Boosting Adaptivo (AdaBoost)

Fernando Lozano

Universidad de los Andes

12 de septiembre de 2014

Algoritmos de Boosting

- Construir un clasificador fuerte usando múltiples clasificadores débiles.

Algoritmos de Boosting

- Construir un clasificador fuerte usando múltiples clasificadores débiles.
- Clasificador combinado:

Algoritmos de Boosting

- Construir un clasificador fuerte usando múltiples clasificadores débiles.
- Clasificador combinado:
    - Obtener clasificadores $h_1, h_2, \ldots, h_T$ minimizando error en diferentes versiones de los datos.

Algoritmos de Boosting

- Construir un clasificador fuerte usando múltiples clasificadores débiles.
- Clasificador combinado:
  - Obtener clasificadores $h_1, h_2, \ldots, h_T$ minimizando error en diferentes versiones de los datos.
  - Formar combinación:

$$f(\mathbf{x}) = \sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})$$

Algoritmos de Boosting

- Construir un clasificador fuerte usando múltiples clasificadores débiles.
- Clasificador combinado:
  - Obtener clasificadores $h_1, h_2, \ldots, h_T$ minimizando error en diferentes versiones de los datos.
  - Formar combinación:

$$f(\mathbf{x}) = \sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})$$

  - Clasificar con el signo de f

$$h(\mathbf{x}) = \text{signo}(f(\mathbf{x})) = \begin{cases} 1 & \text{si } f(\mathbf{x}) \geq 0 \\ 0 & \text{si } f(\mathbf{x}) < 0 \end{cases},$$

Algoritmos de Boosting

- Construir un clasificador fuerte usando múltiples clasificadores débiles.
- Clasificador combinado:
  - Obtener clasificadores $h_1, h_2, \ldots, h_T$ minimizando error en diferentes versiones de los datos.
  - Formar combinación:

$$f(\mathbf{x}) = \sum_{i=1}^{T} \alpha_i h_i(\mathbf{x})$$

  - Clasificar con el signo de f

$$h(\mathbf{x}) = \text{signo}(f(\mathbf{x})) = \begin{cases} 1 & \text{si } f(\mathbf{x}) \geq 0 \\ 0 & \text{si } f(\mathbf{x}) < 0 \end{cases},$$

- Muy efectivos en la práctica.

Algoritmo de Boosting en el modelo PAC

- Requierer oráculo.

Algoritmo de Boosting en el modelo PAC

- Requierer oráculo.
- Genera estructura no regular.

Algoritmo de Boosting en el modelo PAC

- Requierer oráculo.
- Genera estructura no regular.
- Requiere conocer garantía de error del algoritmo débil.

Algoritmo de Boosting en el modelo PAC

- Requierer oráculo.
- Genera estructura no regular.
- Requiere conocer garantía de error del algoritmo débil.
- No es práctico.

Algoritmo de Boosting en el modelo PAC

- Requierer oráculo.
- Genera estructura no regular.
- Requiere conocer garantía de error del algoritmo débil.
- No es práctico.

Boosting Adaptivo

- Datos $S = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, con $\mathbf{x}_i \in \mathcal{X}$ y $y_i \in \{-1, 1\}$

Boosting Adaptivo

- Datos $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$, con $\mathbf{x}_i \in \mathcal{X}$ y $y_i \in \{-1, 1\}$
- Asociamos a los datos un vector de pesos $D = \{D_1, D_2, \ldots, D_n\}$.

Boosting Adaptivo

- Datos $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$, con $\mathbf{x}_i \in \mathcal{X}$ y $y_i \in \{-1, 1\}$
- Asociamos a los datos un vector de pesos $D = \{D_1, D_2, \ldots, D_n\}$.
- $\{D_1, D_2, \ldots, D_n\}$ es una distribución, es decir

$$D_i \geq 0 \quad \text{y} \quad \sum_{i=1}^{n} D_i = 1$$

Boosting Adaptivo

- Datos $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$, con $\mathbf{x}_i \in \mathcal{X}$ y $y_i \in \{-1, 1\}$
- Asociamos a los datos un vector de pesos $D = \{D_1, D_2, \ldots, D_n\}$.
- $\{D_1, D_2, \ldots, D_n\}$ es una distribución, es decir

$$D_i \geq 0 \quad \text{y} \quad \sum_{i=1}^{n} D_i = 1$$

- Clase de hipótesis base: $h \in \mathcal{H}$, y $h : \mathcal{X} \longrightarrow \{-1, 1\}$

Boosting Adaptivo

- Datos $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$, con $\mathbf{x}_i \in \mathcal{X}$ y $y_i \in \{-1, 1\}$
- Asociamos a los datos un vector de pesos $D = \{D_1, D_2, \ldots, D_n\}$.
- $\{D_1, D_2, \ldots, D_n\}$ es una distribución, es decir

$$D_i \geq 0 \quad \text{y} \quad \sum_{i=1}^{n} D_i = 1$$

- Clase de hipótesis base: $h \in \mathcal{H}$, y $h : \mathcal{X} \longrightarrow \{-1, 1\}$
- Error pesado de una hipótesis h de acuerdo a D:

$$e_D(h) = \sum_{i=1}^{n} D_i I_{\{y_i f(\mathbf{x}_i) \leq 0\}} = \sum_{i\,:\,h(\mathbf{x}_i) \neq y_i} D_i$$

- Asumimos acceso a un aprendiz débil A, que recibe S y D y retorna $h \in \mathcal{H}$ con

$$e_D(h) < \frac{1}{2}$$

- Asumimos acceso a un aprendiz débil A, que recibe S y D y retorna $h \in \mathcal{H}$ con
$$e_D(h) < \frac{1}{2}$$
- AdaBoost procede en una serie de rondas $1, 2, \ldots,$ en las que obtiene hipótesis $h_1, h_2, \ldots$.

- Asumimos acceso a un aprendiz débil A, que recibe S y D y retorna $h \in \mathcal{H}$ con
$$e_D(h) < \frac{1}{2}$$
- AdaBoost procede en una serie de rondas $1, 2, \ldots,$ en las que obtiene hipótesis $h_1, h_2, \ldots$.
- En la primera ronda se llama A con la distribución uniforme $D_i = \frac{1}{n}$, $i = 1, 2, \ldots, n$.

- Asumimos acceso a un aprendiz débil A, que recibe S y D y retorna $h \in \mathcal{H}$ con
$$e_D(h) < \frac{1}{2}$$
- AdaBoost procede en una serie de rondas $1, 2, \ldots,$ en las que obtiene hipótesis $h_1, h_2, \ldots$.
- En la primera ronda se llama A con la distribución uniforme $D_i = \frac{1}{n}$, $i = 1, 2, \ldots, n$.
- En la siguiente ronda se modifica D:

- Asumimos acceso a un aprendiz débil A, que recibe S y D y retorna $h \in \mathcal{H}$ con
$$e_D(h) < \frac{1}{2}$$

- AdaBoost procede en una serie de rondas $1, 2, \ldots,$ en las que obtiene hipótesis $h_1, h_2, \ldots$.
- En la primera ronda se llama A con la distribución uniforme $D_i = \frac{1}{n}$, $i = 1, 2, \ldots, n$.
- En la siguiente ronda se modifica D:

$$D_i \begin{cases} \text{aumenta} & \text{si } h_1(\mathbf{x}) \neq y_i, \\ \text{disminuye} & \text{si } h_1(\mathbf{x}) = y_i. \end{cases}$$

- Asumimos acceso a un aprendiz débil A, que recibe S y D y retorna $h \in \mathcal{H}$ con
$$e_D(h) < \frac{1}{2}$$
- AdaBoost procede en una serie de rondas $1, 2, \ldots$, en las que obtiene hipótesis $h_1, h_2, \ldots$.
- En la primera ronda se llama A con la distribución uniforme $D_i = \frac{1}{n}$, $i = 1, 2, \ldots, n$.
- En la siguiente ronda se modifica D:
$$D_i \begin{cases} \text{aumenta} & \text{si } h_1(\mathbf{x}) \neq y_i, \\ \text{disminuye} & \text{si } h_1(\mathbf{x}) = y_i. \end{cases}$$
- Se itera este procedimiento, modificando los pesos en cada ronda de acuerdo a la hipótesis de la ronda anterior.

AdaBoost

- Construir $f(\mathbf{x})$ para minimizar

$$e(f) = \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

AdaBoost

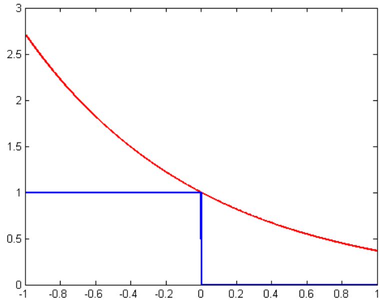
- Construir $f(\mathbf{x})$ para minimizar

$$e(f) = \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \geq \frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i) \leq 0\}}$$

AdaBoost

- Construir $f(\mathbf{x})$ para minimizar

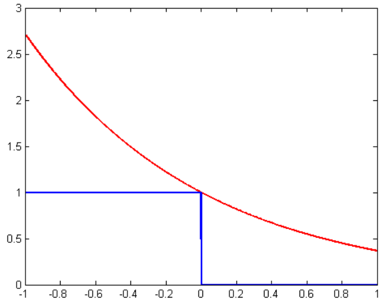
$$e(f) = \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \geq \frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}}$$



AdaBoost

- Construir $f(\mathbf{x})$ para minimizar

$$e(f) = \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \geq \frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i) \leq 0\}}$$



- Maximizar función de costo de los márgenes en los datos.

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

$$e(f_k) = \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

$$
\begin{aligned}
e(f_k) &= \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x}) - y_i \alpha_k h_k(\mathbf{x}_i)}
\end{aligned}
$$

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

$$
\begin{aligned}
e(f_k) &= \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x}) - y_i \alpha_k h_k(\mathbf{x}_i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})} e^{-y_i \alpha_k h_k(\mathbf{x}_i)}
\end{aligned}
$$

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

$$
\begin{aligned}
e(f_k) &= \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x}) - y_i \alpha_k h_k(\mathbf{x}_i)} \\
&= \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})} e^{-y_i \alpha_k h_k(\mathbf{x}_i)} \\
&= \frac{1}{n} \left(\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})} \right) \sum_{i=1}^{n} \left(\frac{e^{-y_i f_{k-1}(\mathbf{x})}}{\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})}} \right) e^{-y_i \alpha_k h_k(\mathbf{x}_i)}
\end{aligned}
$$

- Suponga que conocemos α_j, h_j para $j = 1, 2, \ldots, k-1$, y queremos hallar α_k y h_k. Denote $f_k = \sum_{j=1}^{k} \alpha_j h_j$.

$$
\begin{aligned}
e(f_k) &= \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x}) - y_i \alpha_k h_k(\mathbf{x}_i)} \\
&= \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})} e^{-y_i \alpha_k h_k(\mathbf{x}_i)} \\
&= \frac{1}{n}\left(\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})}\right) \sum_{i=1}^{n} \left(\frac{e^{-y_i f_{k-1}(\mathbf{x})}}{\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})}}\right) e^{-y_i \alpha_k h_k(\mathbf{x}_i)} \\
&= \frac{1}{n}\left(\sum_{i=1}^{n} e^{-y_i f_{k-1}(\mathbf{x})}\right) \sum_{i=1}^{n} D_i e^{-y_i \alpha_k h_k(\mathbf{x}_i)}
\end{aligned}
$$

- Queremos encontrar α, h que minimizan:

$$\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)}$$

- Queremos encontrar α, h que minimizan:

$$\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)}$$

- Queremos encontrar α, h que minimizan:

$$\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)} = \sum_{i:y_i \neq h(\mathbf{x}_i)} D_i e^{\alpha} + \sum_{i:y_i = h(\mathbf{x}_i)} D_i e^{-\alpha}$$

- Queremos encontrar α, h que minimizan:

$$\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)} = \sum_{i:y_i \neq h(\mathbf{x}_i)} D_i e^{\alpha} + \sum_{i:y_i = h(\mathbf{x}_i)} D_i e^{-\alpha}$$

$$= e_D(h) e^{\alpha} + (1 - e_D(h)) e^{-\alpha}$$

- Queremos encontrar α, h que minimizan:

$$\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)} = \sum_{i:y_i \neq h(\mathbf{x}_i)} D_i e^{\alpha} + \sum_{i:y_i = h(\mathbf{x}_i)} D_i e^{-\alpha}$$
$$= e_D(h) e^{\alpha} + (1 - e_D(h)) e^{-\alpha}$$

- $h = \arg\min_{g \in \mathcal{H}} e_D(g)$

- Queremos encontrar α, h que minimizan:

$$\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)} = \sum_{i:y_i \neq h(\mathbf{x}_i)} D_i e^{\alpha} + \sum_{i:y_i = h(\mathbf{x}_i)} D_i e^{-\alpha}$$
$$= e_D(h) e^{\alpha} + (1 - e_D(h)) e^{-\alpha}$$

- $h = \arg\min_{g \in \mathcal{H}} e_D(g) \Longrightarrow$ Aprendiz débil

- Queremos encontrar α, h que minimizan:

$$\begin{aligned}\sum_{i=1}^{n} D_i e^{-y_i \alpha h(\mathbf{x}_i)} &= \sum_{i:y_i \neq h(\mathbf{x}_i)} D_i e^{\alpha} + \sum_{i:y_i = h(\mathbf{x}_i)} D_i e^{-\alpha} \\ &= e_D(h) e^{\alpha} + (1 - e_D(h)) e^{-\alpha}\end{aligned}$$

- $h = \arg\min_{g \in \mathcal{H}} e_D(g) \Longrightarrow$ Aprendiz débil
- Con h fija, encontramos α derivando e igualando a cero:

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - e_D}{e_D} \right)$$

Algorithm 1 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.

Algorithm 2 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.

for $t = 1$ to T **do**

Algorithm 3 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.

for $t = 1$ to T **do**

    $h_t \leftarrow A(S, D_t)$.

Algorithm 4 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \dots n$.

for $t = 1$ to T **do**

$h_t \leftarrow A(S, D_t)$.

$\epsilon_t = \sum_{i: h_t(X_i) \neq y_i} D_t(i)$.

Algorithm 5 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \dots n$.

for $t = 1$ to T **do**

$h_t \leftarrow A(S, D_t)$.

$\epsilon_t = \sum_{i:h_t(X_i) \neq y_i} D_t(i)$.

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

Algorithm 6 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.

for $t = 1$ to T **do**

$h_t \leftarrow A(S, D_t)$.

$\epsilon_t = \sum_{i:h_t(X_i) \neq y_i} D_t(i)$.

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

Actualice D: $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

Algorithm 7 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \dots n$.

for $t = 1$ to T **do**

$h_t \leftarrow A(S, D_t)$.

$\epsilon_t = \sum_{i:h_t(X_i) \neq y_i} D_t(i)$.

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

Actualice D: $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

Donde Z_t normaliza D de manera que $\sum_{i=1}^{t} D_{t+1}(i) = 1$.

Algorithm 8 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.
for $t = 1$ to T **do**
  $h_t \leftarrow A(S, D_t)$.
  $\epsilon_t = \sum_{i:h_t(X_i)\neq y_i} D_t(i)$.
  $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
  Actualice D: $D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
  Donde Z_t normaliza D de manera que $\sum_{i=1}^{t} D_{t+1}(i) = 1$.
end for

Algorithm 9 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.
for $t = 1$ to T **do**
  $h_t \leftarrow A(S, D_t)$.
  $\epsilon_t = \sum_{i:h_t(X_i) \neq y_i} D_t(i)$.
  $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$
  Actualice D: $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
  Donde Z_t normaliza D de manera que $\sum_{i=1}^{t} D_{t+1}(i) = 1$.
end for
Retorne $f(x) = \sum_{i=1}^{T} \alpha_t h_t(x)$

Algorithm 10 AdaBoost

$D_1(i) = 1/n$ para $i = 1 \ldots n$.

for $t = 1$ to T **do**

$h_t \leftarrow A(S, D_t)$.

$\epsilon_t = \sum_{i:h_t(X_i) \neq y_i} D_t(i)$.

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

Actualice D: $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

Donde Z_t normaliza D de manera que $\sum_{i=1}^{t} D_{t+1}(i) = 1$.

end for

Retorne $f(x) = \sum_{i=1}^{T} \alpha_t h_t(x)$

- Note que

$$D_{t+1}(i) = \begin{cases} \frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i) \ , \\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i) \ . \end{cases}$$

- Note que

$$D_{t+1}(i) = \begin{cases} \frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i) \ , \\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i) \ . \end{cases}$$

- El error pesado de h_t con respecto a D_{t+1}:

- Note que

$$D_{t+1}(i) = \begin{cases} \frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i)\ , \\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i)\ . \end{cases}$$

- El error pesado de h_t con respecto a D_{t+1}:

$$e_{D_{t+1}}(h) = \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} D_{t+1}(i)$$

- Note que

$$D_{t+1}(i) = \begin{cases} \frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i) \ , \\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i) \ . \end{cases}$$

- El error pesado de h_t con respecto a D_{t+1}:

$$\begin{aligned} e_{D_{t+1}}(h) &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} D_{t+1}(i) \\ &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} \end{aligned}$$

- Note que

$$D_{t+1}(i)=\begin{cases}\frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i)\ ,\\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i)\ .\end{cases}$$

- El error pesado de h_t con respecto a D_{t+1}:

$$\begin{aligned} e_{D_{t+1}}(h) &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} D_{t+1}(i) \\ &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} \\ &= \frac{\epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{\epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}+(1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}} \end{aligned}$$

- Note que

$$D_{t+1}(i) = \begin{cases} \frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i) \ , \\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i) \ . \end{cases}$$

- El error pesado de h_t con respecto a D_{t+1}:

$$\begin{aligned} e_{D_{t+1}}(h) &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} D_{t+1}(i) \\ &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} \\ &= \frac{\epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{\epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + (1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}} \\ &= \frac{\sqrt{\epsilon_t(1-\epsilon_t)}}{\sqrt{\epsilon_t(1-\epsilon_t)} + \sqrt{\epsilon_t(1-\epsilon_t)}} \end{aligned}$$

- Note que

$$D_{t+1}(i) = \begin{cases} \frac{D_t(i)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}}{Z_t} & \text{si } y_i = h_t(\mathbf{x}_i) \ , \\ \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} & \text{si } y_i \neq h_t(\mathbf{x}_i) \ . \end{cases}$$

- El error pesado de h_t con respecto a D_{t+1}:

$$\begin{aligned} e_{D_{t+1}}(h) &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} D_{t+1}(i) \\ &= \sum_{i\,:\,h(\mathbf{x}_i)\neq y_i} \frac{D_t(i)\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{Z_t} \\ &= \frac{\epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{\epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}} + (1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}}} \\ &= \frac{\sqrt{\epsilon_t(1-\epsilon_t)}}{\sqrt{\epsilon_t(1-\epsilon_t)} + \sqrt{\epsilon_t(1-\epsilon_t)}} = \frac{1}{2} \end{aligned}$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}}$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

y

$$D_{t+1}(i) = \frac{e^{-\sum_t \alpha_t y_i h_t(\mathbf{x}_i)}}{n\prod_t Z_t}$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

y

$$D_{t+1}(i) = \frac{e^{-\sum_t \alpha_t y_i h_t(\mathbf{x}_i)}}{n\prod_t Z_t} = \frac{e^{-y_i f(\mathbf{x}_i)}}{n\prod_t Z_t}$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

y

$$D_{t+1}(i) = \frac{e^{-\sum_t \alpha_t y_i h_t(\mathbf{x}_i)}}{n\prod_t Z_t} = \frac{e^{-y_i f(\mathbf{x}_i)}}{n\prod_t Z_t}$$

luego:

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \prod_t Z_t$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

y

$$D_{t+1}(i) = \frac{e^{-\sum_t \alpha_t y_i h_t(\mathbf{x}_i)}}{n\prod_t Z_t} = \frac{e^{-y_i f(\mathbf{x}_i)}}{n\prod_t Z_t}$$

luego:

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \prod_t Z_t = \prod_t 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

y

$$D_{t+1}(i) = \frac{e^{-\sum_t \alpha_t y_i h_t(\mathbf{x}_i)}}{n\prod_t Z_t} = \frac{e^{-y_i f(\mathbf{x}_i)}}{n\prod_t Z_t}$$

luego:

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \prod_t Z_t = \prod_t 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

- Si $\epsilon_t < \frac{1}{2}$ el error empírico decrece exponencialmente!

Error empírico

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)}$$

y

$$D_{t+1}(i) = \frac{e^{-\sum_t \alpha_t y_i h_t(\mathbf{x}_i)}}{n\prod_t Z_t} = \frac{e^{-y_i f(\mathbf{x}_i)}}{n\prod_t Z_t}$$

luego:

$$\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} \leq \prod_t Z_t = \prod_t 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

- Si $\epsilon_t < \frac{1}{2}$ el error empírico decrece exponencialmente!
- Si $\epsilon_t < \frac{1}{2}$ el error empírico llega a cero en un número finito de pasos.

Consecuencias

- Sea $\mathcal{F} = \text{conv}(\mathcal{H}) = \left\{ f = \sum_{i=1}^{T} \alpha_t h_t \ : \ h \in \mathcal{H}, \ \alpha_t \geq 0, \sum_t \alpha_t = 1 \right\}$

Consecuencias

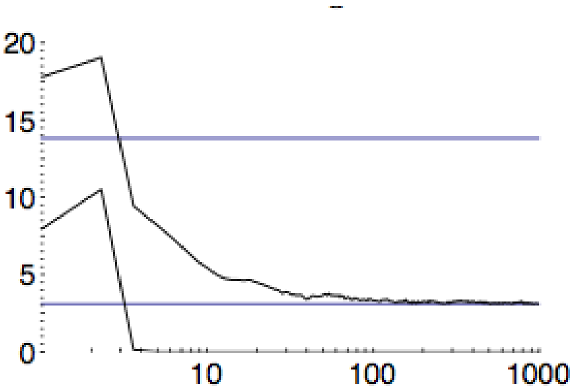
- Sea $\mathcal{F} = \text{conv}(\mathcal{H}) = \left\{ f = \sum_{i=1}^{T} \alpha_t h_t \ : \ h \in \mathcal{H}, \ \alpha_t \geq 0, \sum_t \alpha_t = 1 \right\}$
- $VC(\mathcal{F}) \leq 2(VC(\mathcal{H}) + 1)(T + 1) log_2(e(T + 1))$

Consecuencias

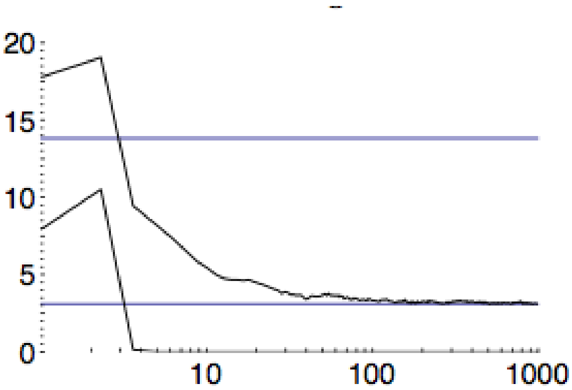
- Sea $\mathcal{F} = \text{conv}(\mathcal{H}) = \left\{ f = \sum_{i=1}^{T} \alpha_t h_t \ : \ h \in \mathcal{H}, \ \alpha_t \geq 0, \sum_t \alpha_t = 1 \right\}$
- $VC(\mathcal{F}) \leq 2(VC(\mathcal{H}) + 1)(T + 1) log_2(e(T + 1))$
- Si $VC(\mathcal{H}) < \infty$ entonces AdaBoost hace boosting en el modelo PAC.

Consecuencias

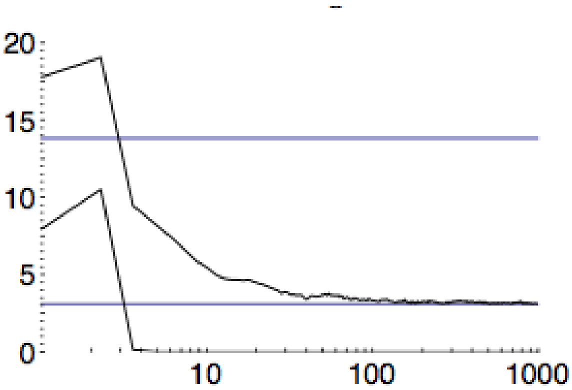
- Sea $\mathcal{F} = \mathrm{conv}(\mathcal{H}) = \left\{ f = \sum_{i=1}^{T} \alpha_t h_t \ : \ h \in \mathcal{H}, \ \alpha_t \geq 0, \sum_t \alpha_t = 1 \right\}$
- $VC(\mathcal{F}) \leq 2(VC(\mathcal{H}) + 1)(T + 1) log_2(e(T + 1))$
- Si $VC(\mathcal{H}) < \infty$ entonces AdaBoost hace boosting en el modelo PAC.
- Sobreajuste de los datos de entrenamiento?



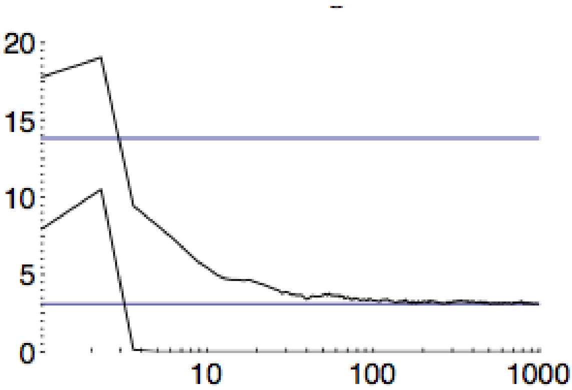
20
15
10
5
0
10
100
1000



- Incluso cuando el error empírico es cero, error real disminuye.



- Incluso cuando el error empírico es cero, error real disminuye.
- Eventualmente puede producir sobreajuste para $T \ggg$.



- Incluso cuando el error empírico es cero, error real disminuye.
- Eventualmente puede producir sobreajuste para $T \ggg$.
- Sobreajuste con datos ruidosos.

Márgenes

Márgenes

- El márgen de f en $(\mathbf{x}_i, y_i)$ es $m_i = y_i f(\mathbf{x}_i)$.

Márgenes

- El márgen de f en $(\mathbf{x}_i, y_i)$ es $m_i = y_i f(\mathbf{x}_i)$.
- $(\mathbf{x}_i, y_i)$ es incorrectamente clasificado si $m_i < 0$.

Márgenes

- El márgen de f en $(\mathbf{x}_i, y_i)$ es $m_i = y_i f(\mathbf{x}_i)$.
- $(\mathbf{x}_i, y_i)$ es incorrectamente clasificado si $m_i < 0$.
- Si $m_i > 0$, podemos interprentar $|m_i|$ como una medida de confianza.

Márgenes

- El márgen de f en $(\mathbf{x}_i, y_i)$ es $m_i = y_i f(\mathbf{x}_i)$.
- $(\mathbf{x}_i, y_i)$ es incorrectamente clasificado si $m_i < 0$.
- Si $m_i > 0$, podemos interprentar $|m_i|$ como una medida de confianza.
- AdaBoost intenta minimizar una función de costo del márgen:

$$\phi(m_1, \ldots, m_n) = \frac{1}{n} \sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} = \frac{1}{n} \sum_{i=1}^{n} e^{-m_i}$$

Generalización de AdaBoost

Theorem (Schapire, Freund, Bartlett y Lee, 1998)

$\forall \alpha \in (0,1)$ *with probability at least* $1-\alpha$ *for all* $f \in \mathrm{conv}(\mathcal{H})$ *the following inequality holds:*

$$P\{(x,y) : yf(x) \leq 0\} \leq$$

$$\inf_{\delta} \left[P_n\{(x,y) : yf(x) \leq \delta\} + \frac{C}{\sqrt{n}} \left(\frac{V(\mathcal{H}) \log^2(\frac{n}{V(\mathcal{H})})}{\delta^2} + \log(1/\alpha) \right)^{1/2} \right].$$

Generalización de AdaBoost

Theorem (Schapire, Freund, Bartlett y Lee, 1998)

$\forall \alpha \in (0,1)$ *with probability at least* $1-\alpha$ *for all* $f \in \mathrm{conv}(\mathcal{H})$ *the following inequality holds:*

$$P\{(x,y) : yf(x) \le 0\} \le$$

$$\inf_{\delta}\left[P_n\{(x,y) : yf(x) \le \delta\} + \frac{C}{\sqrt{n}}\left(\frac{V(\mathcal{H})\log^2(\frac{n}{V(\mathcal{H})})}{\delta^2} + \log(1/\alpha)\right)^{1/2}\right].$$

- Un clasificador combinado con márgenes grandes puede tener probabilidad de error pequeña.

Efecto de Adaboost en los márgenes