IIC 2433 Minería de Datos

https://github.com/marcelomendoza/IIC2433

Ensembles

Ensembles

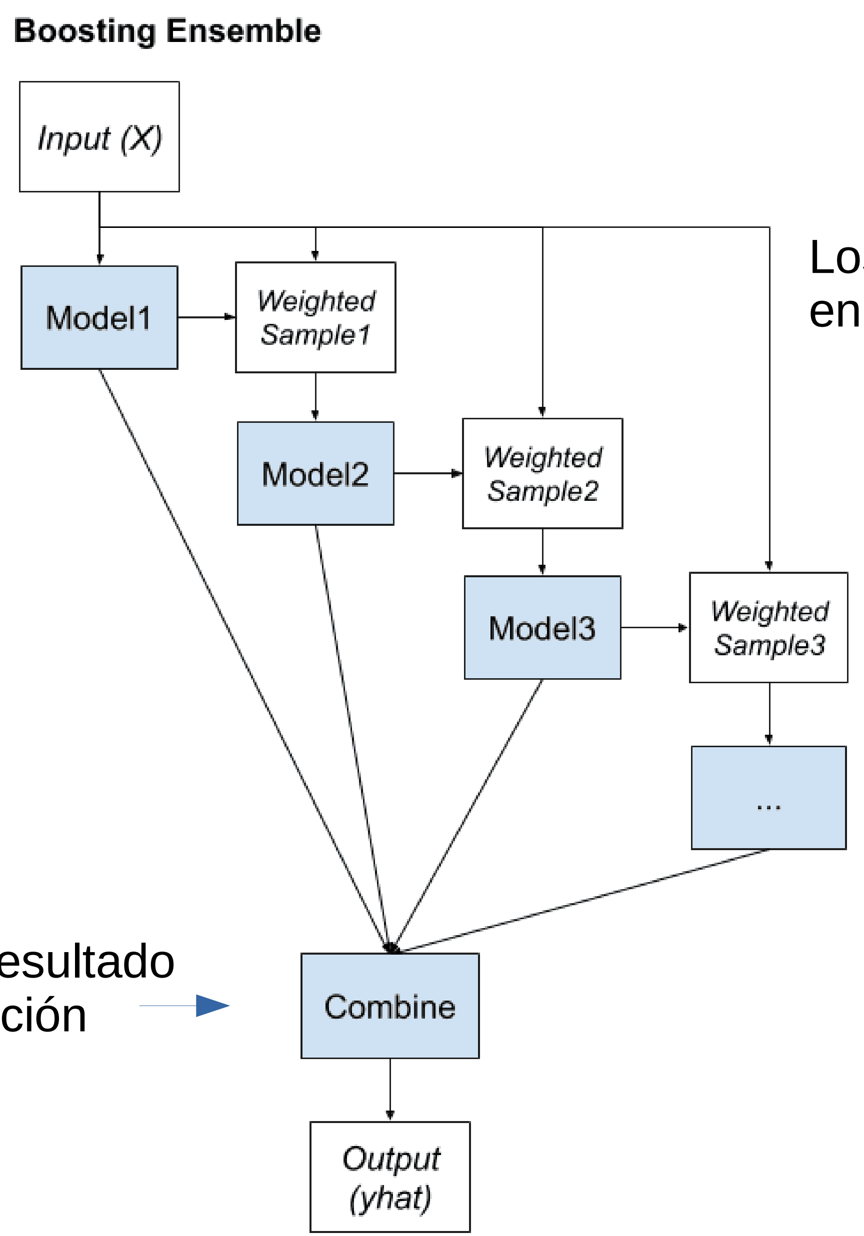
En ensembles asumimos que al usar varios modelos podemos obtener mejores resultados que usando un modelo.

Los ensembles denominan a los modelos base *weak learners*.

Existen tres estrategias para combinar modelos:

- Boosting (AdaBoost, XGBoost, ...)
- Bagging (random forests, extra trees, ...)
- Stacking (canonical stacking, super ensemble, ...)

Boosting



Los nuevos modelos se enfocan en los datos más difíciles.

La salida es el resultado de una combinación

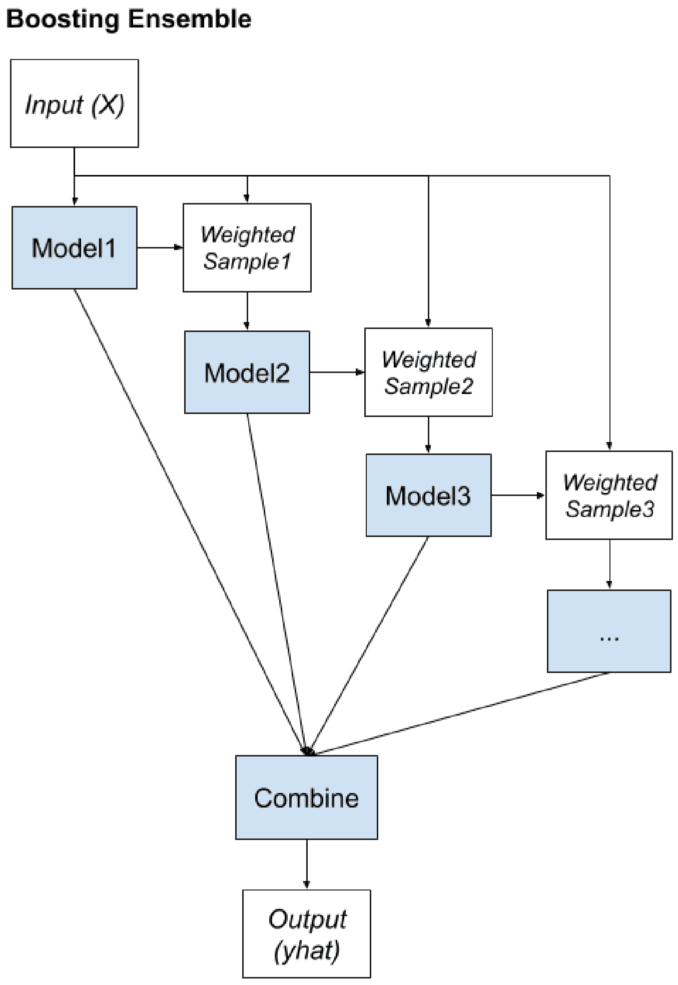
Boosting

Input: Sample distribution $\mathcal{D}$;
Base learning algorithm $\mathfrak{L}$;
Number of learning rounds T.

Process:

1. $\mathcal{D}_1 = \mathcal{D}$. % Initialize distribution
2. **for** $t = 1, \ldots, T$:
3. $h_t = \mathfrak{L}(\mathcal{D}_t)$; % Train a weak learner from $\mathcal{D}_t$
4. $\epsilon_t = P_{\boldsymbol{x} \sim D_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$; % Evaluate the error
5. $\mathcal{D}_{t+1} = Adjust\_Distribution(\mathcal{D}_t, \epsilon_t)$
6. **end**

Output: $H(\boldsymbol{x}) = Combine\_Outputs(\{h_1(\boldsymbol{x}), \ldots, h_t(\boldsymbol{x})\})$



AdaBoost

AdaBoost es un algoritmo basado en Boosting muy usado.

AdaBoost tiene por objetivo minimizar la pérdida según:

$$\ell_{\exp}(h \mid \mathcal{D}) = \mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[e^{-f(\boldsymbol{x})h(\boldsymbol{x})}]$$

Exponential loss

Clases en $\{-1, +1\}$

AdaBoost

AdaBoost es un algoritmo basado en Boosting muy usado.

AdaBoost tiene por objetivo minimizar la pérdida según:

$$\ell_{\exp}(h \mid \mathcal{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})h(\boldsymbol{x})}]$$

Exponential loss

Clases en $\{-1, +1\}$

Los *weak learners* se combinan de forma lineal:

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \ .$$

Hay que ajustarlos

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \,.$$

AdaBoost

Minimizando: $\ell_{\exp}(h \mid \mathcal{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})h(\boldsymbol{x})}]$

$$\begin{aligned}
\frac{\partial e^{-f(\boldsymbol{x})H(\boldsymbol{x})}}{\partial H(\boldsymbol{x})} &= -f(\boldsymbol{x})e^{-f(\boldsymbol{x})H(\boldsymbol{x})} \\
&= -e^{-H(\boldsymbol{x})}P(f(\boldsymbol{x}) = 1 \mid \boldsymbol{x}) + e^{H(\boldsymbol{x})}P(f(\boldsymbol{x}) = -1 \mid \boldsymbol{x}) \\
&= 0 \,.
\end{aligned}$$

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \; .$$

AdaBoost

Minimizando: $\ell_{\exp}(h \mid \mathcal{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})h(\boldsymbol{x})}]$

$$\begin{aligned}
\frac{\partial e^{-f(\boldsymbol{x})H(\boldsymbol{x})}}{\partial H(\boldsymbol{x})} &= -f(\boldsymbol{x})e^{-f(\boldsymbol{x})H(\boldsymbol{x})} \\
&= -e^{-H(\boldsymbol{x})}P(f(\boldsymbol{x}) = 1 \mid \boldsymbol{x}) + e^{H(\boldsymbol{x})}P(f(\boldsymbol{x}) = -1 \mid \boldsymbol{x}) \\
&= 0 \; .
\end{aligned}$$

$$H(\boldsymbol{x}) = \frac{1}{2} \ln \frac{P(f(x) = 1 \mid \boldsymbol{x})}{P(f(x) = -1 \mid \boldsymbol{x})}$$

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) .$$

AdaBoost

Minimizando: $\ell_{\exp}(h \mid \mathcal{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})h(\boldsymbol{x})}]$

$$\begin{aligned}\frac{\partial e^{-f(\boldsymbol{x})H(\boldsymbol{x})}}{\partial H(\boldsymbol{x})} &= -f(\boldsymbol{x})e^{-f(\boldsymbol{x})H(\boldsymbol{x})} \\ &= -e^{-H(\boldsymbol{x})}P(f(\boldsymbol{x}) = 1 \mid \boldsymbol{x}) + e^{H(\boldsymbol{x})}P(f(\boldsymbol{x}) = -1 \mid \boldsymbol{x}) \\ &= 0 .\end{aligned}$$

$$H(\boldsymbol{x}) = \frac{1}{2}\ln\frac{P(f(x) = 1 \mid \boldsymbol{x})}{P(f(x) = -1 \mid \boldsymbol{x})}$$

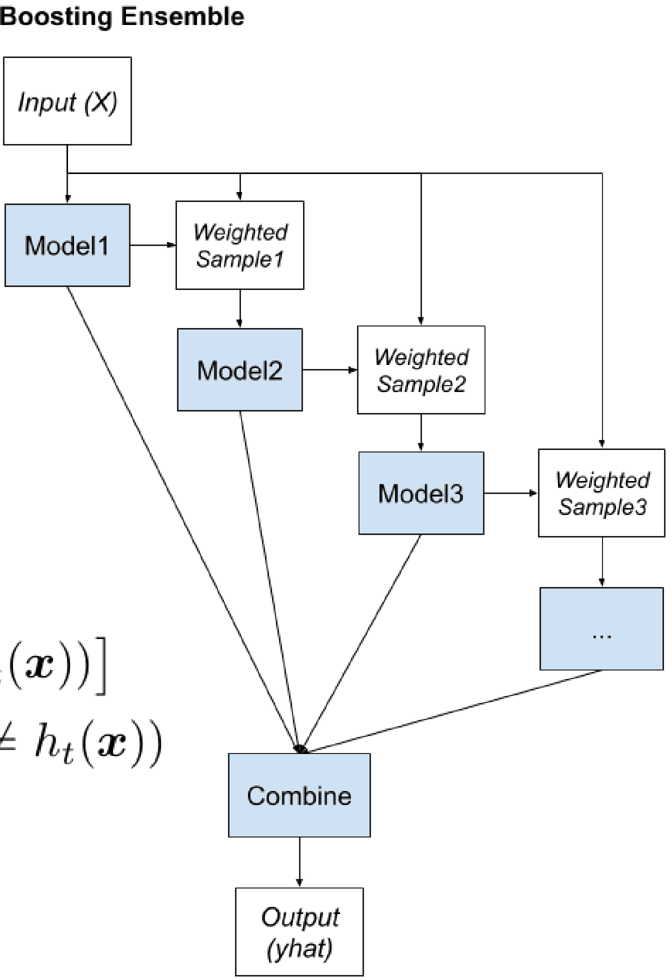
$$\begin{aligned}\mathtt{sign}(H(\boldsymbol{x})) &= \mathtt{sign}\left(\frac{1}{2}\ln\frac{P(f(x) = 1 \mid \boldsymbol{x})}{P(f(x) = -1 \mid \boldsymbol{x})}\right) \\ &= \begin{cases} 1, & P(f(x) = 1 \mid \boldsymbol{x}) > P(f(x) = -1 \mid \boldsymbol{x}) \\ -1, & P(f(x) = 1 \mid \boldsymbol{x}) < P(f(x) = -1 \mid \boldsymbol{x}) \end{cases} \\ &= \underset{y\in\{-1,1\}}{\arg\max}\, P(f(x) = y \mid \boldsymbol{x}),\end{aligned}$$

AdaBoost

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$$ se produce iterativamente.

Cuando entrenamos un *weak learner*, calculamos su coeficiente minimizando:

$$\begin{aligned}
\ell_{\exp}(\alpha_t h_t \mid \mathcal{D}_t) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-\alpha_t}\mathbb{I}(f(\boldsymbol{x}) = h_t(\boldsymbol{x})) + e^{\alpha_t}\mathbb{I}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x}))\right] \\
&= e^{-\alpha_t} P_{\boldsymbol{x}\sim\mathcal{D}_t}(f(\boldsymbol{x}) = h_t(\boldsymbol{x})) + e^{\alpha_t} P_{\boldsymbol{x}\sim\mathcal{D}_t}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})) \\
&= e^{-\alpha_t}(1 - \epsilon_t) + e^{\alpha_t}\epsilon_t
\end{aligned}$$



AdaBoost

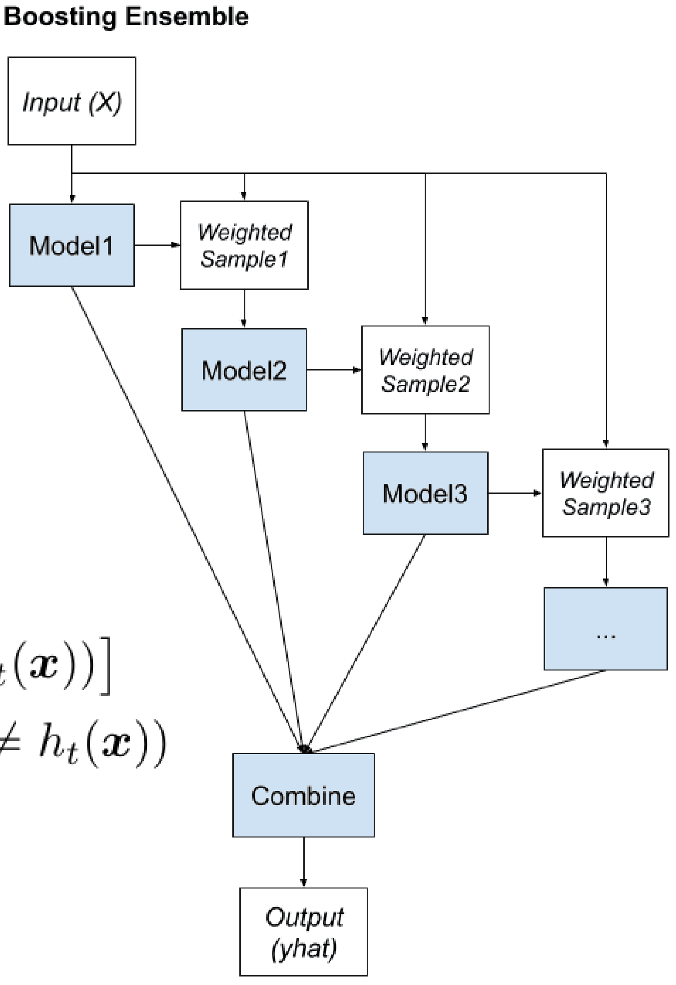
$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$$ se produce iterativamente.

Cuando entrenamos un *weak learner*, calculamos su coeficiente minimizando:

$$\begin{aligned}
\ell_{\exp}(\alpha_t h_t \mid \mathcal{D}_t) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-\alpha_t}\mathbb{I}(f(\boldsymbol{x}) = h_t(\boldsymbol{x})) + e^{\alpha_t}\mathbb{I}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x}))\right] \\
&= e^{-\alpha_t} P_{\boldsymbol{x}\sim\mathcal{D}_t}(f(\boldsymbol{x}) = h_t(\boldsymbol{x})) + e^{\alpha_t} P_{\boldsymbol{x}\sim\mathcal{D}_t}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})) \\
&= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t
\end{aligned}$$

por lo que:

$$\frac{\partial \ell_{\exp}(\alpha_t h_t \mid \mathcal{D}_t)}{\partial \alpha_t} = -e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t = 0$$



AdaBoost

$$H(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x})$$ se produce iterativamente.

Cuando entrenamos un *weak learner*, calculamos su coeficiente minimizando:

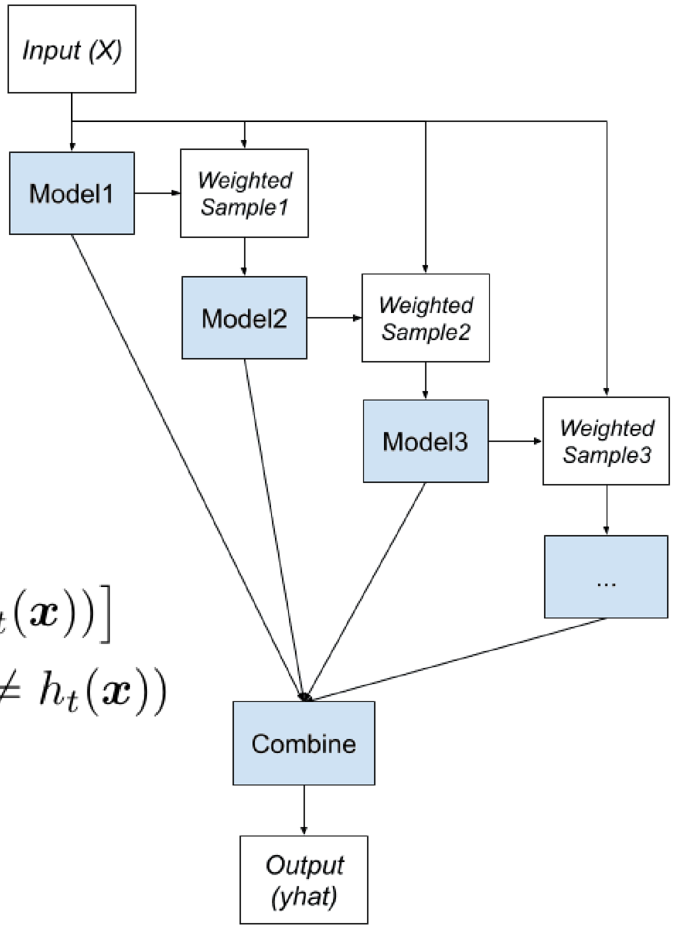
$$\begin{aligned}
\ell_{\exp}(\alpha_t h_t \mid \mathcal{D}_t) &= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}[e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})}] \\
&= \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_t}\left[e^{-\alpha_t}\mathbb{I}(f(\boldsymbol{x}) = h_t(\boldsymbol{x})) + e^{\alpha_t}\mathbb{I}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x}))\right] \\
&= e^{-\alpha_t} P_{\boldsymbol{x}\sim\mathcal{D}_t}(f(\boldsymbol{x}) = h_t(\boldsymbol{x})) + e^{\alpha_t} P_{\boldsymbol{x}\sim\mathcal{D}_t}(f(\boldsymbol{x}) \neq h_t(\boldsymbol{x})) \\
&= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t
\end{aligned}$$

por lo que:

$$\frac{\partial \ell_{\exp}(\alpha_t h_t \mid \mathcal{D}_t)}{\partial \alpha_t} = -e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t = 0$$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Boosting Ensemble



AdaBoost

Ahora vamos a obtener una expresión útil para la distribución:

$$\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}(x)\boxed{e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}]} \longrightarrow \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}e^{-f(\boldsymbol{x})h_t(\boldsymbol{x})}]$$

AdaBoost

Ahora vamos a obtener una expresión útil para la distribución:

$$\mathcal{D}_{t+1}(\boldsymbol{x}) = \frac{\mathcal{D}(x)e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}]} \longrightarrow \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}e^{-f(\boldsymbol{x})h_t(\boldsymbol{x})}]$$

Es decir:

$$\boxed{\mathcal{D}_{t+1}(\boldsymbol{x}) = \mathcal{D}_t(\boldsymbol{x}) \cdot e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})} \frac{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_{t-1}(\boldsymbol{x})}]}{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[e^{-f(\boldsymbol{x})H_t(\boldsymbol{x})}]}}$$

AdaBoost

Unamos las piezas para obtener el algoritmo.

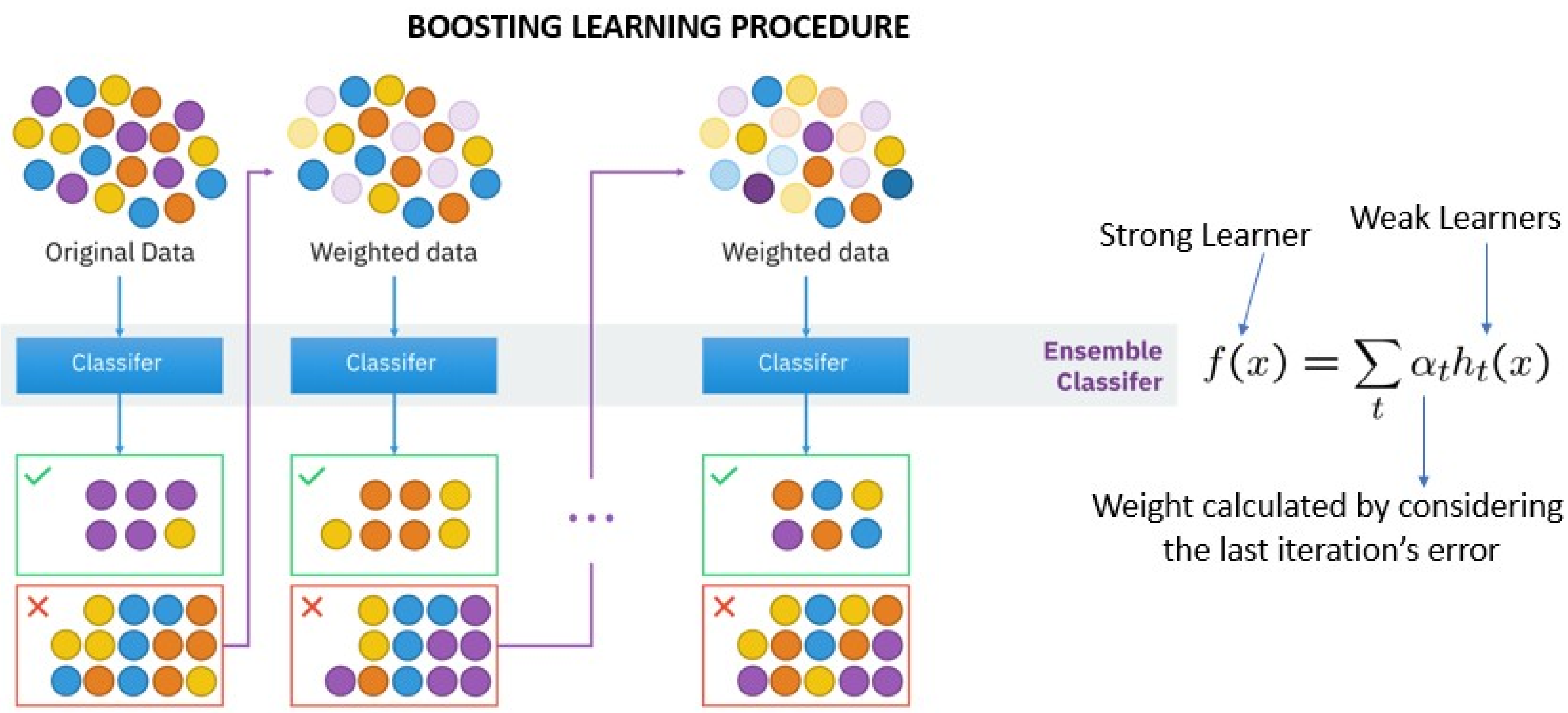
Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
Base learning algorithm $\mathfrak{L}$;
Number of learning rounds T.

Process:

1. $\mathcal{D}_1(\boldsymbol{x}) = 1/m$. % Initialize the weight distribution
2. **for** $t = 1, \ldots, T$:
3. $h_t = \mathfrak{L}(D, \mathcal{D}_t)$; % Train a classifier h_t from D under distribution $\mathcal{D}_t$
4. $\epsilon_t = P_{\boldsymbol{x} \sim \mathcal{D}_t}(h_t(\boldsymbol{x}) \neq f(\boldsymbol{x}))$; % Evaluate the error of h_t
5. **if** $\epsilon_t > 0.5$ **then break**
6. $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$; % Determine the weight of h_t
7. $$\mathcal{D}_{t+1}(\boldsymbol{x}) = \mathcal{D}_t(\boldsymbol{x}) \cdot e^{-f(\boldsymbol{x})\alpha_t h_t(\boldsymbol{x})} \frac{\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[e^{-f(\boldsymbol{x}) H_{t-1}(\boldsymbol{x})}]}{\mathbb{E}_{\boldsymbol{x} \sim \mathcal{D}}[e^{-f(\boldsymbol{x}) H_t(\boldsymbol{x})}]}$$
8. **end**

Output: $H(\boldsymbol{x}) = \mathtt{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \right)$

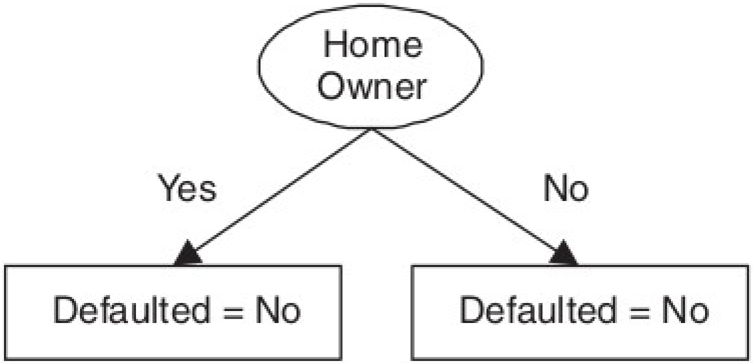
AdaBoost


BOOSTING LEARNING PROCEDURE
Original Data
Weighted data
Weighted data
Classifer
Classifer
Classifer
Ensemble Classifer
Strong Learner
Weak Learners
$f(x) = \sum_t \alpha_t h_t(x)$
Weight calculated by considering the last iteration's error

Weak learner

Weak learner

Árbol de decisión



Objetivo: el split produce nodos puros

Minimizar $$\text{Gini index} = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Fracción de ejemplos de una clase en el nodo

Weak learner

Profundidad máxima

Árbol de decisión

$\texttt{TreeGrowth}\ (E,\ F)$

if $\texttt{stopping\_cond}(E,F) = true$ **then**
    $leaf = \texttt{createNode}()$.
    $leaf.label = \texttt{Classify}(E)$.
    return $leaf$.
else
    $root = \texttt{createNode}()$.
    $root.test\_cond = \texttt{find\_best\_split}(E,\ F)$.
    let $V = \{v | v$ is a possible outcome of $root.test\_cond\ \}$.
    for each $v \in V$ **do**
        $E_v = \{e \mid root.test\_cond(e) = v$ and $e \in E\}$.
        $child = \texttt{TreeGrowth}(E_v,\ F)$.
        add $child$ as descendent of $root$ and label the edge $(root \rightarrow child)$ as v.
    end for
end if
return $root$.

Weak learner

Árbol de decisión