stump es un modelo de ML correspondiente a un árbol de decisión de un nivel, esto es, un árbol con un nodo interno (la raíz) que se conecta inmediatamente con los nodos terminales (las hojas). A decision stump hace una predicción basado en el valor de una sola característica medida.



Contenidos:

- ▶ Modelos de aprendizaje
- ▶ Majority voting
- ▶ Bagging
- ▶ Boosting
- ▶ Random Forests

Random Forest

- ▶ Random forest (o random forests) es un clasificador de ensemble que consiste en calcular varios árboles de decisión y devolver la clase mas frecuente entre las salidas de los árboles, esto es, la moda de las salidas.
- ▶ El término proviene de random decision forests que fue propuesto by Tin Kam Ho of Bell Labs in 1995.
- ▶ El algoritmo combina el concepto de Breiman llamado "bagging" con la selección aleatoria de características.

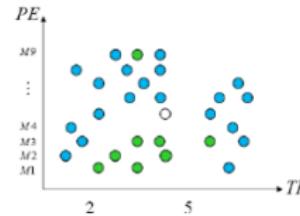
Random Forest

- ▶ Los clasificadores individuales a combinar son árboles de decisión, uno de los métodos de aprendizaje más populares para la exploración de datos.
- ▶ Un tipo de árbol de decisión es el CART, classification and regression tree.
- ▶ CART produce una partición voráz, binaria, de arriba hacia abajo, recursiva, que divide el espacio de características en conjuntos de regiones rectangulares disjuntas.
 - Las regiones deben ser puras con respecto a la variable de respuesta
 - Un modelo simple se aplica en cada región.

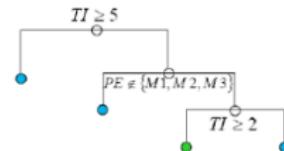
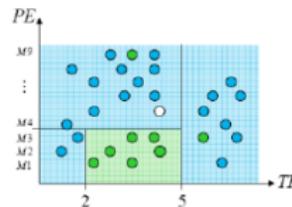
Random Forest

Dataset simple con dos predictores

TI	PE	Response
1.0	M2	good
2.0	M1	bad
**	**	**
4.5	M5	?



Partición recursiva voráz en TI y PE



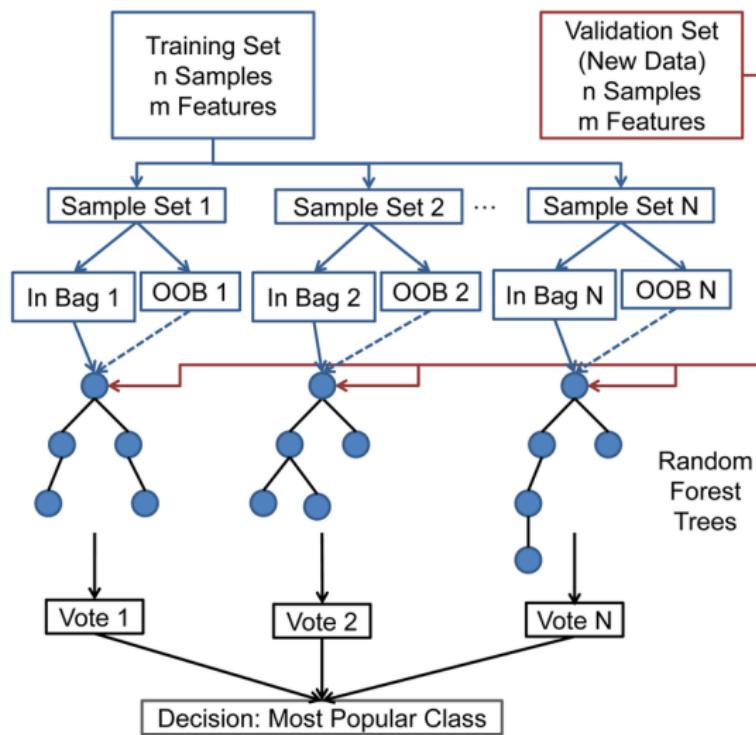
Random Forest

- ▶ Los random forests son algoritmos de aprendizaje automático muy utilizados debido a su buen rendimiento y su facilidad de uso, pues no precisa determinar constantes,
- ▶ En el contexto de bagging, random forest puede entenderse como un bagging de árboles de decisión que no son ajustados usando todas las características sino un subconjunto elegido aleatoriamente, y evaluado en un suconjunto bootstrap de los datos.

Random Forest: Algoritmo

- ▶ Cada árbol se construye utilizando el siguiente algoritmo:
 - Sea N el número de casos de entrenamiento, y M el número de variables en el clasificador.
 - Sabemos el número m de variables de entrada que se utilizarán para determinar la decisión en un nodo del árbol; m debería ser mucho menor que M .
 - Elija un conjunto de entrenamiento para este árbol eligiendo n veces con reemplazo de todos los N casos de entrenamiento disponibles (es decir, tome una muestra bootstrap).
 - Use el resto de los casos para estimar el error del árbol, prediciendo sus clases.
 - Para cada nodo del árbol, elija aleatoriamente m variables en las que basar la decisión en ese nodo. Calcule la mejor división basada en estas m variables en el conjunto de entrenamiento usando el índice de Gini.
 - Cada árbol está completamente desarrollado y no podado.
- ▶ Para la predicción, se aplica el árbol a la muestra y se le asigna la etiqueta del nodo hoja donde termina.
- ▶ Este procedimiento se repite en todos los árboles del conjunto, y el voto mayoritario de todos los árboles se informa como predicción de la random forest.

Random Forest



Estimación del error de prueba

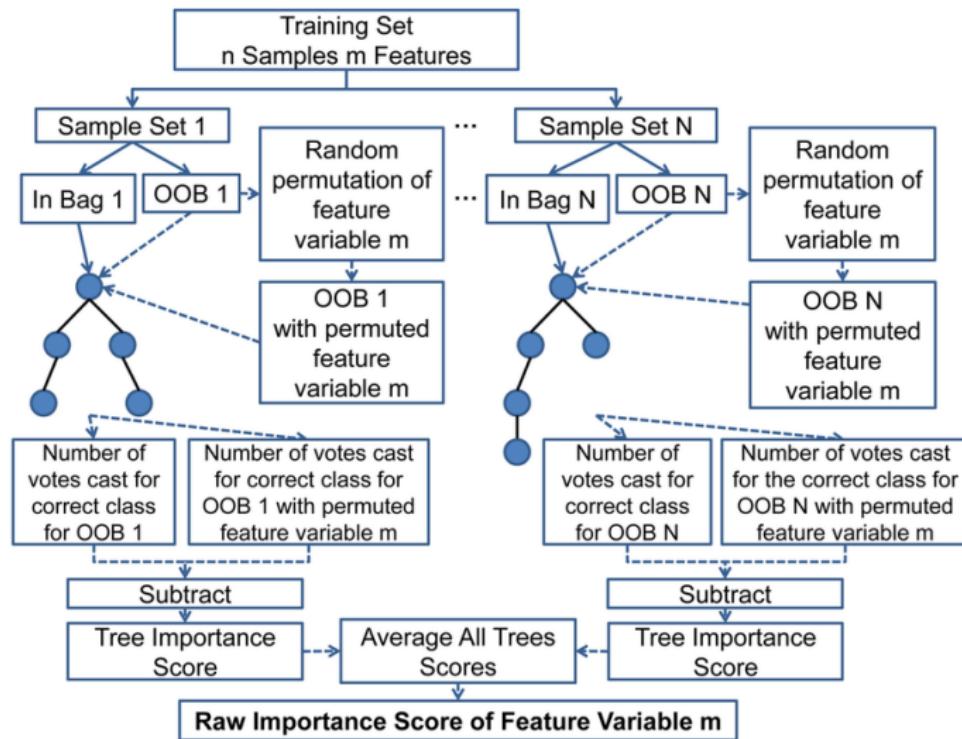
El error de prueba de las muestras de entrenamiento se computa a medida que se construye la random forest de la siguiente forma

- ▶ Para cada árbol generado, 33-36 % de las muestras no son seleccionadas en el bootstrap, y son llamadas muestras Out Of Bootstrap (OOB)
- ▶ Usando muestras OOB como entrada al árbol correspondiente, las predicciones se hacen como si fueran muestras de prueba nuevas
- ▶ A través de contabilidad, el voto mayoritario (clasificación) o el promedio (regresión) se calcula para todas las muestras OOB de todos los árboles.
- ▶ Tal error de prueba estimado es muy preciso en la práctica, para un N razonable

Estimación de la importancia de cada variable:

- ▶ Sea \hat{e} la estimación OOB del error de predicción al usar el conjunto de entrenamiento original, D , esto es, el número de errores que comete el algoritmo.
- ▶ Para cada variable x_p donde $p \in \{1, \dots, k\}$
 - Permutar aleatoriamente la variable p -ésima para generar un nuevo conjunto de muestras D' donde todos los datos son los mismos salvo el de la variable x_p que está mezclado.
 - Se computa el estimador OOB, \hat{e}_p , del error de predicción de las nuevas muestras.
- ▶ Una medida de importancia de la variable x_p es $\hat{e}_p - \hat{e}$, el aumento en el error debido a la perturbación aleatoria de la variable p -ésima.

Random Forest



Random Forest:Detalles

- ▶ Las divisiones se eligen de acuerdo con una medida de pureza,
 - ECM (regresión),
 - Gini index or deviance (clasificación)
- ▶ Como seleccionar N ?
 - Construye árboles hasta que el error no disminuya mas
- ▶ Como seleccionar M ?
 - Intente recomendar valores predeterminados, la mitad de ellos y dos veces, y elija el mejor.

Ventajas de Random Forest:

- ▶ Es uno de los algoritmos de aprendizaje con menor tasa de error disponibles. Para muchos conjuntos de datos, produce un clasificador altamente exacto.
- ▶ Se ejecuta eficientemente en grandes bases de datos.
- ▶ Puede manejar miles de variables de entrada sin eliminación de variables.
- ▶ Da estimaciones de qué variables son importantes en la clasificación.
- ▶ Genera una estimación imparcial interna del error de generalización a medida que avanza la construcción de la forest.
- ▶ Tiene un método efectivo para estimar datos faltantes y mantiene la precisión cuando falta una gran parte de los datos
- ▶ Tiene métodos para equilibrar el error en conjuntos de datos no balanceados de población de clases.

Ventajas de Random Forest:

- ▶ Los bosques generados se pueden guardar para uso futuro en otros datos.
- ▶ Se calculan prototipos que brindan información sobre la relación entre las variables y la clasificación.
- ▶ Calcula las proximidades entre pares de casos que se pueden usar para agrupar, ubicar valores atípicos o (al escalar) dar vistas interesantes de los datos.
- ▶ Las capacidades de lo anterior pueden extenderse a datos no etiquetados, lo que lleva a la agrupación no supervisada, vistas de datos y detección de valores atípicos. Ofrece un método experimental para detectar interacciones variables.
- ▶ En contraste con la publicación original. [Breiman, Random Forests, Machine Learning, 45 (1), 5–2, 2001] la implementación de scikit-learn combina clasificadores promediando sus predicción probabilística, en lugar de dejar que cada clasificador vote por un solo clase. Votación suave

Desventajas de Random Forest:

- ▶ Se ha observado sobreajuste en las tareas de clasificación/regresión con datos ruidosos.
- ▶ A diferencia de los árboles de decisión, la clasificación hecha por Random Forest es difícil de interpretar por el hombre.
- ▶ Para los datos que incluyen variables categóricas con diferente número de niveles, el Random Forest es parcial a favor de los atributos con más niveles. Por consiguiente, el score de importancia de la variable no es fiable para este tipo de datos.
- ▶ Si los datos contienen grupos de atributos correlacionados de relevancia similar para el rendimiento, entonces los grupos más pequeños están favorecidos sobre los grupos más grandes.