



- ❖ **Asignatura:**
Aprendizaje Inteligente
- ❖ **Catedrático:**
FRANCISCO JAVIER LUNA ROSAS
- ❖ **Tema:**
AdaBoost
- ❖ **Equipo:**
Michelle Stephanie Adame
Andrés Eloy Escobedo Esparza
- ❖ **Carrera:**
I.C.I.
- ❖ **Semestre:**
6°

AdaBoost

Boosting: se trata de una familia de "meta-algoritmos" de ensamblado de aprendices capaces de convertir un conjunto de aprendices débiles en un aprendiz fuerte.

Normalmente suponen el entrenamiento de aprendices débiles de forma recursiva: cada vez que se entrena uno, se añade al modelo final, asignándose un peso a cada uno de los aprendices entrenados en función de su eficiencia. Además, los datos analizados reciben también un peso: cada vez que se entrena un aprendiz, los pesos de los datos se modifican de forma que aquellos que han recibido una predicción menos correcta ven aumentada su importancia relativa. De esta forma el siguiente aprendiz puede concentrarse en ellos con el objeto de minimizar el error global.

Hay muchos algoritmos de boosting. El históricamente más conocido es **AdaBoost**

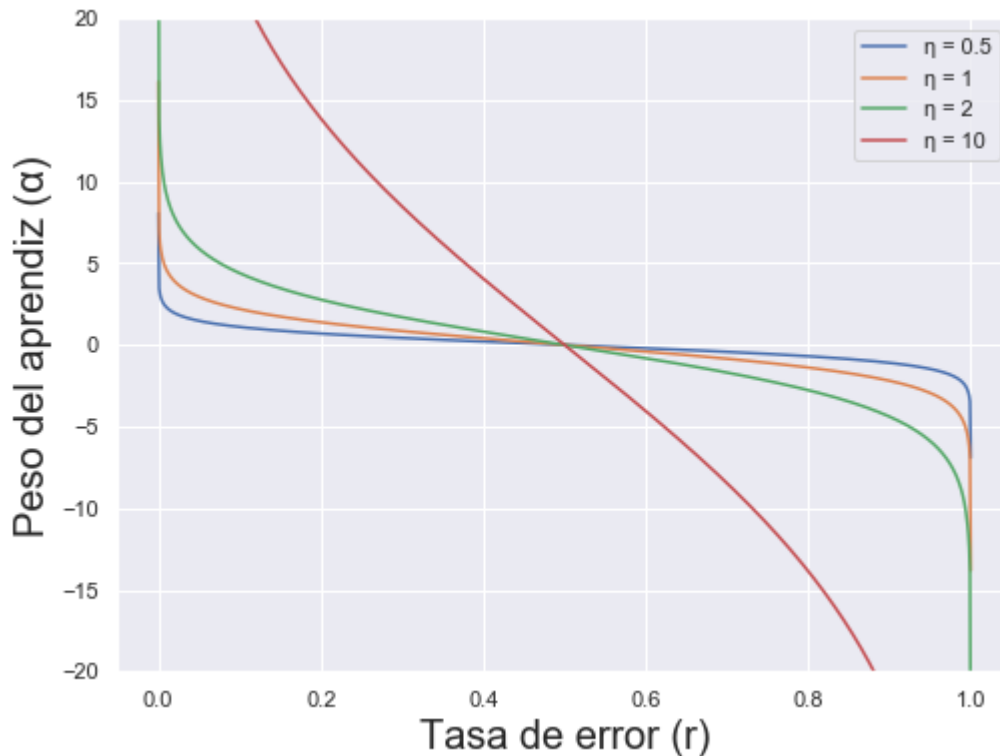
AdaBoost entrena de forma secuencial un conjunto de aprendices débiles a partir de un algoritmo base común. Todos los aprendices son entrenados con el mismo conjunto de datos, pero éstos van recibiendo pesos que dependen de los errores cometidos por cada aprendiz. Así, inicialmente todas las muestras reciben un peso inicial w_i de $1/n$ (suponiendo que haya n muestras). El primer aprendiz es entrenado y se estima su tasa de error. Suponiendo que estemos trabajando en un problema de clasificación, esta tasa de error (en general, para el aprendiz j -ésimo) sería:

$$r_j = \frac{\text{Suma de los pesos de las muestras mal clasificadas}}{\text{Suma de todos los pesos}}$$

continuación, el aprendiz recibe un peso que será tenido en cuenta al final del proceso para estimar una predicción final. Este peso viene dado por la siguiente función:

$$\alpha_j = \eta \cdot \log \frac{1 - r_j}{r_j}$$

Esta función, para diferentes valores de η , tiene la siguiente forma:

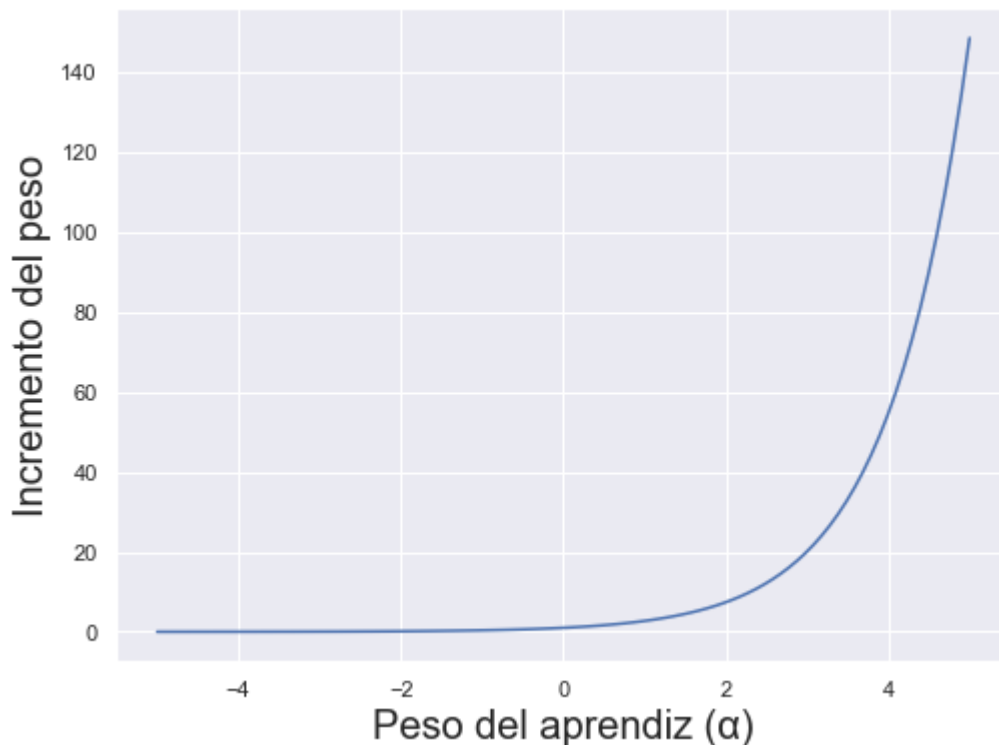


Como vemos, cuanto menor es la tasa de error, mayor será el peso que el aprendiz reciba (en el límite, un error de 0 supone un peso infinito). η es la llamada *tasa de aprendizaje*, o *learning rate*, que determina la contribución de cada aprendiz: una tasa de aprendizaje menor supone una menor variación en los pesos asignados aun cuando el rendimiento haya sido muy diferente. Por el contrario, una tasa de aprendizaje mayor supone que pequeños cambios en la tasa de error se plasme en pesos muy diferentes.

Una vez entrenado el aprendiz y calculada su tasa de error y peso relativo, se actualizan los pesos de las muestras de forma que aquellas para las que la predicción fue incorrecta vean incrementado su peso:

$$w_i = \begin{cases} w_i & \text{si la muestra fue bien clasificada} \\ w_i \cdot e^{\alpha_j} & \text{si la muestra no fue bien clasificada} \end{cases}$$

Es decir, las muestras bien clasificadas no ven modificado su peso. Por el contrario, aquellas mal clasificadas ven su peso multiplicado por la exponencial del peso del aprendiz. Esta función tiene la siguiente forma:



Obsérvese que, cuando el aprendiz ha tenido un muy mal comportamiento (cuando su tasa de error ha sido próxima a 1, lo que supone un peso del aprendiz negativo), los pesos de las muestras mal clasificadas son multiplicados por valores próximos a cero. En circunstancias normales, con una tasa de error del aprendiz inferior a 0.5, las muestras mal clasificadas ven multiplicado su peso por un valor superior a 1 (mayor cuanto mayor haya sido el peso del aprendiz, según la gráfica anterior).

Una vez actualizados los pesos de las muestras, se normalizan dividiéndolos por la suma total de los pesos.

De esta forma, el siguiente aprendiz se ve obligado a concentrarse en aquellas muestras que hayan resultado más difíciles para el aprendiz anterior (aquellas con pesos mayores).

Cuando todos los aprendices han sido entrenados, AdaBoost realiza la predicción final dando a cada aprendiz un voto ponderado según el peso relativo que haya recibido.

Conclusión:

AdaBoost es como una bendición para mejorar la precisión de nuestros algoritmos de clasificación si se usa con precisión. Es el primer algoritmo exitoso para impulsar la clasificación binaria. AdaBoost se utiliza cada vez más en la industria y ha encontrado su lugar en los sistemas de reconocimiento facial para detectar si hay un rostro en la pantalla o no.

Bibliografías:

<https://interactivechaos.com/es/manual/tutorial-de-machine-learning/adaboost>

<https://ichi.pro/es/clasificador-adaboost-en-python-155257329026137>

<https://geoffruddock.com/adaboost-from-scratch-in-python/>