

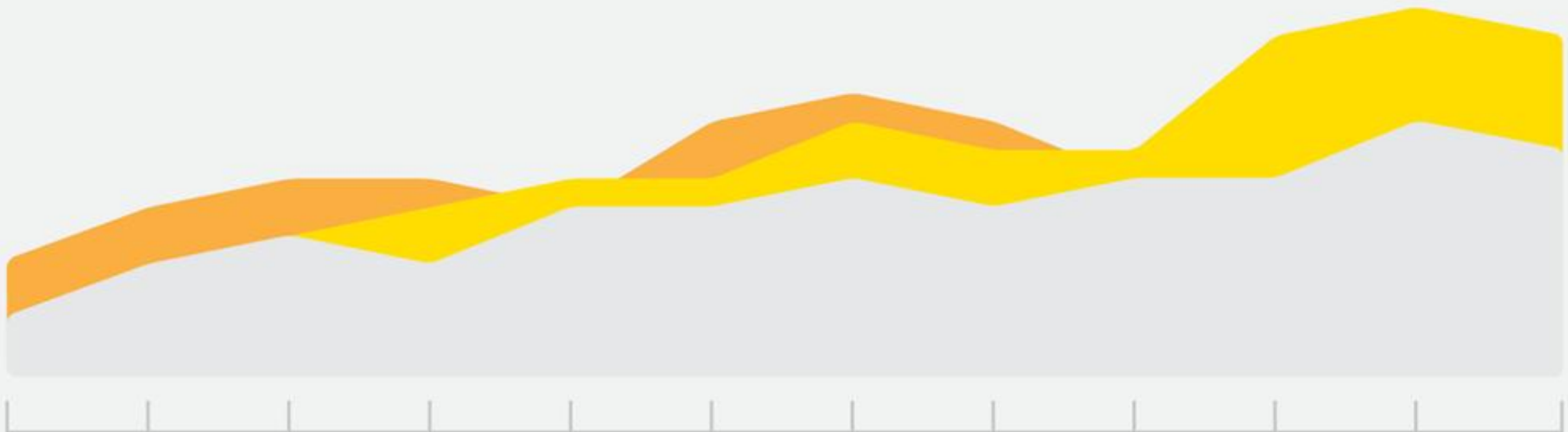


Rev. Thomas Bayes



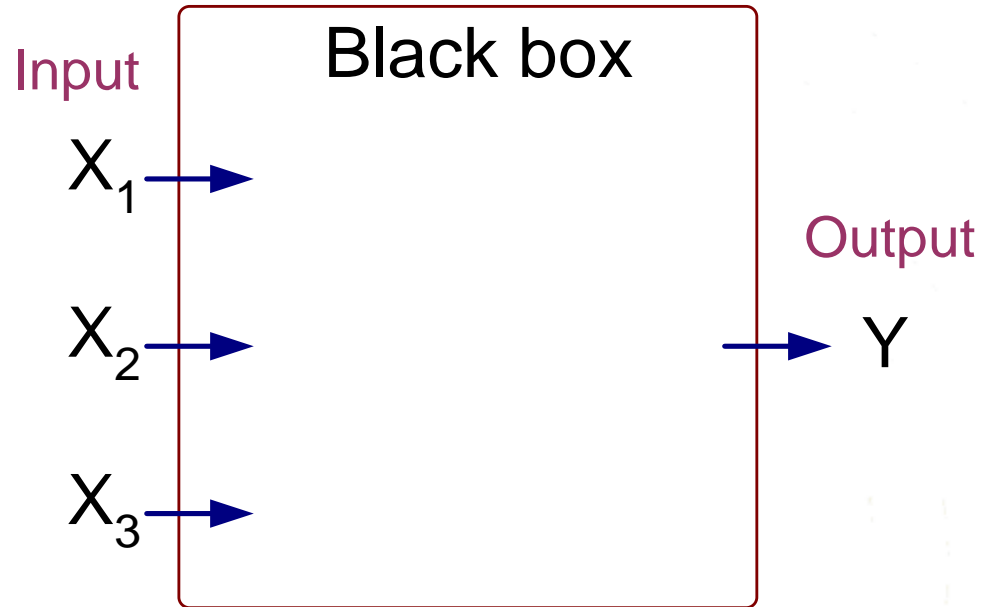
Aprendizaje Supervisado

Redes Neuronales, Métodos de Consenso y Potenciación



Redes Neuronales

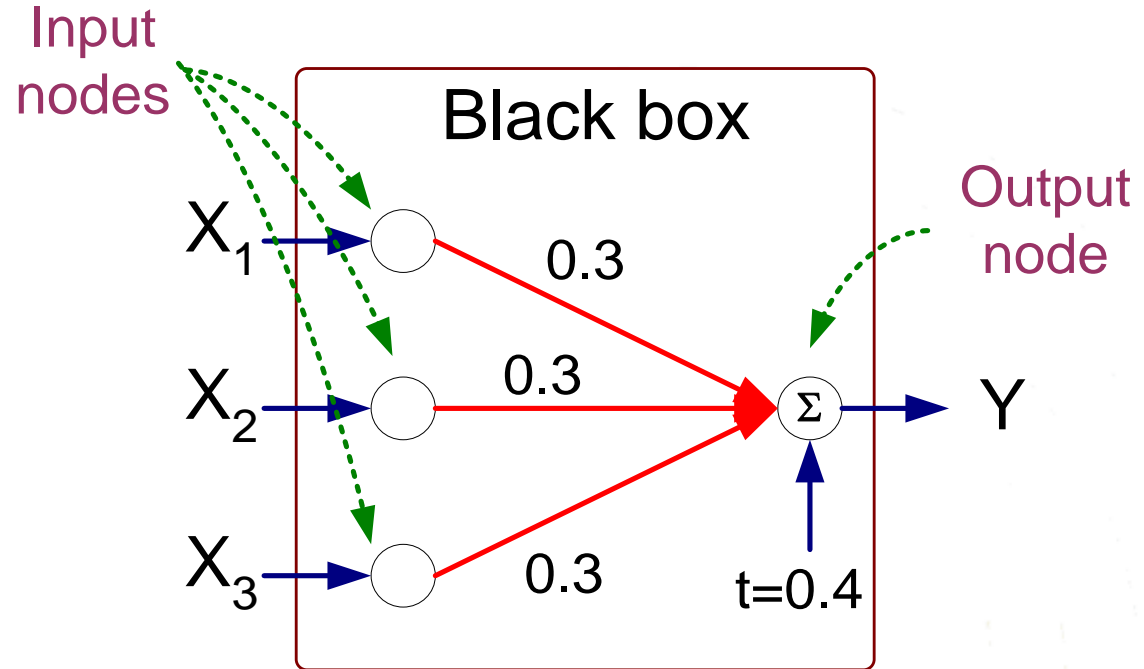
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



La salida Y es 1 si al menos 2 de las 3 entradas son iguales a 1.

Redes Neuronales

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

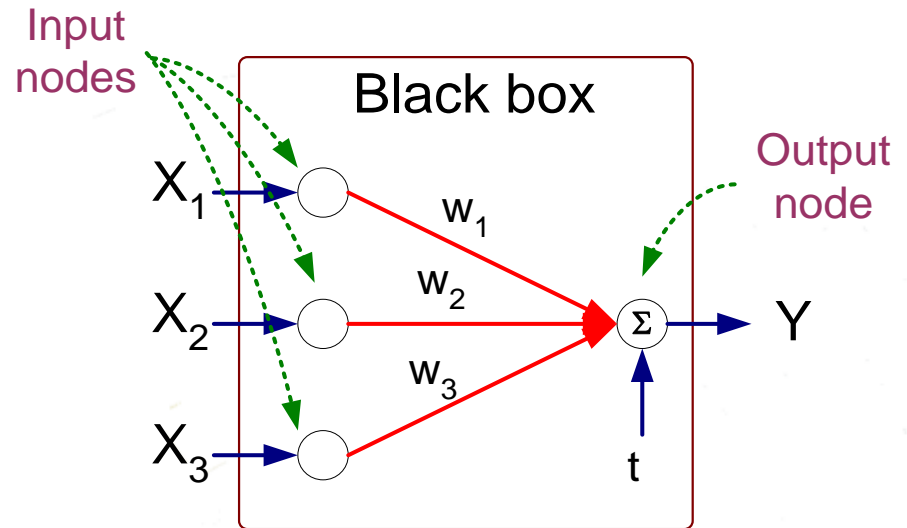


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 > 0.4)$$

$$\text{donde } I(z) = \begin{cases} 1 & \text{Si } z \text{ es verdadero} \\ 0 & \text{En otro caso} \end{cases}$$

Redes Neuronales

- Modelo es un conjunto de nodos interconectados y enlaces ponderados
- Nodo de salida suma cada uno de su valor de entrada de acuerdo a los pesos de sus vínculos
- Comparar nodo de salida contra un **umbral t**



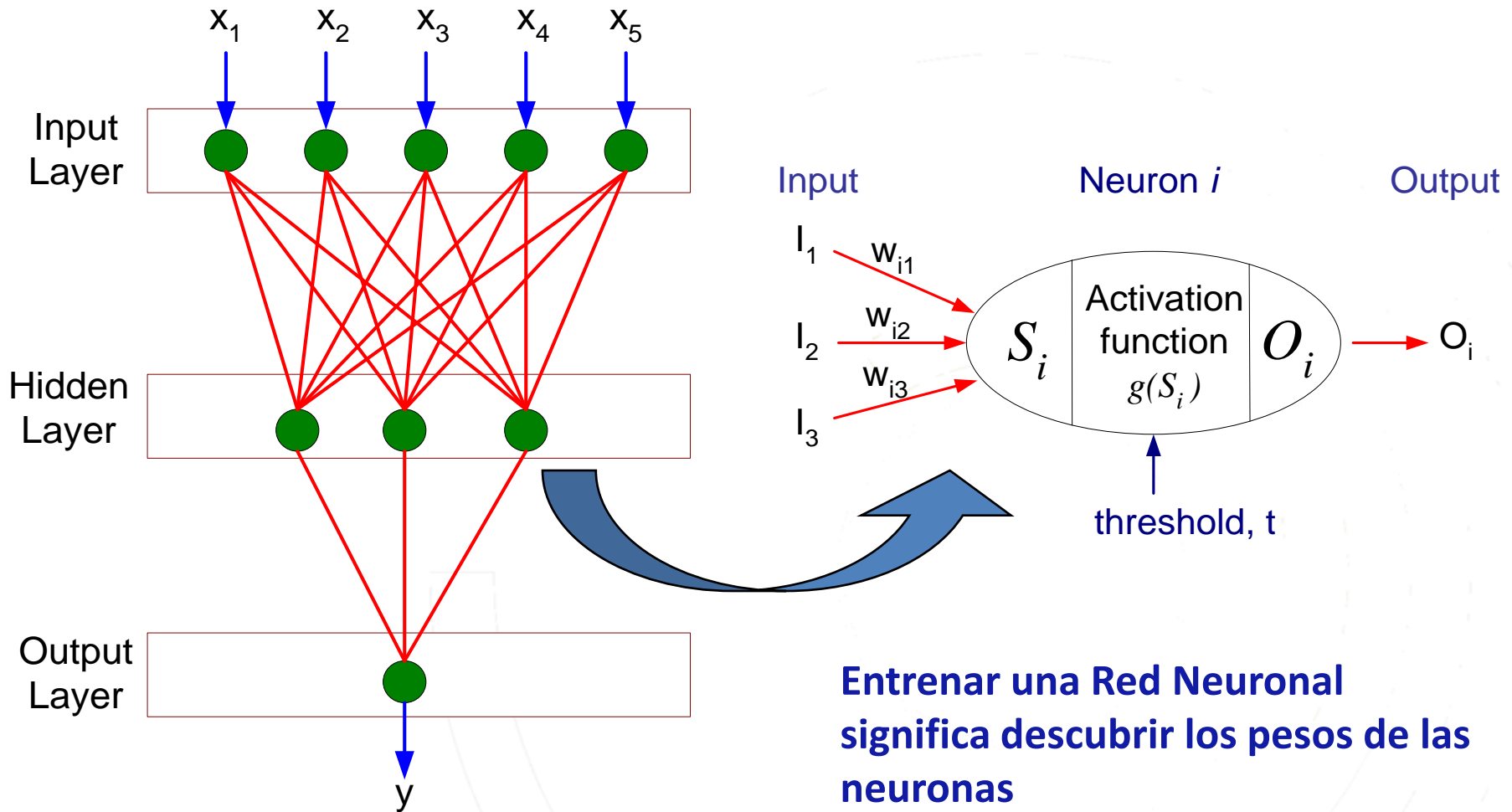
Modelo Perceptron

$$Y = I\left(\sum_i w_i X_i - t\right) \quad \circ$$

$$Y = \text{sign}\left(\sum_i w_i X_i - t\right)$$



Estructura General de una Red Neuronal



Algoritmo de Aprendizaje

- Inicializar los pesos (w_0, w_1, \dots, w_k)
- Ajustar los pesos de tal manera que la salida de la red Neuronal sea consistente con etiquetas de clase de ejemplos de entrenamiento:
$$E = \sum_i [Y_i - f(w_i, X_i)]^2$$
- Función Objetivo:
 - Encuentra el peso w_i 's de que minimizan la función objetivo anterior (Error Cuadrático)
 - Ej. Algoritmo “backpropagation”

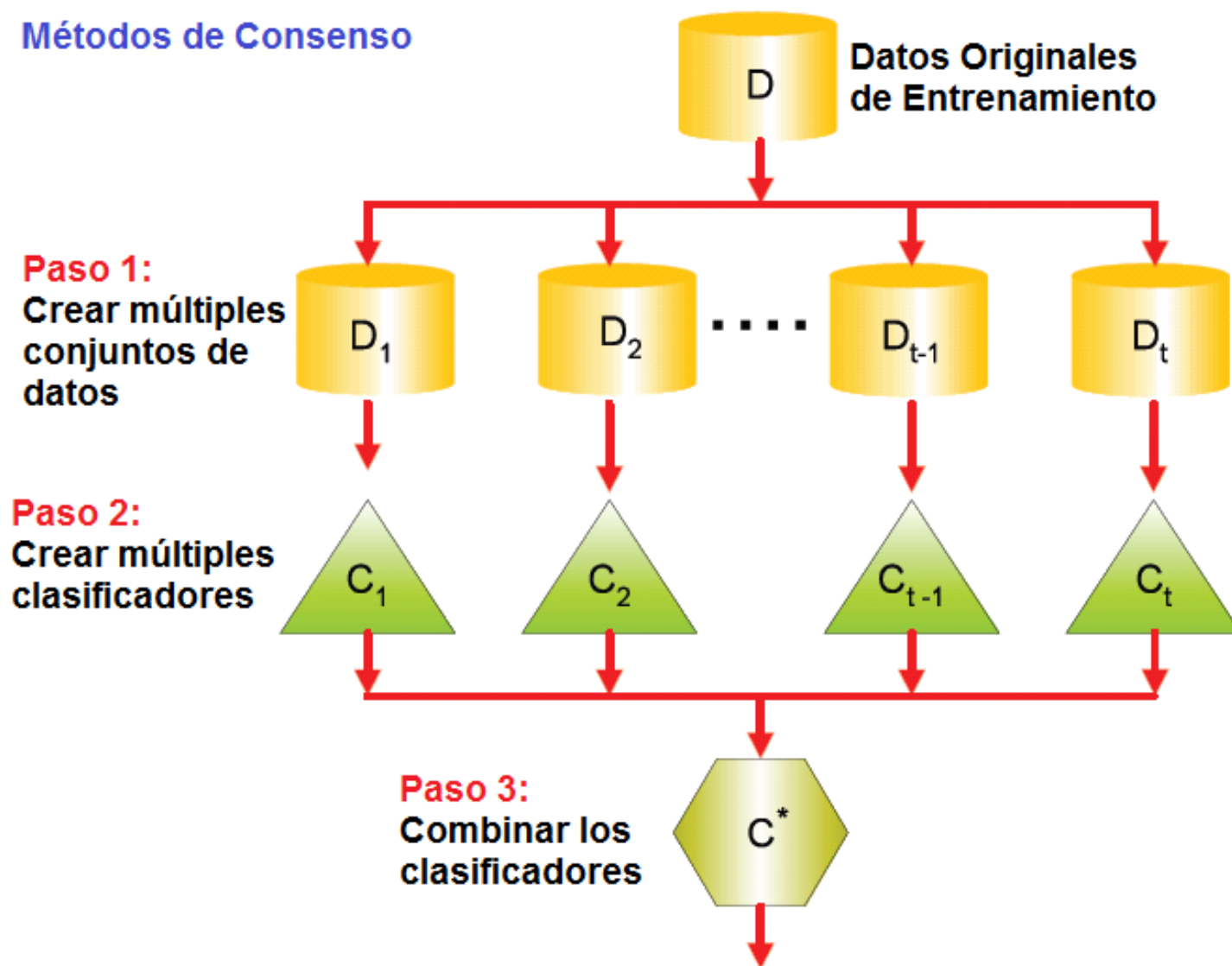


Métodos de Consenso (Bagging)

- La idea es tomar m muestras aleatorias con reemplazo (Bootstrap) de los datos originales y luego aplicar en cada una de ellas un método predictivo para luego con algún criterio establecer un consenso de todos los resultados
- El consenso podría ser un promedio, un promedio ponderado basado en cuál método obtuvo los mejores resultados
- El que obtenga la “mayor cantidad de votos”



Métodos de Consenso

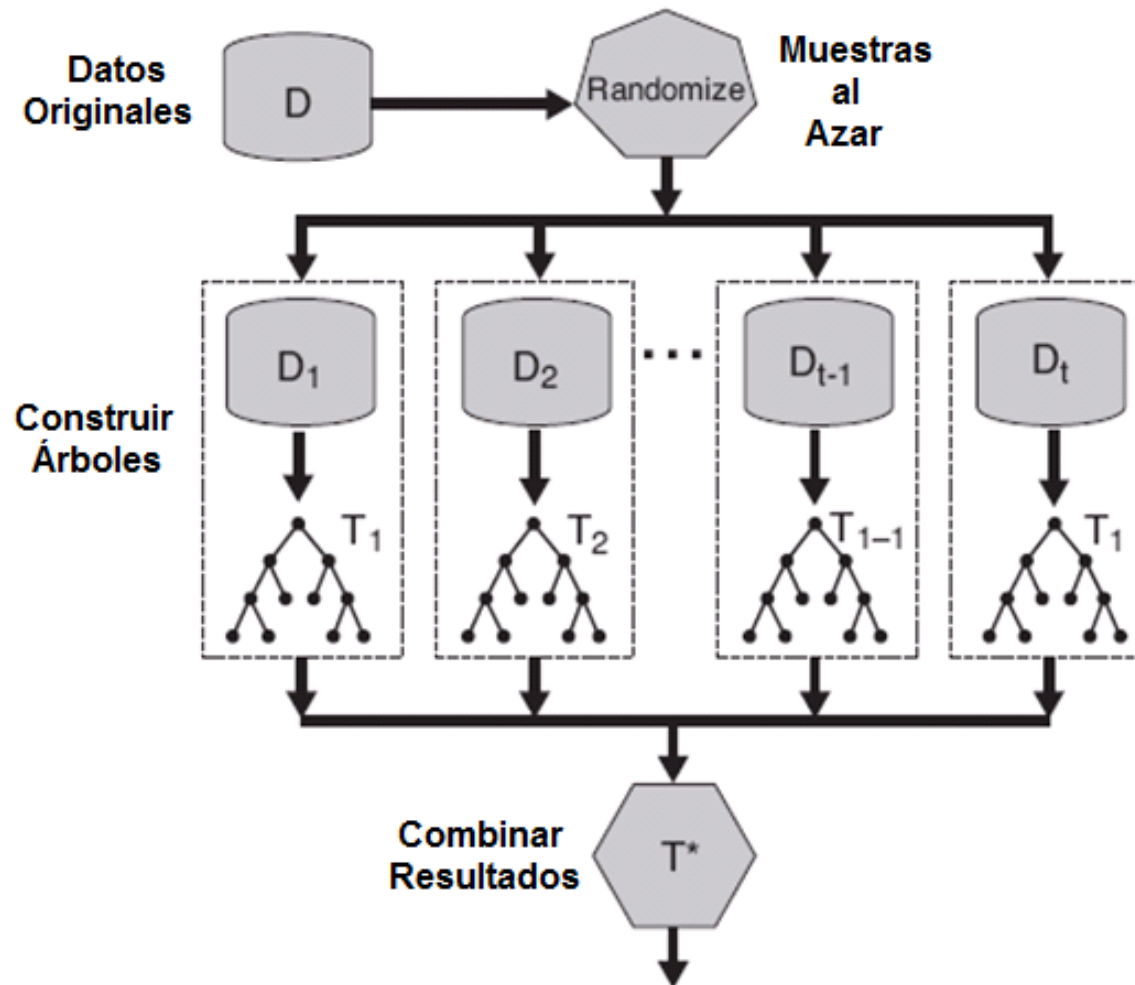


Bosques Aleatorios (Random Forest)

- El caso en el que todos los clasificadores del Método de Consenso son Árboles dicho método se denomina Bosques Aleatorios (Random Forest)



Bosques Aleatorios (Random Forest)

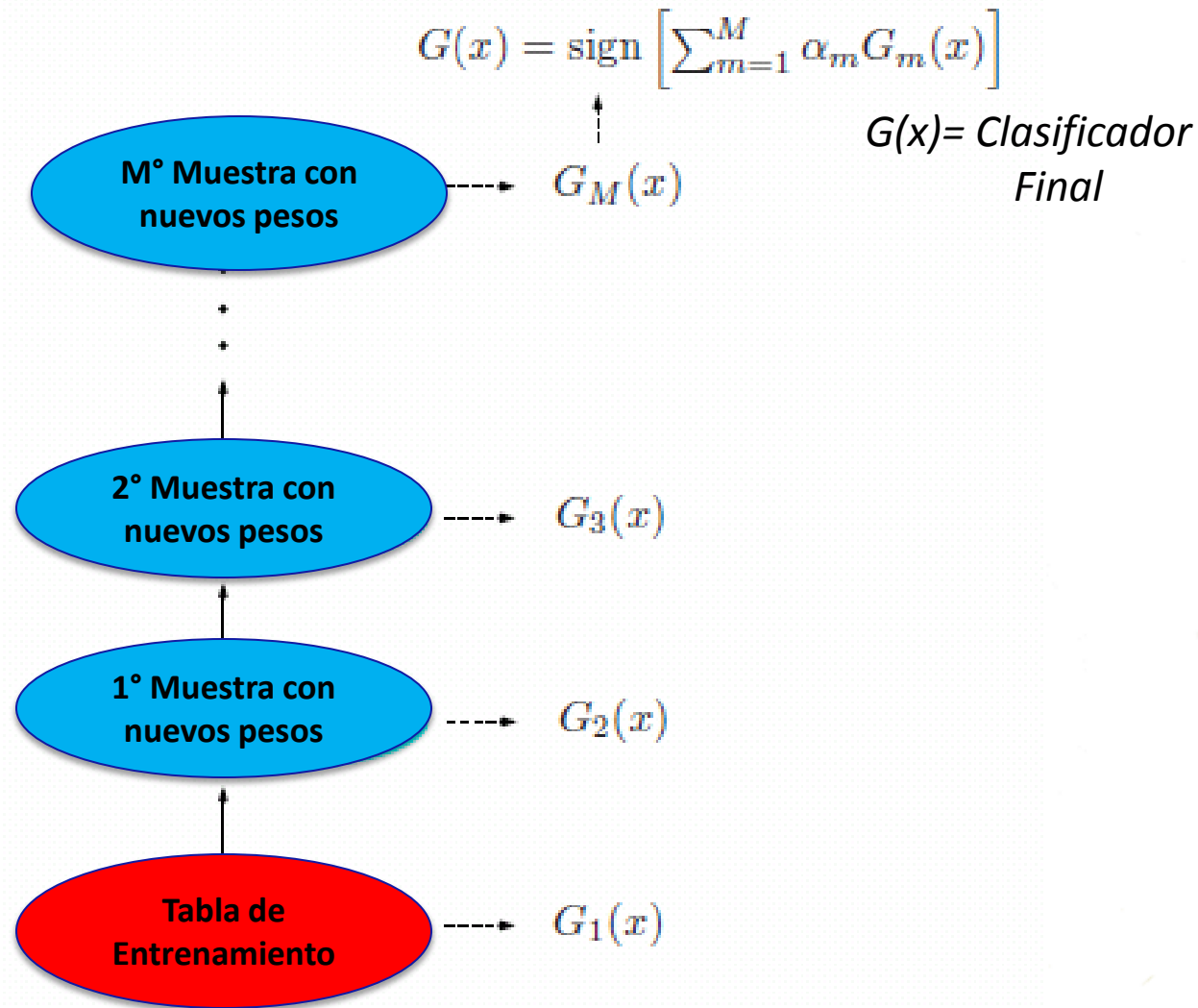


Métodos de Potenciación

- La idea es tomar una muestra aleatoria de los datos originales y aplicar sobre esta un método clasificadorio luego *aumentar el peso (potenciar)* a los individuos mal clasificados para que en la siguiente aplicación del método clasificadorio mejore su clasificación, y así sucesivamente...
- Además le asigna un peso al modelo construido en cada paso.
- **Observación:** Solo funciona para problemas de clasificación **binarios** (de 2 clases).



Métodos de Potenciación



Métodos de Potenciación

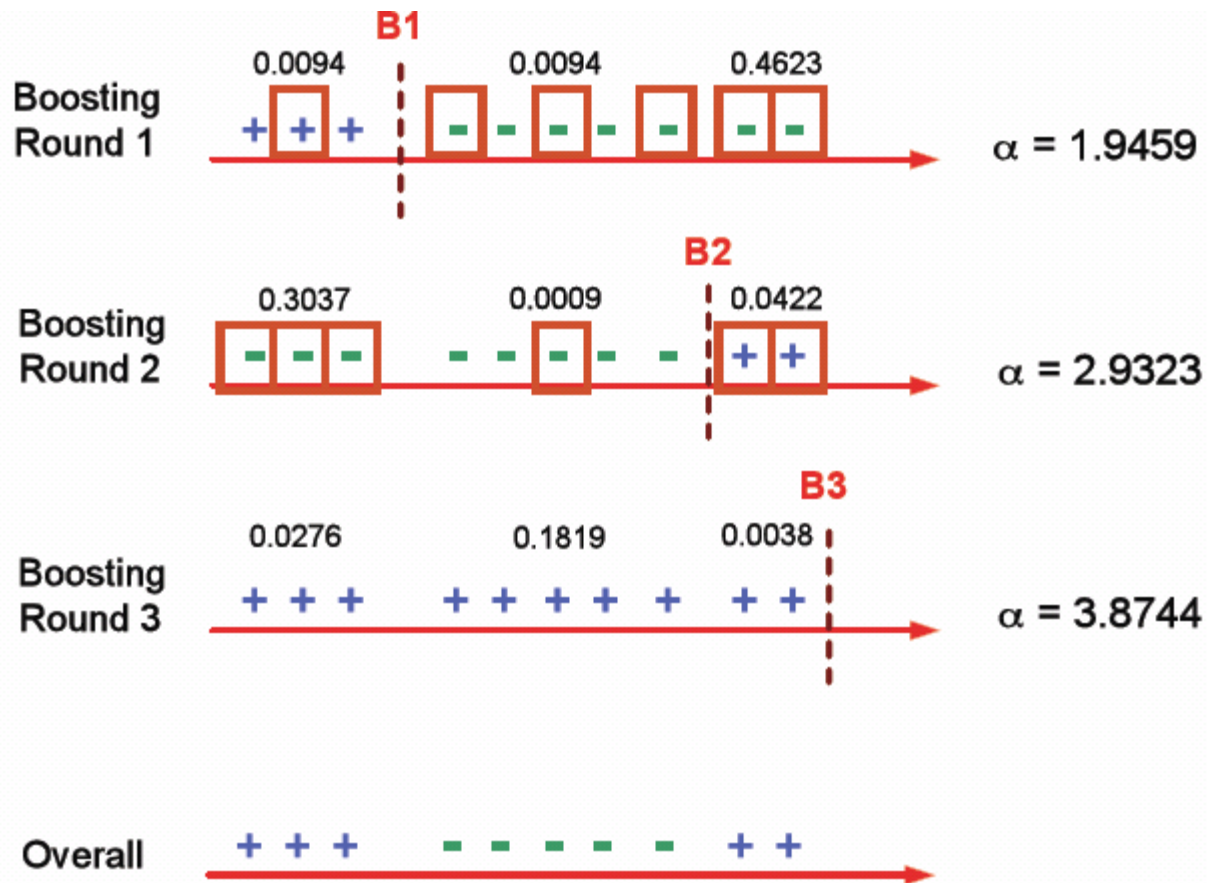
Algoritmo:AdaBoost.M1

1. Initialize the observation weights $w_i = 1/N$, $i = 1, 2, \dots, N$.
2. For $m = 1$ to M :
 - (a) Fit a classifier $G_m(x)$ to the training data using weights w_i .
 - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
 - (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
 - (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$, $i = 1, 2, \dots, N$.
3. Output $G(x) = \text{sign} \left[\sum_{m=1}^M \alpha_m G_m(x) \right]$.



Métodos de Potenciación

Algoritmo: AdaBoost.M1



Gracias....



oldemar **rodríguez**

CONSULTOR en M1N&R14 D& D4T0S