

Combinación de Clasificadores (Ensembles)

Alfredo Cuesta Infante

E. T. S. Ingeniería Informática
Universidad Rey Juan Carlos

Master Univ. en Visión Artificial
Reconocimiento de Patrones

Combinación mediante votación

Bagging

Random Forests

Boosting

Combinación mediante
votación

Bagging

Random Forests

Boosting

Combinación mediante votación

Bagging

Random Forests

Boosting

Combinación mediante
votación

Bagging

Random Forests

Boosting

Combinación mediante votación

Votación *hard* vs. *soft*

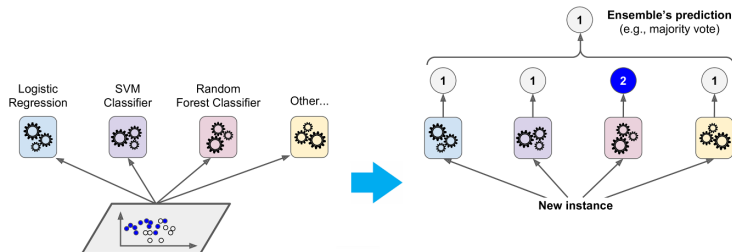


Figura: Combinación mediante votación

Cuadro: Ejemplo de votación *hard* vs. *soft*

	$p(t = 0)$	$p(t = 1)$
C_1	0.37	0.63
C_2	0.87	0.13
C_3	0.27	0.73
Votos	1	2
$\frac{1}{3} \sum_i p_i(t)$	0.503	0.496

Combinación mediante votación

Combinación de
Clasificadores
(Ensembles)

Alfredo Cuesta
Infante

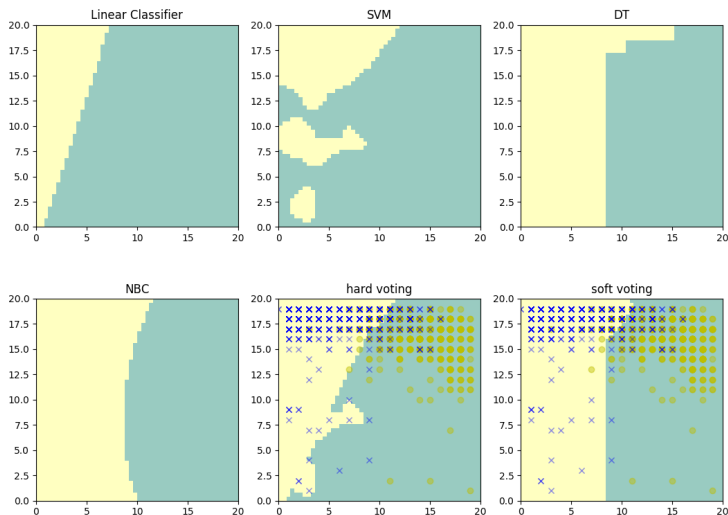
Ejemplo

Combinación mediante
votación

Bagging

Random Forests

Boosting



Combinación mediante votación

Bagging

Random Forests

Boosting

Bagging

- ▶ **bagg ing** = *bootstrap* + *aggregation*.
 - ▶ *bootstrap* es, en estadística, a remuestrear con remplazo
 - ▶ *aggregation* es el proceso de combinar datos de diferentes fuentes
- ▶ NO combina varios clasificadores,
SÍ el resultado de un clasificador sobre diferentes subconjuntos

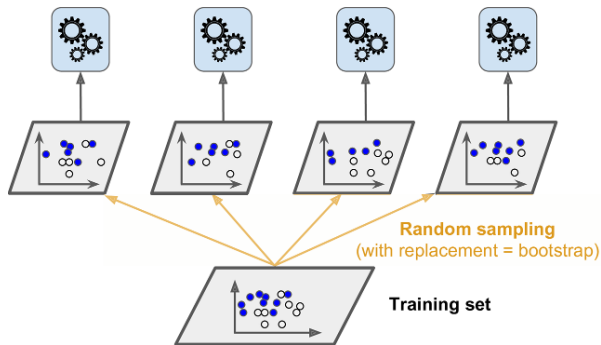


Figura: Bagging

Bagging vs. Pasting

► Bagging.

- A partir del conjunto de entrenamiento se crean K subconjuntos
- formados por N ejemplos aleatorios.
- Con remplazo = cada ejemplo puede ser seleccionado más de una vez durante el proceso de creación.
- Después se aprende un clasificador con cada uno de ellos.

► Pasting

- “Bagging SIN remplazo”...
- Cada ejemplo que pasa a formar parte de un subconjunto no puede volver a ser elegido para ese mismo subconjunto, aunque sí para otro.

Combinación de los resultados

- Podemos utilizar un sistema de votación.

Combinación mediante
votación

Bagging

Random Forests

Boosting

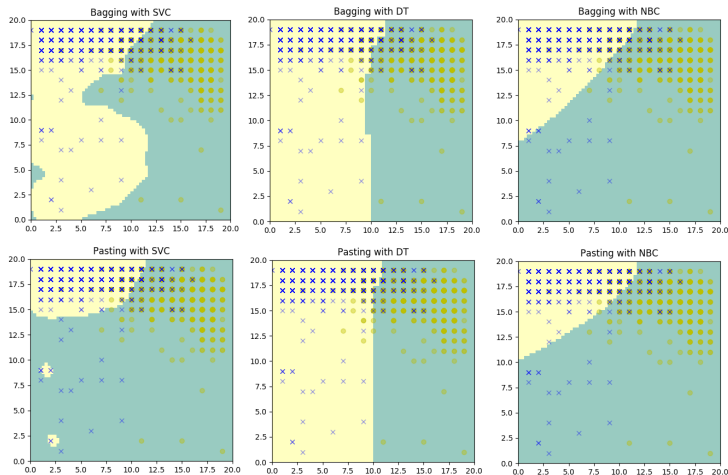


Figura: Comparativa de *bagging* vs. *pasting* con 3 clasificadores (SVM, DT y NBC). En todos los casos se utilizaron sólo 20 muestras y 100 clasificadores.

[Fuente: Original de A. Cuesta]

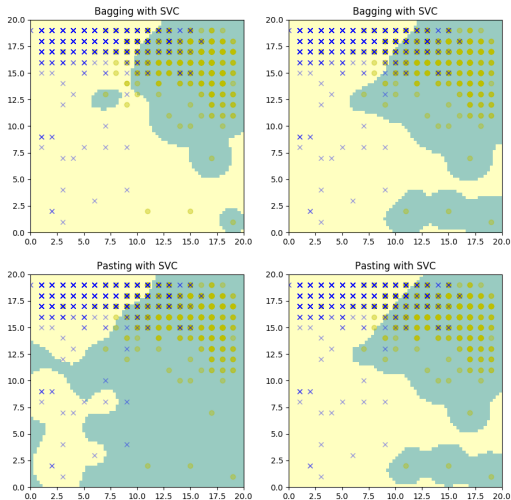


Figura: Comparativa de *bagging* (arriba) y *pasting* (abajo) con 10 clasificadores (izq.) y con 100 clasificadores (der.) [Fuente: Original de A. Cuesta]

Combinación mediante votación

Bagging

Random Forests

Boosting

- ▶ Bagging con árboles de decisión (DT).
- ▶ Utiliza DT muy sencillos, claramente subajustados, pero MUCHOS.
- ▶ Además, a la hora de crear cada DT, se seleccionan de manera **aleatoria** las características que se van a utilizar, con lo que se consigue una gran diversidad de árboles.
- ▶ Tiene un 2º resultado importante:
proporciona una medida de las características más “importantes”.
(las características con mayor poder discriminador aparecen en nodos próximos a la raíz del árbol)

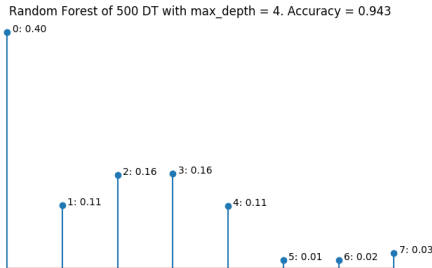


Figura: Obtención de la importancia de las características con RF.

Combinación mediante votación

Bagging

Random Forests

Boosting

Objetivo

- ▶ **To boost** Impulsar, potenciar
- ▶ Lograr un clasificador “potente” a partir de una **secuencia** de entrenamientos de uno o varios clasificadores “débiles”.

Combinación mediante
votación

Bagging

Random Forests

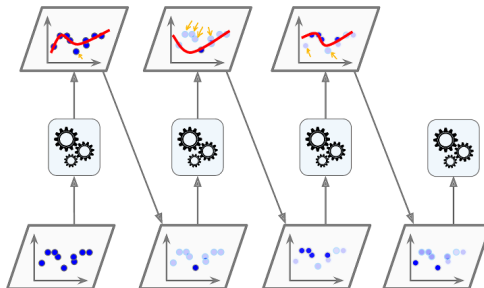
Boosting

“Similar” al turbo: los gases (residuos) de la combustión,
se utilizan para impulsar más aún al motor.



AdaBoost

► Adaptive Boosting



- ▶ Dado un conjunto de datos, ajustar un clasificador “sencillo”
 - Obtenemos errores (residuos)
- ▶ Ponderar los aciertos y los errores y volver a entrenar
 - Regla de actualización de los pesos*
- ▶ Repetir varias veces.
- ▶ Combinar todos los clasificadores obtenidos.

AdaBoost

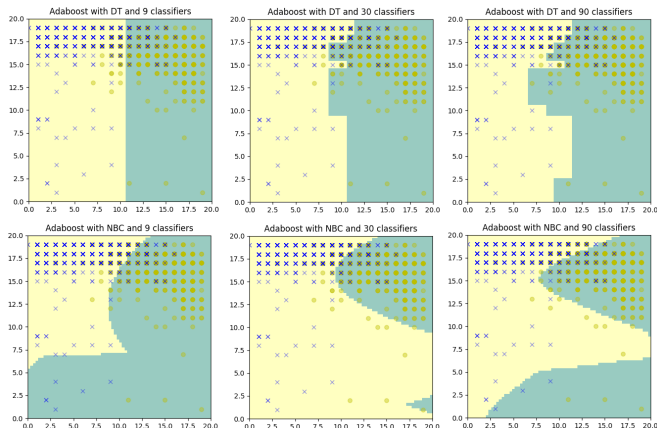


Figura: (Arriba) Adaboost con 9, 30 y 90 árboles de decisión de profundidad 1 y (abajo) con 9, 30 y 90 NBCs. [Fuente: Original de A. Cuesta]

AdaBoost

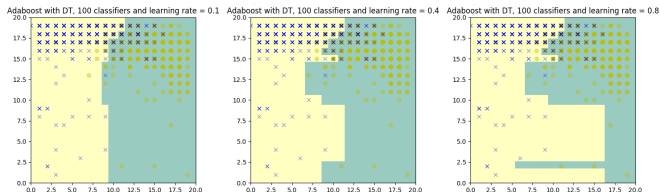


Figura: (Arriba) Adaboost con 100 árboles de decisión de profundidad 1 variando el ratio de aprendizaje: (izq.) 0.1, (centro) 0.4, (der.) 0.8. [Fuente: Original de A. Cuesta]

Gradient Boosting

- Supongamos que existe la función de regresión perfecta $h^*(\mathbf{X}) = \mathbf{y}$
- Vamos a intentar obtenerla mediante aproximaciones sucesivas

$$h^* \simeq h_0 + h_1 + h_2 + \dots$$

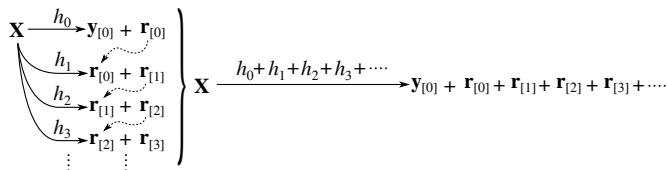


Figura: Aprendizaje de árboles de regresión con *gradient boosting*

Combinación mediante
votación

Bagging

Random Forests

Boosting