



Modelos de ensamble (parte II)

Clase sincrónica

En general, ¿cómo
funcionan los algoritmos
de ensamble?



Autoaprendizaje

Recursos asincrónicos

- ¿Revisaste los recursos de la semana 2 (Guía y desafío)?
- ¿Tienes dudas sobre alguno de ellos?



Ideas fuerza



Ensamble
secuencial
**Adaptive
Boosting** utiliza
**ponderación de
observaciones
mal clasificadas**
para **ir
entrenando un
aprendiz fuerte**



Ensamble
secuencial **Gradient
Boosting** penaliza
los residuos en la
función de **pérdida**
potenciando un
aprendiz fuerte



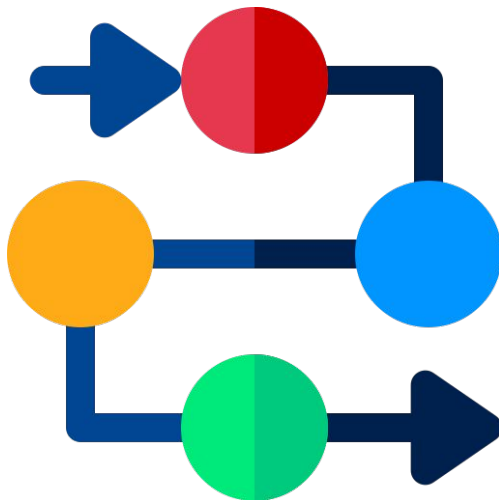
XGradient Boosting
implementación de
Gradient Boosting
de alto rendimiento
computacional, y
buen desempeño.
Determinar un buen
Learning rate es
importante para
lograr un buen
rendimiento.

/* Boosting */

Boosting

Definición

Enfoque de ensamble secuencial, donde se usan aprendices débiles para conformar el ensamble, los cuales son potenciados secuencialmente para dar origen a un aprendiz fuerte.



Boosting

Ensambls paralelos y ensambles secuenciales

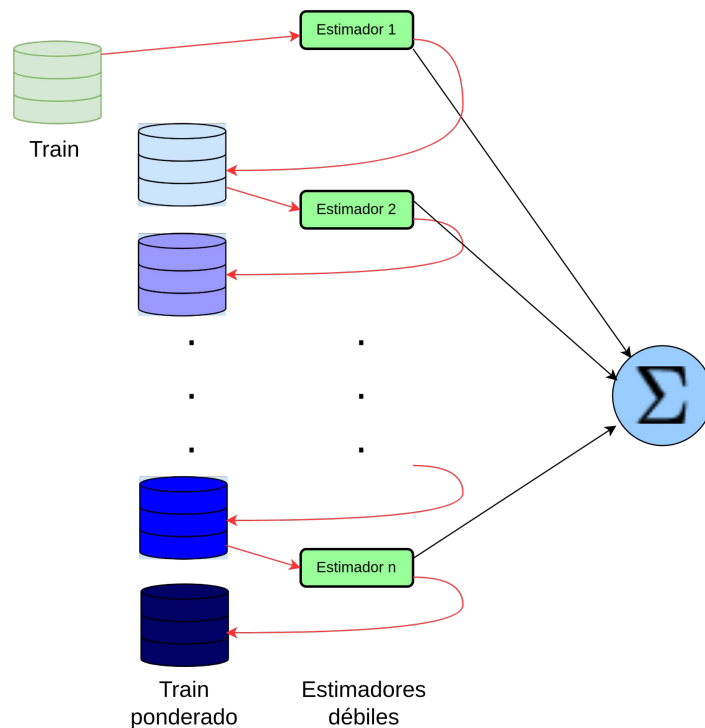
- En los ensambles paralelos se utilizan *aprendices fuertes*. En cambio, los ensambles secuenciales utilizan *aprendices débiles*.
- Para ambos tipos de ensamble el objetivo es ir dirigiendo el siguiente entrenamiento de un aprendiz de acuerdo a los errores del aprendiz actual.
- Los aprendices en los ensambles paralelos pueden ser entrenados en forma independiente, pudiendo aprovechar las actuales arquitecturas multicore. En los ensambles secuenciales esto no ocurre, ya que precisamente se va utilizando un aprendiz después de otro.

/* Adaptive Boosting*/

Adaptive Boosting

Algoritmo de ensamble secuencial

- Inicialización
- Iterar para cada aprendizaje débil
 - Entrenamiento del aprendiz débil
 - Calcular el error ponderado
 - Calcular coeficiente alpha
 - Actualizar los pesos de la instancia y su normalización
- Salida construcción del modelo final.



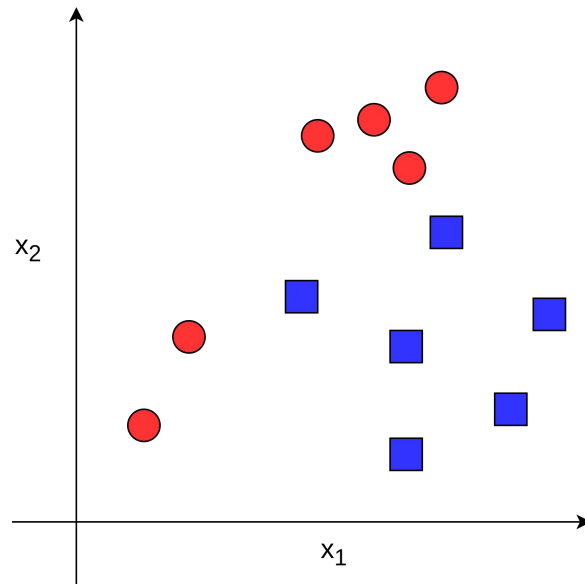
Adaptive Boosting

Inicialización

Se toma el conjunto de entrenamiento y aplicando una ponderación uniforme a cada observación se procede a iniciar el proceso.

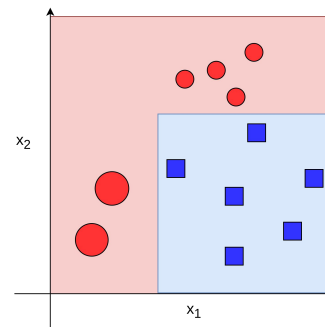
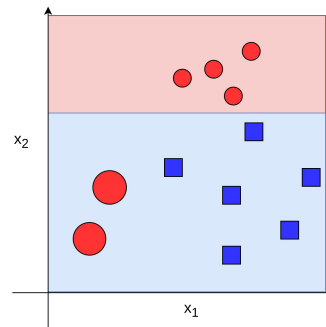
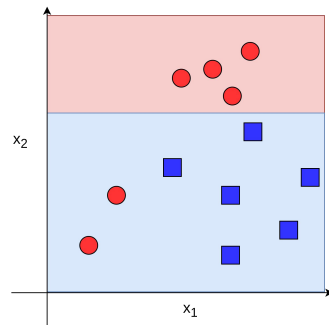
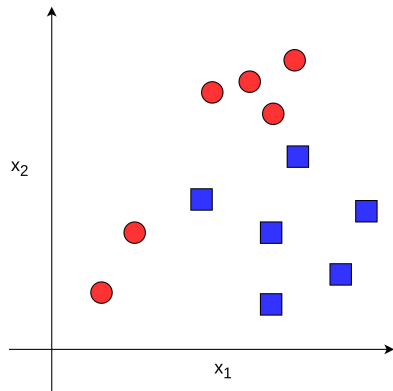
$$w_i^m = \frac{1}{n}$$

$$\sum_{i=1}^n w_i^m = 1$$



Adaptive Boosting

Construcción del modelo final para predicciones



/* Gradient Boosting */

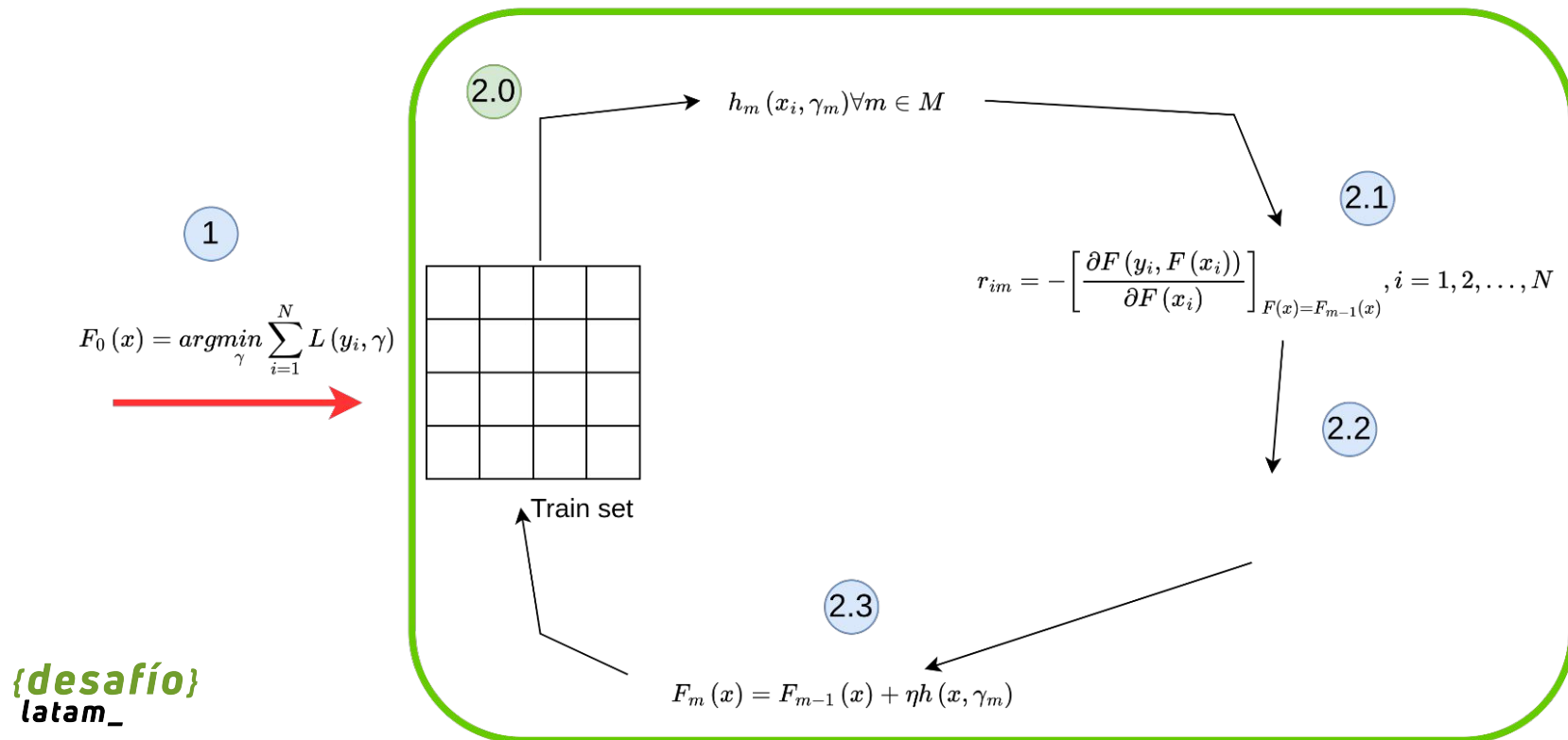
Gradient Boosting

Descripción

- Gradient Boosting entrena cada aprendiz débil de acuerdo a los errores residuales del aprendiz anterior.
- Adaptive Boosting pondera las observaciones mal clasificadas, en cambio Gradient Boosting penaliza en base al error residual
- La penalización realizada en Gradient Boosting se hace por medio de una **función de pérdida** o **función de costo**. función de pérdida o costo, que mide la discrepancia entre los verdaderos valores y los valores predichos. Usando esta función el problema de encontrar el “mejor” modelo, se traduce en un problema de optimización.

Gradient Boosting

Algoritmo paso a paso



Gradient Boosting

Hiper parámetro `learning_rate`

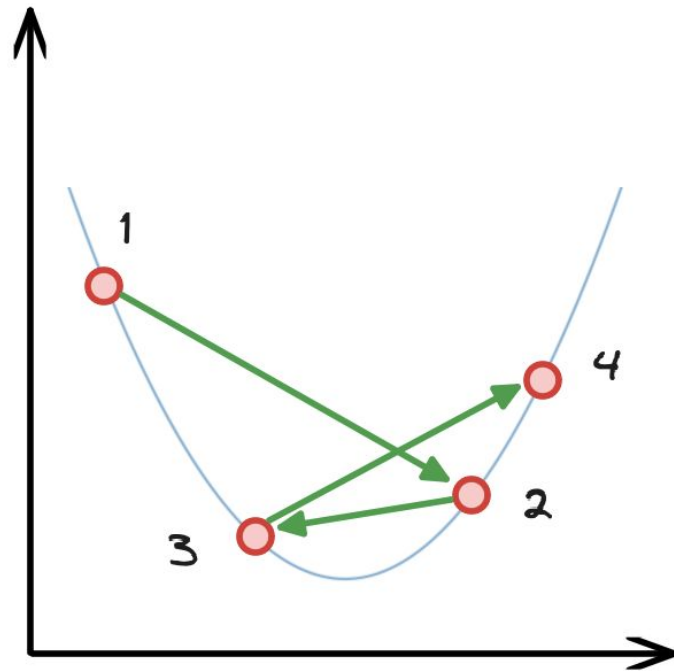
- ***learning_rate*** define el “paso” que se dará en la función de pérdida en cada iteración, con el objetivo de alcanzar un mínimo.
- Recordemos que en el descenso del gradiente nos ubicamos en un punto inicial y calculamos la dirección en que la función desciende más rápidamente. Esto se hace calculando el gradiente; junto con ello debemos definir cuánto nos moveremos en la dirección encontrada.

Gradient Boosting

Hiper parámetro `learning_rate`

`learning_rate` muy alto

- Provoca que nos movamos más rápido en la función de pérdida
- Puede provocar falta de convergencia, en cuanto a que podríamos estar cerca del óptimo pero ya que el paso es alto nos haría volver a alejarnos una y otra vez.

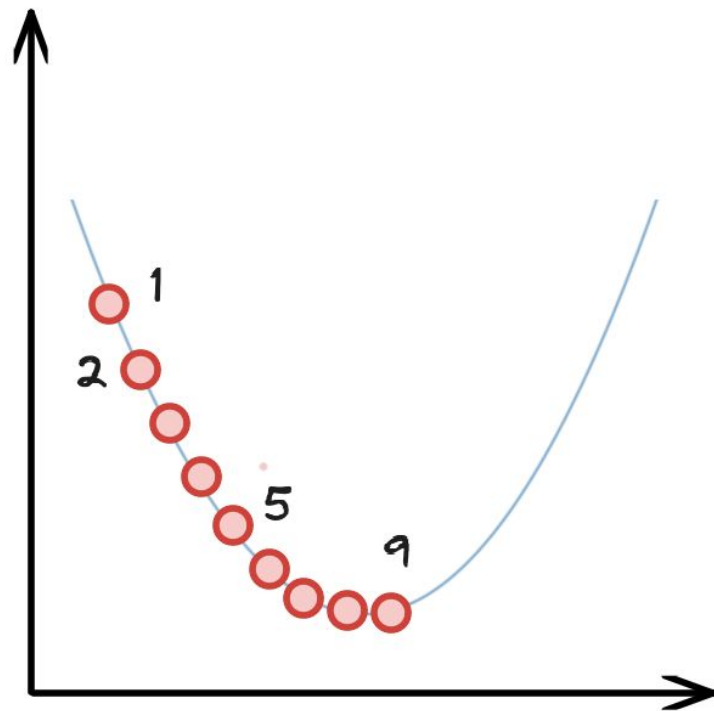


Gradient Boosting

Hiper parámetro *learning_rate*

learning_rate muy bajo

- Provoca que nos movamos lentamente en la función de pérdida, provocando un tiempo excesivo en el entrenamiento.
- De esta forma podremos alcanzar un mínimo. sin embargo, podemos caer fácilmente en mínimos locales.



/* XGradient Boosting*/

XGradient Boosting

Descripción

- Implementación de Gradient Boosting que acelera el entrenamiento, presenta un alto rendimiento.
- Ha ganado popularidad ya que ha ganado varios concursos de Kaggle

Se recomienda para:

- Conjuntos de alta dimensión, para grandes volúmenes de datos, en que se requiera alta precisión.

¡Manos a la obra!

Aplicando modelos boosting



¡Manos a la obra!

Aplicando modelos boosting

Veremos a continuación cómo implementar modelos Boosting con Python. Para ello, utiliza el archivo **01 - Boosting** y sigue los pasos que te indicará tu profesor.



Desafío “Detección de cardiopatía”



Desafío

“Detección de cardiopatía”

- ¿Leíste el desafío de esta semana? ¿Comprendes bien lo que se solicita en cada caso?
- ¿Hay contenidos que necesitas repasar antes de comenzar este desafío?
- ¿Necesitas algún ejemplo o indicación para alguna pregunta o requerimiento específico?



Cada aprendiz débil es un
paso a la perfección,
corrigiendo errores y
refinando la predicción.



{desafío}
latam_

*Academia de
talentos digitales*

