Boosting

Il Boosting nasce come risposta ad una domanda teorica all'interno del PAC Learning:

Dato un **Weak Learner** L: algoritmo di apprendimento L che soddisfa la PAC apprendibilità per valori fissati ϵ_0 e δ_0 (e non per tutti i valori di ϵ e δ)
è possibile usarlo come subroutine all'interno di un algoritmo L' che soddisfi la PAC apprendibilità per ogni valore di ϵ e δ ?

Boosting

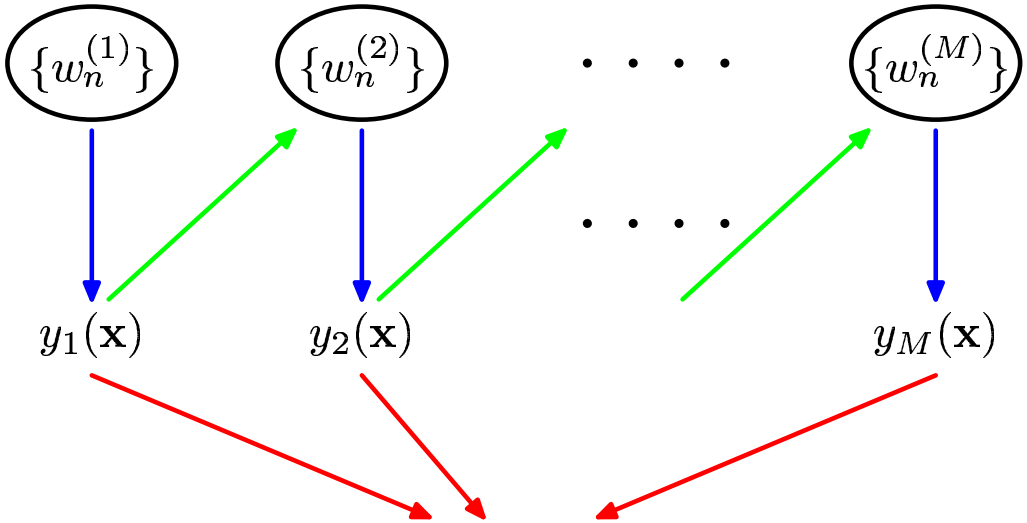
Il Boosting nasce come risposta ad una domanda teorica all'interno del PAC Learning:

Dato un **Weak Learner** L: algoritmo di apprendimento L che soddisfa la PAC apprendibilità per valori fissati ϵ_0 e δ_0 (e non per tutti i valori di ϵ e δ)
è possibile usarlo come subroutine all'interno di un algoritmo L' che soddisfi la PAC apprendibilità per ogni valore di ϵ e δ ?

La risposta (sorprendente) è SI e la dimostrazione è costruttiva!

$\exists$ varie versioni su come costruire L', noi vediamo una delle più popolari: AdaBoost

AdaBoost: lo schema generale di apprendimento



$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_m^M \alpha_m y_m(\mathbf{x})\right)$$

- usa distribuzione di probabilità sugli esempi
- apprende ipotesi usando la distribuzione
- modifi ca la distribuzione in base agli errori (+ peso)
- itera M volte e al termine combina le ipotesi generate

AdaBoost: parte 1

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = 1/N$ for $n = 1, \ldots, N$.
2. For $m = 1, \ldots, M$:
   (a) Fit a classifier $y_m(\mathbf{x})$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)$$

where $I(y_m(\mathbf{x}_n) \neq t_n)$ is the indicator function and equals 1 when $y_m(\mathbf{x}_n) \neq t_n$ and 0 otherwise.

AdaBoost: parte 2

(b) Evaluate the quantities

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

and then use these to evaluate

$$\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\}.$$

AdaBoost: parte 3

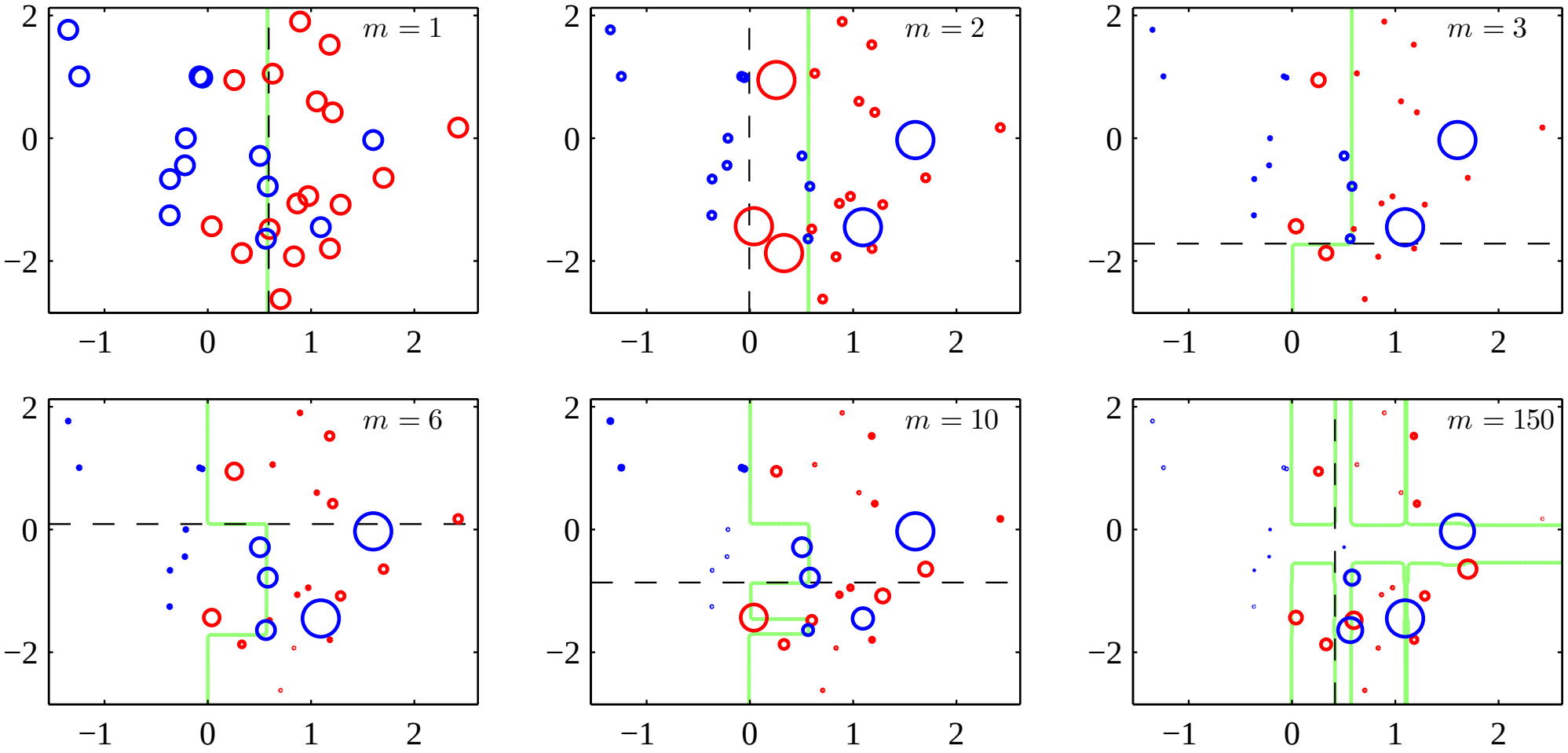
(c) Update the data weighting coefficients

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\}$$

3. Make predictions using the final model, which is given by

$$Y_M(\mathbf{x}) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(\mathbf{x})\right).$$

AdaBoost



Funzione minimizzata

Quale funzione obiettivo minimizza AdaBoost ?

Consideriamo la funzione

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(\mathbf{x}_n)\}$$

dove

$$f_m(\mathbf{x}) = \frac{1}{2} \sum_{l=1}^{m} \alpha_l y_l(\mathbf{x})$$

costituisce un classificatore che combina linearmente le ipotesi di base.

E deve essere minimizzata rispetto ai coefficienti α_l e i parametri delle ipotesi di base $y_1(\mathbf{x}), \ldots, y_m(\mathbf{x})$

Funzione minimizzata

Se però supponiamo che le ipotesi di base $y_1(\mathbf{x}), \ldots, y_{m-1}(\mathbf{x})$, e i rispettivi coefficienti $\alpha_1, \ldots, \alpha_{m-1}$, siano fissati, allora E deve essere minimizzato solo rispetto a $y_m(\mathbf{x})$ e α_m.

Possiamo quindi riscrivere E tenendo conto di questo fatto:

$$
\begin{aligned}
E &= \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\} \\
&= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n)\right\}
\end{aligned}
$$

dove $w_n^{(m)} = \exp\{-t_n f_{m-1}(\mathbf{x}_n)\}$ (che è costante rispetto a $y_m(\mathbf{x})$ e α_m)

Funzione minimizzata

Se con

- $\mathcal{T}_m$ indichiamo l'insieme delle istanze che sono classifi cate correttamente da $y_m(\mathbf{x})$
- $\mathcal{M}_m$ l'insieme delle istanze classifi cate erroneamente

possiamo riscrivere E come

$$
\begin{aligned}
E &= e^{-\alpha_m/2} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\alpha_m/2} \sum_{n \in \mathcal{M}_m} w_n^{(m)} \\
&= (e^{\alpha_m/2} - e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)}
\end{aligned}
$$

Funzione minimizzata

Si noti che minimizzare J_m di fatto permette di minimizzare E rispetto a $y_m(\mathbf{x})$.

Infatti

$$E = (e^{\alpha_m/2} - e^{-\alpha_m/2})J_m + \text{termine costante}$$

In modo analogo, usare $\alpha_m = ln\left\{\frac{1-\epsilon_m}{\epsilon_m}\right\}$ corrisponde a minimizzare E rispetto a α_m

Avendo trovato $y_m(\mathbf{x})$ e α_m ottimi, la distribuzione dei pesi diventa:

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{-\frac{1}{2}t_n\alpha_m y_m(\mathbf{x}_n)\right\}$$

e sfruttando il fatto $t_n y_m(\mathbf{x}) = 1 - 2I(y_m(\mathbf{x}) \neq t_n)$, può essere riscritta come

$$w_n^{(m+1)} = w_n^{(m)} \exp(-\alpha_m/2) \exp\left\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\right\}$$

che equivale all'aggiornamento di AdaBoost, in quanto $\exp(-\alpha_m/2)$ è indipendente da n