(F94) Metodi statistici per l'apprendimento

Classificatori lineari

Docente: *Nicolò Cesa-Bianchi* versione 11 aprile 2016

Introduciamo l'importante famiglia dei **classificatori lineari**. Consideriamo il caso $\mathcal{X} = \mathbb{R}^d$. Indicheremo con $\|\boldsymbol{x}\|$ la norma Euclidea di un vettore $\boldsymbol{x} = (x_1, \dots, x_d)$, ovvero

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

Come noto, possiamo rappresentare geometricamente un classificatore binario $h : \mathbb{R}^d \to \{-1, +1\}$ con la partizione $\{\mathcal{X}_{-1}, \mathcal{X}_{+1}\}$ di $\mathbb{R}^d$ tale che

$$\mathcal{X}_y = \Big\{\boldsymbol{x} \in \mathbb{R}^d \ : \ h(\boldsymbol{x}) = y\Big\} \qquad \text{per } y \in \{-1, +1\} \ .$$

Consideriamo lo spazio dei modelli consistente nell'insieme dei classificatori h la cui partizione di $\mathbb{R}^d$ è determinata dai semispazi generati da tutti i possibili iperpiani. Un iperpiano S in $\mathbb{R}^d$ è il luogo dei punti $\boldsymbol{x} \in \mathbb{R}^d$ che soddisfano l'equazione $v_1 x_1 + \ldots + v_d x_d = c$ per un dato insieme $v_1, \dots, v_n, c$ di coefficienti reali.

Definiamo la notazione $\boldsymbol{u}^\top \boldsymbol{v} = \sum_{i=1}^{d} u_i v_i$ per il prodotto interno. Possiamo quindi riscrivere l'iperpiano $S = S(\boldsymbol{v}, c)$ come $S(\boldsymbol{v}, c) = \big\{\boldsymbol{x} \in \mathbb{R}^d \ : \ \boldsymbol{v}^\top \boldsymbol{x} = c\big\}$.

Si ricordi che, detto θ l'angolo fra due vettori $\boldsymbol{v}$ e $\boldsymbol{x}$, la quantità $\boldsymbol{v}^\top \boldsymbol{x} = \|\boldsymbol{v}\| \, \|\boldsymbol{x}\| \cos\theta$ è la lunghezza della proiezione di $\boldsymbol{x}$ su $\boldsymbol{v}$ moltiplicata per $\|\boldsymbol{v}\|$ o, equivalentemente, la lunghezza della proiezione di $\boldsymbol{v}$ su $\boldsymbol{x}$ moltiplicata per $\|\boldsymbol{x}\|$.

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

L'iperpiano $S = \big\{\boldsymbol{x} \ : \ \boldsymbol{v}^\top \boldsymbol{x} = c\big\}$ definisce due semispazi

$$S^+(\boldsymbol{v}, c) = \Big\{\boldsymbol{x} \ : \ \boldsymbol{v}^\top \boldsymbol{x} \geq c\Big\} \qquad \text{e} \qquad S^-(\boldsymbol{v}, c) = \Big\{\boldsymbol{x}' \ : \ \boldsymbol{v}^\top \boldsymbol{x}' < c\Big\}$$

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

Un classificatore lineare con coefficienti $(\boldsymbol{v}, c)$ corrisponde alla funzione $h : \mathbb{R}^d \to \{-1, +1\}$ tale che

$$h(\boldsymbol{x}) = \begin{cases} 1 & \text{se } \boldsymbol{x} \in S^+(\boldsymbol{v}, c), \\ -1 & \text{se } \boldsymbol{x} \in S^-(\boldsymbol{v}, c). \end{cases}$$

Equivalentemente, $h(\boldsymbol{x}) = \text{sgn}(\boldsymbol{v}^\top \boldsymbol{x} - c)$, dove la funzione sgn è definita come

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

Iperpiani della forma $S(\boldsymbol{v}, 0) = \left\{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{v}^\top \boldsymbol{x} = 0\right\}$ passano per l'origine e vengono detti iperpiani omogenei. Un iperpiano non omogeneo $S(\boldsymbol{v}, c)$ in d dimensioni è equivalente all'iperpiano omogeneo $S(\bar{\boldsymbol{v}}, 0)$ in $d+1$ dimensioni con $\bar{\boldsymbol{v}} = (v_1, \ldots, v_d, -c)$ quando i punti $\boldsymbol{x} \in \mathbb{R}^d$ vengono mappati nei punti $\boldsymbol{x}' = (x_1, \ldots, x_d, 1) \in \mathbb{R}^{d+1}$. Infatti, $\text{sgn}(\boldsymbol{v}^\top \boldsymbol{x} - c) = \text{sgn}(\bar{\boldsymbol{v}}^\top \boldsymbol{x}')$. 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 $h(\boldsymbol{x}) = \text{sgn}\big(\boldsymbol{w}^\top \boldsymbol{x}\big)$ per $\boldsymbol{w} \in \mathbb{R}^d$, consideriamo l'algoritmo ERM che, dato un training set $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_m, y_m) \in \mathbb{R}^d \times \{-1, +1\}$ trova

$$\widehat{h} = \operatorname*{argmin}_{h \in \mathcal{H}_d} \frac{1}{m} \sum_{t=1}^{m} \mathbb{I}_{\{h(\boldsymbol{x}_t) \neq y_t\}} \ . \tag{1}$$

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

MinDisagreement.
Istanza: Coppie $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m) \in \{0,1\}^d \times \{-1, +1\}$. Intero k.
Domanda: Esiste un vettore $\boldsymbol{w} \in \mathbb{Q}^d$ tale che $y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$ per al più k indici $t = 1, \ldots, m$?

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 ottimimizzazione del problema MinDisagreement.

MinDisOpt.
Istanza: Coppie $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m) \in \{0,1\}^d \times \{-1, +1\}$.
Soluzione: Un vettore $\boldsymbol{w} \in \mathbb{Q}^d$ che minimizzi il numero di indici $t = 1, \ldots, m$ tali che $y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$.

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 $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$, per ogni iperpiano $\boldsymbol{u}$ definiamo il **margine**

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

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

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

$$y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t > 0 \qquad t = 1, \ldots, m \ .$$

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 $\boldsymbol{w}$) ogni volta che questa sbaglia a classificare il prossimo elemento $(\boldsymbol{x}_t, y_t)$ del training set.

Algoritmo: **Perceptrone**
Parametri: N (intero positivo).
Input: Training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$.
Inizializza $\boldsymbol{w} = (0, \ldots, 0)$.
Ripeti N volte:

    Per $t = 1, \ldots, m$

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

Output $\boldsymbol{w}$.

Quando $y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$, ovvero $\boldsymbol{w}$ sbaglia a classificare $(\boldsymbol{x}_t, y_t)$, l'algoritmo sposta $\boldsymbol{w}$ verso $\boldsymbol{x}_t$ se $y_t = 1$ e lo allontana da $\boldsymbol{x}_t$ se $y_t = -1$. La modifica di $\boldsymbol{w}$ ogniqualvolta $y_t \, \boldsymbol{w}^\top \boldsymbol{x}_t \leq 0$ viene chiamata aggiornamento.

Si noti che il Perceptrone aggiorna l'iperpiano corrente $\boldsymbol{w}$ al passo t soltanto quando $\operatorname{sgn}\big(\boldsymbol{w}^\top \boldsymbol{x}_t\big) \neq y_t$. Ciò significa che 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" (fascicolo 8 delle note al corso). L'apparente difficoltà nel fare ciò deriva dal fatto che il Perceptrone compie in generale N 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 (con lo stesso numero N di epoche come input) 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 di eseguire l'algoritmo del Perceptrone su S con parametro N arbitrario e chiamiamo $\boldsymbol{w}$ l'iperpiano ottenuto e $h_{\boldsymbol{w}}$ il corrispondente classificatore lineare. Allora, se $M \leq \frac{m}{2}$ è la cardinalità del sottoinsieme di S degli esempi su cui è stato effettuato almeno un aggiornamento, allora*

$$\mathrm{er}(h_{\boldsymbol{w}}) \leq \widetilde{\mathrm{er}}(h_{\boldsymbol{w}}) + \sqrt{\frac{1}{m}\left(\big(M+1\big)\ln m + \ln\frac{e}{\delta}\right)}$$

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

Dimostriamo ora il teorema di convergenza del Perceptrone, ovvero la convergenza dell'algoritmo ad un iperpiano separatore nel caso in cui il training set sia linearmente separabile.

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

$$M \leq \|\boldsymbol{u}\|^2 \left(\max_{t=1,\ldots,m} \|\boldsymbol{x}_t\|\right)^2 \tag{2}$$

aggiornamenti.

Dimostrazione. Sia $\boldsymbol{w}_1 = (0, \ldots, 0)$ l'iperpiano iniziale. Indichiamo con $\boldsymbol{w}_T$ l'iperpiano dopo $T-1$ passi dell'algoritmo per un qualunque $T > 1$. Sia M il numero di aggiornamenti eseguiti nei $T-1$ passi e siano $1 \leq t_1 < \ldots < t_M < T$ i passi in cui questi aggiornamenti sono stati effettuati. Ricaviamo un maggiorante di M derivando un maggiorante e un minorante a $\|\boldsymbol{w}_T\|\,\|\boldsymbol{u}\|$ come segue. Dato che $\boldsymbol{w}_T$ è stato aggiornato l'ultima volta al passo t_M, abbiamo $\boldsymbol{w}_T = \boldsymbol{w}_{t_M} + y_{t_M}\boldsymbol{x}_{t_M}$. E quindi

$$\begin{aligned}
\|\boldsymbol{w}_T\|^2 &= \|\boldsymbol{w}_{t_M} + y_{t_M}\boldsymbol{x}_{t_M}\|^2 \\
&= \|\boldsymbol{w}_{t_M}\|^2 + \|\boldsymbol{x}_{t_M}\|^2 + 2\, y_{t_M}\boldsymbol{w}_{t_M}^\top \boldsymbol{x}_{t_M} \\
&\leq \|\boldsymbol{w}_{t_M}\|^2 + \|\boldsymbol{x}_{t_M}\|^2
\end{aligned}$$

in quanto $y_{t_M}\boldsymbol{w}_{t_M}^\top \boldsymbol{x}_{t_M} < 0$ dall'ipotesi che al passo t_M sia avvenuto un aggiornamento. Iterando questo ragionamento per M volte otteniamo

$$\|\boldsymbol{w}_T\|^2 \leq \|\boldsymbol{w}_1\|^2 + \sum_{i=1}^{M} \|\boldsymbol{x}_{t_i}\|^2 \leq M \max_{t=1,\ldots,m} \|\boldsymbol{x}_t\|^2 \ .$$

Quindi,

$$\|\boldsymbol{w}_T\| \, \|\boldsymbol{u}\| \leq \|\boldsymbol{u}\| \max_{t=1,\ldots,m} \|\boldsymbol{x}_t\| \sqrt{M} \ .$$

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

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

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

$$\|\boldsymbol{w}_T\| \, \|\boldsymbol{u}\| \geq \boldsymbol{w}_1^\top \boldsymbol{u} + M = M$$

dove abbiamo usato $\boldsymbol{w}_1^\top \boldsymbol{u} = 0$ dato che $\boldsymbol{w}_1 = (0, \ldots, 0)$. Mettendo insieme maggiorante e minorante abbiamo

$$M \leq \|\boldsymbol{u}\| \max_{1 \leq t \leq m} \|\boldsymbol{x}_t\| \sqrt{M} \ .$$

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