Boosting Adaptivo (AdaBoost)

Fernando Lozano

Universidad de los Andes

16 de septiembre de 2014

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 $i = 1 \ldots n$.

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

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

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

Algorithm 5 AdaBoost

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

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

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

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

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

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 $i = 1 \ldots n$.

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 $\sum_{i=1}^{t} D_{t+1}(i) = 1$.

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

Boosting usando predictores con confianza

Boosting usando predictores con confianza

- Deducción de AdaBoost para clasificadores binarios.

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \{-1, 1\}$$

Boosting usando predictores con confianza

- Deducción de AdaBoost para clasificadores binarios.

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \{-1, 1\}$$

- Predictores con confianza:

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \mathbb{R}$$

Boosting usando predictores con confianza

- Deducción de AdaBoost para clasificadores binarios.

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \{-1, 1\}$$

- Predictores con confianza:

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \mathbb{R}$$

  - Predice clase con $\text{sign}(h(x))$.

Boosting usando predictores con confianza

- Deducción de AdaBoost para clasificadores binarios.

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \{-1, 1\}$$

- Predictores con confianza:

$$h \in \mathcal{H}, \quad h : \mathcal{X} \to \mathbb{R}$$

  - Predice clase con $\text{sign}(h(x))$.
  - $|h(x)|$ es confianza en la predicción.

Algorithm 11 AdaBoost

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

Algorithm 12 AdaBoost

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

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

Algorithm 13 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 14 AdaBoost

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

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

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

Escoja α_t

Algorithm 15 AdaBoost

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

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

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

Escoja α_t

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

Algorithm 16 AdaBoost

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

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

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

Escoja α_t

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 17 AdaBoost

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

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

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

Escoja α_t

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 18 AdaBoost

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

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

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

Escoja α_t

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 19 AdaBoost

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

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

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

Escoja α_t

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

Teorema

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

Teorema

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

Demostración.

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

Teorema

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

Demostración.

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

Teorema

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

Demostración.

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

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

Teorema

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

Demostración.

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

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} &\leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \\ &= \sum_{i=1}^{n} D_{t+1}(i)\left(\prod_{t=1}^{T} Z_t\right)\end{aligned}$$

Teorema

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

Demostración.

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

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} I_{\{y_i f(\mathbf{x}_i)\leq 0\}} &\leq \frac{1}{n}\sum_{i=1}^{n} e^{-y_i f(\mathbf{x}_i)} \\ &= \sum_{i=1}^{n} D_{t+1}(i)\left(\prod_{t=1}^{T} Z_t\right) = \prod_{t=1}^{T} Z_t\end{aligned}$$

Cómo escoger α_t?

Cómo escoger α_t?

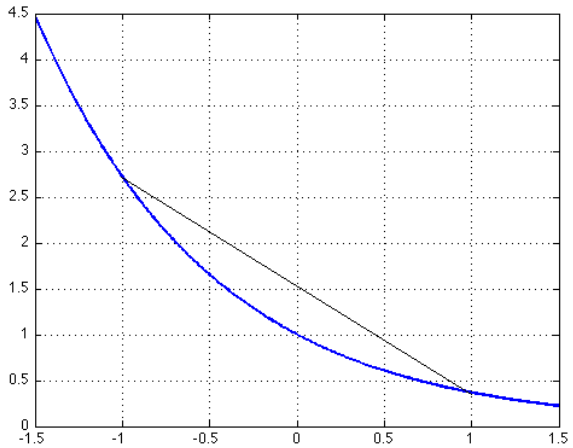
- Idea: Minimizar $\prod_{t=1}^{T} Z_t$ de manera greedy.

Cómo escoger α_t?

- Idea: Minimizar $\prod_{t=1}^{T} Z_t$ de manera greedy.
- Sea $u_i \equiv y_i h_t(x_i)$, $Z \equiv Z_t$, $D \equiv D_t$, $\alpha \equiv \alpha_t$

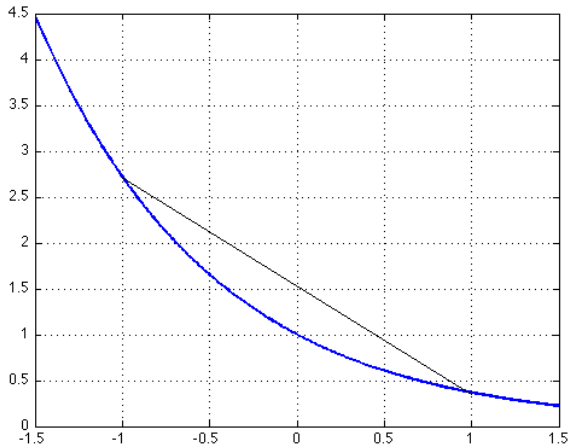
Cómo escoger α_t?

- Idea: Minimizar $\prod_{t=1}^{T} Z_t$ de manera greedy.
- Sea $u_i \equiv y_i h_t(x_i)$, $Z \equiv Z_t$, $D \equiv D_t$, $\alpha \equiv \alpha_t$
- e^{-z} es convexa:



Cómo escoger α_t?

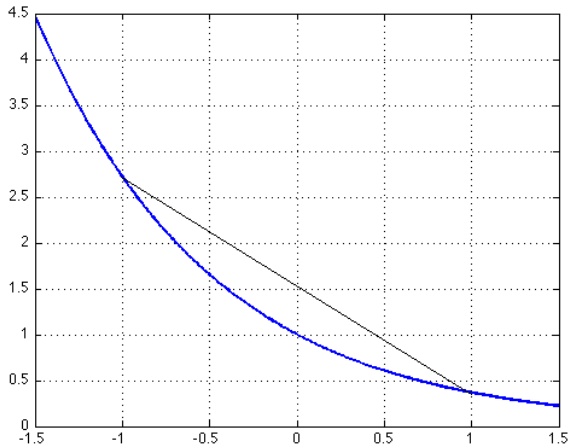
- Idea: Minimizar $\prod_{t=1}^{T} Z_t$ de manera greedy.
- Sea $u_i \equiv y_i h_t(x_i)$, $Z \equiv Z_t$, $D \equiv D_t$, $\alpha \equiv \alpha_t$
- e^{-z} es convexa:



$$Z = \sum_{i=1}^{n} D(i) e^{-\alpha u_i}$$

Cómo escoger α_t?

- Idea: Minimizar $\prod_{t=1}^{T} Z_t$ de manera greedy.
- Sea $u_i \equiv y_i h_t(x_i)$, $Z \equiv Z_t$, $D \equiv D_t$, $\alpha \equiv \alpha_t$
- e^{-z} es convexa:



$$Z = \sum_{i=1}^{n} D(i) e^{-\alpha u_i} \leq \sum_{i=1}^{n} D(i) \left(\frac{1+u_i}{2} e^{-\alpha} + \frac{1-u_i}{2} e^{\alpha} \right)$$

- Minimizando con respecto a α:

$$\alpha = \frac{1}{2} \ln \left(\frac{\sum_{i=1}^{n} D(i) \frac{1+u_i}{2}}{\sum_{i=1}^{n} D(i) \frac{1-u_i}{2}} \right)$$

- Minimizando con respecto a α:

$$\alpha = \frac{1}{2} \ln \left(\frac{\sum_{i=1}^{n} D(i) \frac{1+u_i}{2}}{\sum_{i=1}^{n} D(i) \frac{1-u_i}{2}} \right) = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right)$$

donde $r = \sum_{i=1}^{n} D(i) u_i$ es correlación entre pesos y márgenes.

- Minimizando con respecto a α:

$$\alpha = \frac{1}{2}\ln\left(\frac{\sum_{i=1}^{n} D(i)\frac{1+u_i}{2}}{\sum_{i=1}^{n} D(i)\frac{1-u_i}{2}}\right) = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)$$

  donde $r = \sum_{i=1}^{n} D(i)u_i$ es correlación entre pesos y márgenes.

- Reemplazando:

$$Z \leq \sum_{i=1}^{n} D(i)\left(\frac{1+u_i}{2}e^{-\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)} + \frac{1-u_i}{2}e^{\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)}\right)$$

- Minimizando con respecto a α:

$$\alpha = \frac{1}{2}\ln\left(\frac{\sum_{i=1}^{n} D(i)\frac{1+u_i}{2}}{\sum_{i=1}^{n} D(i)\frac{1-u_i}{2}}\right) = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)$$

  donde $r = \sum_{i=1}^{n} D(i)u_i$ es correlación entre pesos y márgenes.

- Reemplazando:

$$\begin{aligned} Z &\leq \sum_{i=1}^{n} D(i)\left(\frac{1+u_i}{2}e^{-\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)} + \frac{1-u_i}{2}e^{\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)}\right) \\ &= \frac{1}{2}(1+r)\sqrt{\frac{1-r}{1+r}} + \frac{1}{2}(1-r)\sqrt{\frac{1+r}{1-r}} \end{aligned}$$

- Minimizando con respecto a α:

$$\alpha = \frac{1}{2}\ln\left(\frac{\sum_{i=1}^{n} D(i)\frac{1+u_i}{2}}{\sum_{i=1}^{n} D(i)\frac{1-u_i}{2}}\right) = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)$$

  donde $r = \sum_{i=1}^{n} D(i)u_i$ es correlación entre pesos y márgenes.

- Reemplazando:

$$Z \leq \sum_{i=1}^{n} D(i)\left(\frac{1+u_i}{2}e^{-\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)} + \frac{1-u_i}{2}e^{\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)}\right)$$

$$= \frac{1}{2}(1+r)\sqrt{\frac{1-r}{1+r}} + \frac{1}{2}(1-r)\sqrt{\frac{1+r}{1-r}} = \sqrt{1-r^2}$$

- Minimizando con respecto a α:

$$\alpha = \frac{1}{2}\ln\left(\frac{\sum_{i=1}^{n} D(i)\frac{1+u_i}{2}}{\sum_{i=1}^{n} D(i)\frac{1-u_i}{2}}\right) = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)$$

  donde $r = \sum_{i=1}^{n} D(i)u_i$ es correlación entre pesos y márgenes.

- Reemplazando:

$$Z \leq \sum_{i=1}^{n} D(i)\left(\frac{1+u_i}{2}e^{-\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)} + \frac{1-u_i}{2}e^{\frac{1}{2}\ln\left(\frac{1+r}{1-r}\right)}\right)$$

$$= \frac{1}{2}(1+r)\sqrt{\frac{1-r}{1+r}} + \frac{1}{2}(1-r)\sqrt{\frac{1+r}{1-r}} = \sqrt{1-r^2}$$

- Obtener h_t maximizando r_t.

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i)e^{-\alpha u_i}$.

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i)e^{-\alpha u_i}$.
- Derivando:

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i) e^{-\alpha u_i}$.
- Derivando:

$$\frac{dZ}{d\alpha} =$$

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i)e^{-\alpha u_i}$.
- Derivando:

$$\frac{dZ}{d\alpha} = -\sum_{i=1}^{n} D(i)u_i e^{-\alpha u_i}$$

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i)e^{-\alpha u_i}$.
- Derivando:

$$\frac{dZ}{d\alpha} = -\sum_{i=1}^{n} D(i)u_i e^{-\alpha u_i} = -Z(\alpha)\sum_{i=1}^{n} D_{t+1}(i)u_i$$

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i)e^{-\alpha u_i}$.
- Derivando:

$$\frac{dZ}{d\alpha} = -\sum_{i=1}^{n} D(i)u_i e^{-\alpha u_i} = -Z(\alpha)\sum_{i=1}^{n} D_{t+1}(i)u_i$$

- Para minimizar $Z(\alpha)$ se escoge α de tal forma que

$$\mathbf{E}_{D_{t+1}}\left[y_i h_t(x_i)\right] = \sum_{i=1}^{n} D_{t+1}u_i = 0$$

Caso General

- Sea $Z(\alpha) = \sum_{i=1}^{n} D(i)e^{-\alpha u_i}$.
- Derivando:

$$\frac{dZ}{d\alpha} = -\sum_{i=1}^{n} D(i)u_i e^{-\alpha u_i} = -Z(\alpha)\sum_{i=1}^{n} D_{t+1}(i)u_i$$

- Para minimizar $Z(\alpha)$ se escoge α de tal forma que

$$\mathbf{E}_{D_{t+1}}\left[y_i h_t(x_i)\right] = \sum_{i=1}^{n} D_{t+1}u_i = 0$$

Es decir, la correlación entre las etiquetas y la predicción con respecto a la nueva distribución es cero.

Ejemplo: Hipótesis con abstención

Ejemplo: Hipótesis con abstención

- $h_t(x) \in \{-1, 0, 1\}$.

Ejemplo: Hipótesis con abstención

- $h_t(x) \in \{-1, 0, 1\}$.
- Sea $W_0 = \sum_{i\,:\,u_1=0} D(i)$, $W_+ = \sum_{i\,:\,u_1=1} D(i)$, $W_- = \sum_{i\,:\,u_1=-1} D(i)$,

Ejemplo: Hipótesis con abstención

- $h_t(x) \in \{-1, 0, 1\}$.
- Sea $W_0 = \sum_{i\,:\,u_1=0} D(i)$, $W_+ = \sum_{i\,:\,u_1=1} D(i)$, $W_- = \sum_{i\,:\,u_1=-1} D(i)$,

$$Z = \sum_{i=1}^{n} D(i)e^{-\alpha u_i} = W_0 + W_- e^{\alpha} + W_+ e^{-\alpha}$$

Ejemplo: Hipótesis con abstención

- $h_t(x) \in \{-1, 0, 1\}$.
- Sea $W_0 = \sum_{i\,:\,u_1=0} D(i)$, $W_+ = \sum_{i\,:\,u_1=1} D(i)$, $W_- = \sum_{i\,:\,u_1=-1} D(i)$,

$$Z = \sum_{i=1}^{n} D(i)e^{-\alpha u_i} = W_0 + W_- e^{\alpha} + W_+ e^{-\alpha}$$

- Derivando e igualando a cero,

Ejemplo: Hipótesis con abstención

- $h_t(x) \in \{-1, 0, 1\}$.
- Sea $W_0 = \sum_{i\,:\,u_1=0} D(i)$, $W_+ = \sum_{i\,:\,u_1=1} D(i)$, $W_- = \sum_{i\,:\,u_1=-1} D(i)$,

$$Z = \sum_{i=1}^{n} D(i)e^{-\alpha u_i} = W_0 + W_- e^{\alpha} + W_+ e^{-\alpha}$$

- Derivando e igualando a cero,

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

y $Z = W_0 + 2\sqrt{W_+ W_-}$

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.
- Multiclase:
  - (x, y), $y \in \mathcal{Y}$.

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.
- Multiclase:
  - (x, y), $y \in \mathcal{Y}$.
  - Meta de aprendizaje: minimizar $\mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$.

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.
- Multiclase:
  - (x, y), $y \in \mathcal{Y}$.
  - Meta de aprendizaje: minimizar $\mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$.
- Multiclase y multietiqueta:

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.
- Multiclase:
  - (x, y), $y \in \mathcal{Y}$.
  - Meta de aprendizaje: minimizar $\mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$.
- Multiclase y multietiqueta:
  - (x, Y), $Y \subseteq \mathcal{Y}$.

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.
- Multiclase:
  - (x, y), $y \in \mathcal{Y}$.
  - Meta de aprendizaje: minimizar $\mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$.
- Multiclase y multietiqueta:
  - (x, Y), $Y \subseteq \mathcal{Y}$.
  - Meta?

Problemas multiclase y multietiqueta

- $\mathcal{Y}$: posibles etiquetas.
- $|\mathcal{Y}| = k \geq 2$.
- Multiclase:
  - (x, y), $y \in \mathcal{Y}$.
  - Meta de aprendizaje: minimizar $\mathbf{P}_{\mathcal{D}}\left[h(x) \neq y\right]$.
- Multiclase y multietiqueta:
  - (x, Y), $Y \subseteq \mathcal{Y}$.
  - Meta? depende del problema.

Distancia de Hamming

- Hipótesis $h : \mathcal{X} \to 2^{\mathcal{Y}}$

Distancia de Hamming

- Hipótesis $h : \mathcal{X} \to 2^{\mathcal{Y}}$
- Función de pérdida de Hamming:

$$\underset{\mathcal{D}}{\text{hloss}}(h) = \frac{1}{k}\mathbf{E}_{\mathcal{D}}\left[|h(x)\triangle Y|\right]$$

Distancia de Hamming

- Hipótesis $h : \mathcal{X} \to 2^{\mathcal{Y}}$
- Función de pérdida de Hamming:

$$\underset{\mathcal{D}}{\text{hloss}}(h) = \frac{1}{k}\mathbf{E}_{\mathcal{D}}\left[|h(x) \triangle Y|\right]$$

- Promedio del error en k problemas binarios.

Distancia de Hamming

- Hipótesis $h : \mathcal{X} \to 2^{\mathcal{Y}}$
- Función de pérdida de Hamming:

$$\underset{\mathcal{D}}{\text{hloss}}(h) = \frac{1}{k}\mathbf{E}_{\mathcal{D}}\left[|h(x)\triangle Y|\right]$$

- Promedio del error en k problemas binarios.
- Para $Y \subseteq \mathcal{Y}$ definimos:

$$Y[l] = \begin{cases} +1 & \text{si } l \in Y \\ -1 & \text{si } l \notin Y \end{cases}$$

Distancia de Hamming

- Hipótesis $h : \mathcal{X} \to 2^{\mathcal{Y}}$
- Función de pérdida de Hamming:

$$\underset{\mathcal{D}}{\text{hloss}}(h) = \frac{1}{k}\mathbf{E}_{\mathcal{D}}\left[|h(x) \triangle Y|\right]$$

- Promedio del error en k problemas binarios.
- Para $Y \subseteq \mathcal{Y}$ definimos:

$$Y[l] = \begin{cases} +1 & \text{si } l \in Y \\ -1 & \text{si } l \notin Y \end{cases}$$

- Identificamos función $h : \mathcal{X} \to 2^{\mathcal{Y}}$ con $h : \mathcal{X} \times \mathcal{Y} \to \{-1, +1\}$, con $h(x, l) = h(x)[l]$.

Distancia de Hamming

- Hipótesis $h : \mathcal{X} \to 2^{\mathcal{Y}}$
- Función de pérdida de Hamming:

$$\underset{\mathcal{D}}{\text{hloss}}(h) = \frac{1}{k}\mathbf{E}_{\mathcal{D}}\left[|h(x)\triangle Y|\right]$$

- Promedio del error en k problemas binarios.
- Para $Y \subseteq \mathcal{Y}$ definimos:

$$Y[l] = \begin{cases} +1 & \text{si } l \in Y \\ -1 & \text{si } l \notin Y \end{cases}$$

- Identificamos función $h : \mathcal{X} \to 2^{\mathcal{Y}}$ con $h : \mathcal{X} \times \mathcal{Y} \to \{-1, +1\}$, con $h(x, l) = h(x)[l]$.
- Dato $(x_i, Y_i) \longrightarrow k$ datos $((x_i, l), Y_i[l])$

Algorithm 20 AdaBoost.MH

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

Algorithm 21 AdaBoost.MH

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

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

Algorithm 22 AdaBoost.MH

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

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

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

Algorithm 23 AdaBoost.MH

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

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

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

Escoja α_t

Algorithm 24 AdaBoost.MH

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

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

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

Escoja α_t

Actualice D: $D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t}$

Algorithm 25 AdaBoost.MH

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

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

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

Escoja α_t

Actualice D: $D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

Algorithm 26 AdaBoost.MH

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

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

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

Escoja α_t

Actualice D: $D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

end for

Algorithm 27 AdaBoost.MH

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

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

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

Escoja α_t

Actualice D: $D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

end for

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

Algorithm 28 AdaBoost.MH

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

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

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

Escoja α_t

Actualice D: $D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

end for

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

Teorema

$$\underset{\mathcal{D}}{\text{hloss}}(f) \leq \prod_{t=1}^{T} Z_t$$

Escogencia de α_t y h_t

Escogencia de α_t y h_t

- Si h_t es binaria, $\alpha_t = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right)$, donde

$$r_t = \sum_{l=1}^{k} \sum_{i=1}^{n} D_t(i,l) Y_i[l] h_t(x_i, l)$$

Escogencia de α_t y h_t

- Si h_t es binaria, $\alpha_t = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right)$, donde

$$r_t = \sum_{l=1}^{k} \sum_{i=1}^{n} D_t(i,l) Y_i[l] h_t(x_i, l)$$

- $Z_t = \sqrt{1 - r_t^2}$

Escogencia de α_t y h_t

- Si h_t es binaria, $\alpha_t = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right)$, donde

$$r_t = \sum_{l=1}^{k} \sum_{i=1}^{n} D_t(i,l) Y_i[l] h_t(x_i, l)$$

- $Z_t = \sqrt{1 - r_t^2} \Rightarrow$ clasificador débil maximiza $|r_t|$

Clasificación multiclase usando ranking

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
- Para un x dado la etiqueta l_1 tiene un ranking más alto que la etiqueta l_2 si $f(x, l_1) > f(x, l_2)$.

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
- Para un x dado la etiqueta l_1 tiene un ranking más alto que la etiqueta l_2 si $f(x, l_1) > f(x, l_2)$.
- Para un dato (x, Y) consideramos pares de etiquetas cruciales: $l_1, l_2 : l_1 \notin Y, l_2 \in Y$

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
- Para un x dado la etiqueta l_1 tiene un ranking más alto que la etiqueta l_2 si $f(x, l_1) > f(x, l_2)$.
- Para un dato (x, Y) consideramos pares de etiquetas cruciales: $l_1, l_2 : l_1 \notin Y, l_2 \in Y$
- f desordena (l_1, l_2) si $f(x, l_1) \geq f(x, l_2)$.

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
- Para un x dado la etiqueta l_1 tiene un ranking más alto que la etiqueta l_2 si $f(x, l_1) > f(x, l_2)$.
- Para un dato (x, Y) consideramos pares de etiquetas cruciales: $l_1, l_2 : l_1 \notin Y, l_2 \in Y$
- f desordena (l_1, l_2) si $f(x, l_1) \geq f(x, l_2)$.
- Meta:

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
- Para un x dado la etiqueta l_1 tiene un ranking más alto que la etiqueta l_2 si $f(x, l_1) > f(x, l_2)$.
- Para un dato (x, Y) consideramos pares de etiquetas cruciales: $l_1, l_2 : l_1 \notin Y, l_2 \in Y$
- f desordena (l_1, l_2) si $f(x, l_1) \geq f(x, l_2)$.
- Meta: Encontrar f con pocos pares cruciales desordenados.

Clasificación multiclase usando ranking

- Hipótesis asigna ranking a las etiquetas.
- Queremos que etiquetas correctas reciban ranking más alto.
- Hipótesis $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.
- Para un x dado la etiqueta l_1 tiene un ranking más alto que la etiqueta l_2 si $f(x, l_1) > f(x, l_2)$.
- Para un dato (x, Y) consideramos pares de etiquetas cruciales: $l_1, l_2 : l_1 \notin Y, l_2 \in Y$
- f desordena (l_1, l_2) si $f(x, l_1) \geq f(x, l_2)$.
- Meta: Encontrar f con pocos pares cruciales desordenados.

$$\text{rloss}_{\mathcal{D}}(f) = \mathbf{E}_{\mathcal{D}}\left[\frac{|\{(l_1, l_2) \in (\mathcal{Y} - Y) \times Y : f(x, l_1) \geq f(x, l_2)\}|}{|Y|\,|\mathcal{Y} - Y|}\right]$$

Algorithm 29 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

Algorithm 30 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

Algorithm 31 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Algorithm 32 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Escoja α_t

Algorithm 33 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Escoja α_t

Actualice D: $D_{t+1}(i, l_1, l_2) = \frac{D_t(i,l_1,l_2) \exp\left(\frac{1}{2}\alpha_t(h_t(x_i,l_1)-h_t(x_i,l_2))\right)}{Z_t}$

Algorithm 34 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Escoja α_t

Actualice D: $D_{t+1}(i, l_1, l_2) = \frac{D_t(i,l_1,l_2) \exp\left(\frac{1}{2}\alpha_t(h_t(x_i,l_1)-h_t(x_i,l_2))\right)}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

Algorithm 35 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Escoja α_t

Actualice D: $D_{t+1}(i, l_1, l_2) = \frac{D_t(i,l_1,l_2) \exp\left(\frac{1}{2}\alpha_t(h_t(x_i,l_1)-h_t(x_i,l_2))\right)}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

end for

Algorithm 36 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Escoja α_t

Actualice D: $D_{t+1}(i, l_1, l_2) = \frac{D_t(i,l_1,l_2) \exp\left(\frac{1}{2}\alpha_t(h_t(x_i,l_1)-h_t(x_i,l_2))\right)}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

end for

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

Algorithm 37 AdaBoost.MR

$$D_1(i, l_1, l_2) = \begin{cases} \frac{1}{n|Y_i||\mathcal{Y}-Y_i|} & \text{si } l_1 \notin Y_i, l_2 \in Y_i \\ 0 & \text{en otro caso} \end{cases}$$

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

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

Escoja α_t

Actualice D: $D_{t+1}(i, l_1, l_2) = \frac{D_t(i,l_1,l_2) \exp\left(\frac{1}{2}\alpha_t(h_t(x_i,l_1)-h_t(x_i,l_2))\right)}{Z_t}$

Donde Z_t normaliza D de manera que sea una distribución.

end for

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

Teorema

$$rloss_{\mathcal{D}}(f) \leq \prod_{t=1}^{T} Z_t$$