

Classificatori lineari

Introduciamo l'importante famiglia dei **classificatori lineari** per il caso $\mathcal{X} = \mathbb{R}^d$. Possiamo rappresentare geometricamente un classificatore binario $h : \mathbb{R}^d \rightarrow \{-1, +1\}$ con la partizione $\{S^+, S^-\}$ di \mathbb{R}^d tale che

$$h(\mathbf{x}) = \begin{cases} +1 & \text{se } \mathbf{x} \in S^+ \\ -1 & \text{se } \mathbf{x} \in S^- \end{cases}$$

I classificatori lineari sono quei classificatori h dove S^+ e S^- sono i semispazi definiti da un iperpiano S in \mathbb{R}^d . Algebricamente, un iperpiano è il luogo dei punti $\mathbf{x} \in \mathbb{R}^d$ che soddisfano l'equazione $v_1x_1 + \dots + v_dx_d = c$ dove v_1, \dots, v_n, c sono coefficienti reali. Se definiamo la notazione

$$\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^d u_i v_i$$

per il prodotto interno, possiamo riscrivere l'iperpiano come $S(\mathbf{v}, c) = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{v}^\top \mathbf{x} = c\}$.

Si ricordi che, detto θ l'angolo fra due vettori \mathbf{v} e \mathbf{x} , la quantità $\mathbf{v}^\top \mathbf{x} = \|\mathbf{v}\| \|\mathbf{x}\| \cos \theta$ è la lunghezza della proiezione di \mathbf{x} su \mathbf{v} moltiplicata per $\|\mathbf{v}\|$ o, equivalentemente, la lunghezza della proiezione di \mathbf{v} su \mathbf{x} moltiplicata per $\|\mathbf{x}\|$. Qui usiamo $\|\mathbf{x}\|$ per indicare la norma Euclidea di un vettore $\mathbf{x} = (x_1, \dots, x_d)$, ovvero

$$\|\mathbf{x}\| = \sqrt{\sum_{i=1}^d x_i^2}.$$

Dato un iperpiano $S = S(\mathbf{v}, c)$, e considerando $\mathbf{v}^\top \mathbf{x}$ come la lunghezza della proiezione di \mathbf{x} su \mathbf{v} moltiplicata per $\|\mathbf{v}\|$, si ha che il vettore \mathbf{v} è perpendicolare all'iperpiano S che lo taglia alla distanza $c/\|\mathbf{v}\|$ dall'origine.

I semispazi S^+ e S^- definiti da $S = \{\mathbf{x} : \mathbf{v}^\top \mathbf{x} = c\}$ sono

$$S^+ = \{\mathbf{x} : \mathbf{v}^\top \mathbf{x} > c\} \quad \text{e} \quad S^- = \{\mathbf{x}' : \mathbf{v}^\top \mathbf{x}' \leq c\}$$

ovvero l'insieme dei vettori \mathbf{x} tali che la proiezione su \mathbf{v} è almeno $c/\|\mathbf{v}\|$ e l'insieme dei vettori \mathbf{x}' tali che la proiezione su \mathbf{v} è minore di $c/\|\mathbf{v}\|$.

Possiamo rappresentare un classificatore lineare h associato all'iperpiano $S(\mathbf{v}, c)$ come $h(\mathbf{x}) = \text{sgn}(\mathbf{v}^\top \mathbf{x} - c)$, dove la funzione sgn è definita come

$$\text{sgn}(x) = \begin{cases} 1 & \text{se } x > 0, \\ -1 & \text{altrimenti.} \end{cases}$$

Iperpiani della forma $S(\mathbf{v}, 0) = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{v}^\top \mathbf{x} = 0\}$ passano per l'origine e vengono detti iperpiani omogenei. Un iperpiano non omogeneo $S(\mathbf{v}, c)$ in d dimensioni è equivalente all'iperpiano omogeneo

$S(\tilde{\mathbf{v}}, 0)$ in $d + 1$ dimensioni con $\tilde{\mathbf{v}} = (v_1, \dots, v_d, -c)$ quando i punti $\mathbf{x} \in \mathbb{R}^d$ vengono mappati nei punti $\tilde{\mathbf{x}} = (x_1, \dots, x_d, 1) \in \mathbb{R}^{d+1}$. Infatti, $\text{sgn}(\mathbf{v}^\top \mathbf{x} - c) = \text{sgn}(\tilde{\mathbf{v}}^\top \tilde{\mathbf{x}})$. Per questa ragione, parleremo senza perdita di generalità soltanto di algoritmi che generano classificatori lineari corrispondenti a iperpiani omogenei.

Cominciamo ora a occuparci del problema di apprendere classificatori lineari. Detta \mathcal{H}_d la famiglia dei classificatori lineari $h(\mathbf{x}) = \text{sgn}(\mathbf{w}^\top \mathbf{x})$ per $\mathbf{w} \in \mathbb{R}^d$, consideriamo l'algoritmo ERM che, dato un training set $(\mathbf{x}_1, y_1), (\mathbf{x}_m, y_m) \in \mathbb{R}^d \times \{-1, +1\}$ trova

$$\hat{h} = \underset{h \in \mathcal{H}_d}{\text{argmin}} \frac{1}{m} \sum_{t=1}^m \mathbb{I}\{h(\mathbf{x}_t) \neq y_t\}. \quad (1)$$

Putroppo, è improbabile trovare implementazioni efficienti di questo algoritmo. Infatti, il problema di decisione associato è NP-completo, anche quando $\mathbf{x}_t \in \{0, 1\}^d$ per $t = 1, \dots, m$. Più precisamente, definiamo il seguente problema di decisione.

MinDisagreement.

Istanza: Coppie $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m) \in \{0, 1\}^d \times \{-1, +1\}$. Intero k .

Domanda: Esiste un vettore $\mathbf{w} \in \mathbb{Q}^d$ tale che $y_t \mathbf{w}^\top \mathbf{x}_t \leq 0$ per al più k indici $t = 1, \dots, m$?

Vale il seguente risultato.

Teorema 1. *Il problema MinDisagreement è NP-completo.*

In aggiunta, è possibile dimostrare il risultato ancora più forte, relativo alla seguente versione di ottimizzazione del problema MinDisagreement.

MinDisOpt.

Istanza: Coppie $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m) \in \{0, 1\}^d \times \{-1, +1\}$.

Soluzione: Un vettore $\mathbf{w} \in \mathbb{Q}^d$ che minimizzi il numero di indici $t = 1, \dots, m$ tali che $y_t \mathbf{w}^\top \mathbf{x}_t \leq 0$.

Data un'istanza S (cioè un training set) di MinDisOpt, sia $\text{Opt}(S)$ il numero di esempi di S classificati in modo errato dall'iperpiano ottimo.

Teorema 2. *Se $P \neq NP$, per ogni $c > 0$ non esiste un algoritmo polinomiale che risolva ogni istanza S di MinDisOpt con un numero di esempi classificati in modo errato pari ad al più $c \text{Opt}(S)$.*

Questo significa che a meno che $P = NP$ (cosa ritenuta improbabile), non è possibile trovare un algoritmo che approssimi la soluzione di (1) a meno di un fattore costante in tempo polinomiale nella dimensione dell'input, ovvero polinomiale in m e d .

Il problema (1) diventa facile quando il training set è **linearmente separabile**. Dato un training set $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$, per ogni iperpiano \mathbf{u} definiamo il **margin**

$$\gamma(\mathbf{u}) \stackrel{\text{def}}{=} \min_{t=1, \dots, m} y_t \mathbf{u}^\top \mathbf{x}_t.$$

Un training set è linearmente separabile quando esiste un iperpiano separatore $\mathbf{u} \in \mathbb{R}^d$ tale che $\gamma(\mathbf{u}) > 0$. Si noti che $\gamma(\mathbf{u})/\|\mathbf{u}\|$ è la distanza dall'iperpiano separatore \mathbf{u} dell'elemento del training set ad esso più vicino. Dato che $\gamma(\mathbf{u})$ può essere moltiplicato per un qualunque fattore positivo riscalando \mathbf{u} , convenzionalmente assumiamo che un iperpiano separatore soddisfi sempre $\gamma(\mathbf{u}) \geq 1$.

Ora, in caso di training set linearmente separabile, il problema (1) è equivalente al seguente problema di programmazione lineare: trova un vettore $\mathbf{w} \in \mathbb{R}^d$ che soddisfi le disequazioni lineari

$$y_t \mathbf{w}^\top \mathbf{x}_t > 0 \quad t = 1, \dots, m .$$

Questo problema può quindi essere risolto in tempo polinomiale usando un qualsiasi algoritmo efficiente di programmazione lineare.

Vediamo ora un semplicissimo algoritmo di apprendimento per classificatori lineari che può essere usato per risolvere il problema nel caso separabile. L'algoritmo del Perceptrone, descritto nel box qui sotto, costruisce un classificatore lineare omogeneo esaminando gli elementi del training set in modo incrementale. L'esame del training set avviene aggiornando il modello lineare corrente (rappresentato da un iperpiano omogeneo con parametri \mathbf{w}) ogni volta che questa sbaglia a classificare il prossimo elemento (\mathbf{x}_t, y_t) del training set.

Algoritmo: Perceptrone

Input: Training set $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$.

Inizializza $\mathbf{w} = (0, \dots, 0)$.

Ripeti

 Leggi il prossimo esempio (\mathbf{x}_t, y_t) nel training set

 Se $y_t \mathbf{w}^\top \mathbf{x}_t \leq 0$, allora $\mathbf{w} \leftarrow \mathbf{w} + y_t \mathbf{x}_t$

finché $y_t \mathbf{w}^\top \mathbf{x}_t > 0$ per ogni $t = 1, \dots, m$

Output \mathbf{w}

Si noti che se l'algoritmo termina allora \mathbf{w} è un iperpiano separatore. L'aggiornamento $\mathbf{w} \leftarrow \mathbf{w} + y_t \mathbf{x}_t$ quando $y_t \mathbf{w}^\top \mathbf{x}_t \leq 0$ aumenta il valore di $y_t \mathbf{w}^\top \mathbf{x}_t$. Infatti,

$$y_t (\mathbf{w} + y_t \mathbf{x}_t)^\top \mathbf{x}_t = y_t \mathbf{w}^\top \mathbf{x}_t + \|\mathbf{x}_t\|^2 > y_t \mathbf{w}^\top \mathbf{x}_t$$

Geometricamente, l'algoritmo sposta \mathbf{w} verso \mathbf{x}_t se $y_t = 1$ e lo allontana da \mathbf{x}_t se $y_t = -1$.

Dato che l'iperpiano corrente \mathbf{w} è aggiornato al passo t soltanto quando $\text{sgn}(\mathbf{w}^\top \mathbf{x}_t) \neq y_t$, se l'algoritmo termina allora l'iperpiano finale dipende soltanto dal sottoinsieme del training set contenente tutti e soli gli esempi su cui è stato fatto almeno un aggiornamento. Quindi possiamo adattare a questo caso l'analisi svolta in "Compression Bounds" (dispensa 8 del corso). L'apparente difficoltà nel fare ciò deriva dal fatto che il Perceptrone compie in generale più epoche sul training set, e potenzialmente può effettuare più aggiornamenti sullo stesso esempio. Questo però non è un problema dato che, una volta che rimuoviamo dal training set gli esempi sui quali non è stato mai fatto un aggiornamento, possiamo far girare il Perceptrone sul nuovo training set e ottenere lo stesso iperpiano che avevamo ottenuto sul training set originale. Usando la disuguaglianza (4) in "Compression Bounds" otteniamo il seguente risultato.

Corollario 3. Sia S un training set di m esempi estratti in modo i.i.d. da un modello statistico (D, η) fissato ma ignoto. Supponiamo che l'algoritmo del Perceptrone con input S termini con output \mathbf{w} e sia $h_{\mathbf{w}}$ il classificatore lineare definito da \mathbf{w} . Se $M \leq \frac{m}{2}$ è la cardinalità del sottoinsieme di S degli esempi su cui è stato effettuato almeno un aggiornamento, allora

$$\text{er}(h_{\mathbf{w}}) \leq \tilde{\text{er}}(h_{\mathbf{w}}) + \sqrt{\frac{1}{m} \left((M+1) \ln m + \ln \frac{e}{\delta} \right)}$$

con probabilità almeno $1 - \delta$ rispetto all'estrazione del training set S , dove $\tilde{\text{er}}(h_{\mathbf{w}})$ denota la frazione di errori compiuti da $h_{\mathbf{w}}$ sui $m - M$ esempi in $S \setminus S'$.

Dimostriamo ora il teorema di convergenza del Perceptrone, ovvero che l'algoritmo termina sempre nel caso in cui il training set sia linearmente separabile.

Teorema 4 (Convergenza del Perceptrone). Per ogni training set $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$ linearmente separabile e per ogni iperpiano separatore \mathbf{u} con margine $\gamma(\mathbf{u}) \geq 1$, l'algoritmo del Perceptrone determina un iperpiano separatore \mathbf{w} (generalmente diverso da \mathbf{u}) dopo al più

$$M \leq \|\mathbf{u}\|^2 \left(\max_{t=1, \dots, m} \|\mathbf{x}_t\|^2 \right) \quad (2)$$

aggiornamenti.

Dimostrazione. Sia $\mathbf{w}_0 = (0, \dots, 0)$ l'iperpiano iniziale. Indichiamo con \mathbf{w}_M l'iperpiano dopo M aggiornamenti e indichiamo con $t_M \in \{1, \dots, m\}$ l'indice dell'esempio $(\mathbf{x}_{t_M}, y_{t_M})$ del training set che ha causato l' M -esimo aggiornamento, ovvero $\mathbf{w}_M = \mathbf{w}_{M-1} + y_{t_M} \mathbf{x}_{t_M}$. Ricaviamo un maggiorante di M derivando un maggiorante e un minorante a $\|\mathbf{w}_M\| \|\mathbf{u}\|$ come segue,

$$\|\mathbf{w}_M\|^2 = \|\mathbf{w}_{M-1} + y_{t_M} \mathbf{x}_{t_M}\|^2 = \|\mathbf{w}_{M-1}\|^2 + \|\mathbf{x}_{t_M}\|^2 + 2 y_{t_M} \mathbf{w}_{M-1}^\top \mathbf{x}_{t_M} \leq \|\mathbf{w}_{M-1}\|^2 + \|\mathbf{x}_{t_M}\|^2$$

in quanto $y_{t_M} \mathbf{w}_{M-1}^\top \mathbf{x}_{t_M} \leq 0$ dato che è stato eseguito l'aggiornamento $\mathbf{w}_M = \mathbf{w}_{M-1} + y_{t_M} \mathbf{x}_{t_M}$. Iterando questo ragionamento per M volte, e ricordando che $\mathbf{w}_0 = (0, \dots, 0)$, otteniamo

$$\|\mathbf{w}_M\|^2 \leq \|\mathbf{w}_0\|^2 + \sum_{i=1}^M \|\mathbf{x}_{t_i}\|^2 \leq M \left(\max_{t=1, \dots, m} \|\mathbf{x}_t\|^2 \right).$$

Quindi,

$$\|\mathbf{w}_M\| \|\mathbf{u}\| \leq \|\mathbf{u}\| \left(\max_{t=1, \dots, m} \|\mathbf{x}_t\| \right) \sqrt{M}.$$

Ora, per il minorante, consideriamo un qualunque iperpiano separatore \mathbf{u} e sia θ l'angolo fra \mathbf{u} e \mathbf{w}_M . Abbiamo

$$\begin{aligned} \|\mathbf{w}_M\| \|\mathbf{u}\| &\geq \|\mathbf{w}_M\| \|\mathbf{u}\| \cos(\theta) && \text{(dato che } -1 \leq \cos(\theta) \leq 1) \\ &= \mathbf{w}_M^\top \mathbf{u} && \text{(per definizione del prodotto interno } \mathbf{w}_T^\top \mathbf{u}) \\ &= (\mathbf{w}_{M-1} + y_{t_M} \mathbf{x}_{t_M})^\top \mathbf{u} \\ &= \mathbf{w}_{M-1}^\top \mathbf{u} + y_{t_M} \mathbf{u}^\top \mathbf{x}_{t_M} \\ &\geq \mathbf{w}_{M-1}^\top \mathbf{u} + 1 \end{aligned}$$

in quanto $1 \leq \gamma(\mathbf{u}) \leq y_t \mathbf{u}^\top \mathbf{x}_t$ per ogni $t = 1, \dots, m$. Iterando per M volte otteniamo

$$\|\mathbf{w}_M\| \|\mathbf{u}\| \geq \mathbf{w}_0^\top \mathbf{u} + M = M$$

dove abbiamo usato $\mathbf{w}_0^\top \mathbf{u} = 0$ dato che $\mathbf{w}_0 = (0, \dots, 0)$. Mettendo insieme maggiorante e minorante abbiamo

$$M \leq \|\mathbf{u}\| \left(\max_{t=1, \dots, M} \|\mathbf{x}_t\| \right) \sqrt{M} .$$

Risolvendo rispetto a M otteniamo (2). Dato che $\|\mathbf{u}\|$ è costante, il Perceptrone esegue un numero M di aggiornamenti maggiorato da una costante, e perciò converge ad un iperpiano separatore in un numero di aggiornamenti al più pari a tale costante. \square