(F94) Metodi statistici per l'apprendimento

Funzioni kernel

Docente: *Nicolò Cesa-Bianchi* versione 4 maggio 2017

I predittori lineari, che usano la mappa $\boldsymbol{x} \mapsto \boldsymbol{w}^\top \boldsymbol{x}$, possono soffrire di un errore di bias molto elevato. Ovvero, la distribuzione $\mathcal{D}$ che genera i dati può essere tale che il rischio statistico $\ell_{\mathcal{D}}(h)$ è alto per ogni elemento h in una classe $\mathcal{H}$ di predittori lineari. In questa situazione anche il miglior predittore in $\mathcal{H}$ è destinato ad avere un test error tipicamente alto. D'altra parte, i predittori lineari sono semplici da implementare e da analizzare.

Utilizzando una mappa non lineare per trasformare le istanze di un problema di apprendimento, è però possibile utilizzare predittori lineari per rappresentare predittori non lineari. Consideriamo infatti una funzione nonlineare $\boldsymbol{\phi} : \mathbb{R}^d \to \mathcal{V}$ che mappa le istanze $\boldsymbol{x} \in \mathbb{R}^d$ in un nuovo spazio $\mathcal{V}$ con un numero molto maggiore (anche infinito) di dimensioni. Costruendo la funzione $\boldsymbol{\phi}$ opportunamente, abbiamo che superfici nonlineari in $\mathbb{R}^d$ vengono mappate in superfici lineari (iperpiani) in $\mathcal{V}$.

Per esempio, consideriamo la funzione $\boldsymbol{\phi} : \mathbb{R}^2 \to \mathbb{R}^6$, tale che $\boldsymbol{\phi}(x_1, x_2) = \big(1, x_1^2, x_2^2, x_1, x_2, x_1 x_2\big)$. Un iperpiano omogeneo in $\mathbb{R}^6$ con coefficienti $\boldsymbol{w} = (w_1, \dots, w_6)$ ha forma

$$\Big\{ \boldsymbol{z} \in \mathbb{R}^6 \,:\, \boldsymbol{w}^\top \boldsymbol{z} = 0 \Big\} \ .$$

Questo iperpiano può essere utilizzato per descrivere una superficie nonlineare in $\mathbb{R}^2$. Infatti, l'insieme dei $\boldsymbol{x} \in \mathbb{R}^2$ tali che $\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) = 0$ per definizione di $\boldsymbol{\phi}$ soddisfano l'equazione

$$w_1 + w_2 x_1^2 + w_3 x_2^2 + w_4 x_1 + w_5 x_2 + w_6 x_1 x_2 = 0$$

che è l'equazione di una conica nello spazio Euclideo bidimensionale (ovvero la famiglia a cui appartengono ellissi, parabole e iperboli).

Se ora consideriamo una generica mappa $\boldsymbol{\phi} : \mathbb{R}^d \to \mathbb{R}^N$, con $N \gg d$, un classificatore lineare in $\mathbb{R}^N$ per le istanze mappate da $\mathbb{R}^d$ avrà forma

$$h\big(\boldsymbol{\phi}(\boldsymbol{x})\big) = \operatorname{sgn}\big(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})\big) \qquad \text{dove} \quad \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) = \sum_{i=1}^{N} w_i \, \phi(\boldsymbol{x})_i \ .$$

Se $N \gg d$ il calcolo di $\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})$, che richiede la somma di N prodotti, può però risultare computazionalmente eccessivo.

In realtà, questo problema computazionale può essere enormemente alleviato combinando certi tipi di mappe col modo con cui gli algoritmi di apprendimento lineare tipicamente rapprensentano i loro modelli. Infatti, ricordando per esempio la regola di aggiornamento del Perceptrone $\boldsymbol{w} \leftarrow \boldsymbol{w} + y\boldsymbol{x}$, possiamo scrivere un generico classificatore lineare h generato dal Perceptrone come

$$h(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{j=1}^{M} y_{t_j} \boldsymbol{x}_{t_j}^\top \boldsymbol{x} \right)$$

dove $(\boldsymbol{x}_{t_1}, y_{t_1}), \ldots, (\boldsymbol{x}_{t_M}, y_{t_M})$ sono gli esempi sui quali il Perceptrone ha effettuato un aggiornamento. Se quindi facciamo girare il Perceptrone in $\mathcal{V}$, il classificatore lineare h diventa

$$\widetilde{h}(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{j=1}^{M} y_{t_j} \boldsymbol{\phi}(\boldsymbol{x}_{t_j})^\top \boldsymbol{\phi}(\boldsymbol{x})\right) .$$

Per calcolare $\widetilde{h}(\boldsymbol{x})$ efficientemente dobbiamo poter calcolare ciascun termine $\boldsymbol{\phi}(\boldsymbol{x}_{t_j})^\top \boldsymbol{\phi}(\boldsymbol{x})$ efficientemente. La tecnica delle funzioni kernel ci dice come scegliere $\boldsymbol{\phi}$ in modo tale che esista una funzione $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ tale che

$$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \qquad \text{per ogni } \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d. \tag{1}$$

Per esempio, la funzione kernel corrispondente a $\boldsymbol{\phi}(x_1, x_2) = \left(1, x_1^2, x_2^2, \sqrt{2}\, x_1, \sqrt{2}\, x_2, \sqrt{2}\, x_1 x_2\right)$ è $K(\boldsymbol{x}, \boldsymbol{x}') = \left(1 + \boldsymbol{x}^\top \boldsymbol{x}'\right)^2$, come è immediato verificare.

Fissata una funzione kernel K, possiamo allora scrivere il classificatore lineare a soglia generato dal Perceptrone come

$$h_K(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_{j=1}^{M} y_{t_j} K(\boldsymbol{x}_{t_j}, \boldsymbol{x})\right) . \tag{2}$$

Si veda qui sotto per una descrizione del Perceptrone con kernel.

Algoritmo: **Perceptrone con kernel**
Inizializza S con l'insieme vuoto
Per ogni $t = 1, 2, \ldots$

1. Leggi il prossimo esempio $(\boldsymbol{x}_t, y_t)$
2. Calcola $\hat{y}_t = \operatorname{sgn}\left(\sum_{s \in S} y_s\, K(\boldsymbol{x}_s, \boldsymbol{x}_t)\right)$
3. Se $\hat{y}_t \neq y_t$ aggiungi t a S

La generalizzazione ovvia del kernel per apprendere coniche in $\mathbb{R}^2$ è il kernel polinomiale $K_n(\boldsymbol{x}, \boldsymbol{x}') = \left(1 + \boldsymbol{x}^\top \boldsymbol{x}'\right)^n$ con $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$. Questo kernel permette di apprendere classificatori di grado n in $\mathbb{R}^d$ rappresentandoli come iperpiani in uno spazio $\mathbb{R}^N$. Possiamo esplicitare la funzione $\boldsymbol{\phi}_n$ tale che $K_n(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}_n(\boldsymbol{x})^\top \boldsymbol{\phi}_n(\boldsymbol{x}')$ usando il teorema del binomio di Newton,

$$\left(1 + \boldsymbol{x}^\top \boldsymbol{x}'\right)^n = \sum_{k=0}^{n} \binom{n}{k} \left(\boldsymbol{x}^\top \boldsymbol{x}'\right)^k . \tag{3}$$

Osserviamo poi che vale

$$\left(\boldsymbol{x}^\top \boldsymbol{x}'\right)^k = \left(\sum_{i=1}^{d} x_i x_i'\right)^k = \sum_{\boldsymbol{v} \in \{1,\ldots,d\}^k} \left(\prod_{s=1}^{k} x_{v_s} x'_{v_s}\right) .$$

Quindi,

$$\boldsymbol{\phi}_n(\boldsymbol{x}) = \left(\sqrt{\binom{n}{k}} \prod_{s=1}^{k} x_{v_s} \right)_{k=0,\ldots,n,\ \boldsymbol{v}\in\{1,\ldots,d\}^k} .$$

In altre parole, il mapping $\boldsymbol{\phi}_n : \mathbb{R}^d \to \mathbb{R}^N$ associato al kernel polinomiale K_n è un vettore le cui componenti sono indicizzate da tutti i monomi $\prod_{s=1}^{k} x_{v_s}$ di grado al più n pesati dalle radici dei coefficienti binomiali.

Un ulteriore tipo di kernel è il kernel Gaussiano,

$$K_\gamma(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{1}{2\gamma} \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|^2\right) \qquad \gamma > 0 .$$

Per trovare la mappa $\boldsymbol{\phi}_\gamma$ associata a K_γ procediamo come segue,

$$\begin{aligned}
\exp\left(-\frac{1}{2\gamma} \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|^2\right) &= \exp\left(-\frac{1}{2\gamma}\left(\|\boldsymbol{x}\|^2 + \left\|\boldsymbol{x}'\right\|^2\right)\right) \exp\left(\frac{1}{\gamma}\left(\boldsymbol{x}^\top \boldsymbol{x}'\right)\right) \\
&= \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2\gamma}\right) \exp\left(-\frac{\|\boldsymbol{x}'\|^2}{2\gamma}\right) \sum_{n=0}^{\infty} \frac{1}{n!} \frac{\left(\boldsymbol{x}^\top \boldsymbol{x}'\right)^n}{\gamma^n}
\end{aligned}$$

dove abbiamo usato l'espansione in serie di Taylor $e^x = 1 + x + \frac{x^2}{2!} + \cdots$. Osservando la formula (3) ci rendiamo conto che il kernel Gaussiano è il limite di una successione di somme parziali in cui gli elementi della successione sommata sono gli elementi di un kernel polinomiale pesati dal reciproco del fattoriale. Il parametro γ svolge il ruolo di fattore di scala per i prodotti $\boldsymbol{x}^\top \boldsymbol{x}'$. Infine i fattori $e^{-\|\boldsymbol{x}\|^2/(2\gamma)} e^{-\|\boldsymbol{x}'\|^2/(2\gamma)}$ normalizzano rispetto a $\boldsymbol{x}$ e $\boldsymbol{x}'$. Infatti, $K_\gamma(\boldsymbol{x}, \boldsymbol{x}) = 1$ per ogni $\boldsymbol{x} \in \mathbb{R}^d$.

Il kernel Gaussiano è *universale*, nel senso che un classificatore della forma (2), cioè tale che $h(\boldsymbol{x}) = \operatorname{sgn}\big(g(\boldsymbol{x})\big)$, con $g : \mathbb{R}^d \to \mathbb{R}$ appartenente alla classe

$$\mathcal{H}_\gamma = \left\{ \sum_{i=1}^{N} \alpha_i \, K_\gamma(\boldsymbol{x}_i, \cdot) \,:\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d,\, \alpha_1, \ldots, \alpha_N \in \mathbb{R},\, N \in \mathbb{N} \right\} \tag{4}$$

può approssimare con precisione arbitraria una qualunque funzione continua da $\mathbb{R}^d$ a $\mathbb{R}$ per ogni valore di $\gamma > 0$. Questo significa che se (D, η) è un modello statistico con η continuo, allora possiamo approssimare in modo arbitrariamente accurato il classificatore Bayesiano ottimo $f^*(x) = \operatorname{sgn}\big(2\eta(x) - 1\big)$ usando funzioni della forma $\operatorname{sgn}(g)$ con $g \in \mathcal{H}_\gamma$.

Dato un qualunque insieme $\mathcal{X}$ a cui appartengono le istanze $\boldsymbol{x}$ (non necessariamente $\mathbb{R}^d$), è possibile dimostrare che una qualunque funzione simmetrica $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ è un kernel se e solo se per ogni $m \in \mathbb{N}$ e per ogni $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathcal{X}$, la matrice $\boldsymbol{K}$ di dimensioni $m \times m$ e tale che $\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ è positiva semidefinita. Ovvero, $\boldsymbol{z}^\top \boldsymbol{K} \boldsymbol{z} \geq 0$ per ogni $\boldsymbol{z} \in \mathbb{R}^m$. La rappresentazione della mappa $\phi_K : \mathcal{X} \to \mathcal{H}_K$ associata ad un kernel K e la rappresentazione dello spazio $\mathcal{H}_K$ non sono uniche. Però possiamo sempre rappresentare ϕ_K come $\phi_K(\boldsymbol{x}) = K(\boldsymbol{x}, \cdot)$ e $\mathcal{H}_K$ come lo spazio

$$\mathcal{H}_K = \left\{ \sum_{i=1}^{N} \alpha_i \, K(\boldsymbol{x}_i, \cdot) \,:\, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^d,\, \alpha_1, \ldots, \alpha_N \in \mathbb{R},\, N \in \mathbb{N} \right\}$$

equivalente a quello definito in (4) per i kernel Gaussiani.

$\mathcal{H}_K$ è uno spazio lineare di funzioni dotato di un'operazione di prodotto interno (l'analogo del prodotto scalare fra vettori) che soddisfa

$$\big\langle K(\boldsymbol{x},\cdot), K(\boldsymbol{x}',\cdot)\big\rangle = K(\boldsymbol{x},\boldsymbol{x}')$$

in modo che $\big\langle \phi_K(\boldsymbol{x}), \phi_K(\boldsymbol{x}')\big\rangle = K(\boldsymbol{x},\boldsymbol{x}')$ —ovvero, il kernel implementa il prodotto interno su ϕ_K. In generale, se

$$f = \sum_{i=1}^{N} \alpha_i\, K(\boldsymbol{x}_i,\cdot) \qquad \text{e} \qquad g = \sum_{j=1}^{M} \alpha_j\, K(\boldsymbol{x}'_j,\cdot)$$

allora

$$\begin{aligned}
\big\langle f, g\big\rangle &= \left\langle \sum_{i=1}^{N} \alpha_i\, K(\boldsymbol{x}_i,\cdot), \sum_{j=1}^{M} \beta_j\, K(\boldsymbol{x}'_j,\cdot) \right\rangle \\
&= \sum_{i=1}^{N} \alpha_i \left\langle K(\boldsymbol{x}_i,\cdot), \sum_{j=1}^{M} \beta_j\, K(\boldsymbol{x}'_j,\cdot) \right\rangle \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i\beta_j \big\langle K(\boldsymbol{x}_i,\cdot), K(\boldsymbol{x}'_j,\cdot)\big\rangle \\
&= \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i\beta_j K(\boldsymbol{x}_i,\boldsymbol{x}'_j)
\end{aligned}$$

dove abbiamo usato le proprietà di linearità che valgono per qualsiasi prodotto interno.

Dato che i predittori con kernel sono predittori lineari in $\mathcal{H}_K$, è possibile limitare il rischio (statistico o sequenziale) usando le tecniche utilizzate in precedenza per i predittori lineari. Il maggiorante sul rischio dipenderà dai parametri usuali, ovvero $\|\boldsymbol{w}\|$ e $\max_t \|\boldsymbol{x}_t\|$, dove però le norme sono calcolate in $\mathcal{H}_K$. In analogia con la definizione di norma di vettori $\|\boldsymbol{x}\|^2 = \boldsymbol{x}^\top\boldsymbol{x}$, la norma di $f \in \mathcal{H}_K$ è definita come $\|f\|^2 = \langle f, f\rangle$.

Questo ci dà subito il seguente risultato: $\|\phi_K(\boldsymbol{x})\| = \sqrt{\big\langle K(\boldsymbol{x},\cdot), K(\boldsymbol{x},\cdot)\big\rangle} = \sqrt{K(\boldsymbol{x},\boldsymbol{x})}$.

Nel caso di un kernel K, per ogni $g \in \mathcal{H}_K$ abbiamo che

$$\|g\| = \left\| \sum_{i=1}^{N} \alpha_i\, K(\boldsymbol{x}_i,\cdot) \right\| = \sqrt{\left\langle \sum_{i=1}^{N} \alpha_i\, K(\boldsymbol{x}_i,\cdot), \sum_{j=1}^{N} \alpha_j\, K(\boldsymbol{x}_j,\cdot) \right\rangle} = \sqrt{\sum_{i,j=1}^{N} \alpha_i\alpha_j\, K(\boldsymbol{x}_i,\boldsymbol{x}_j)}\ .$$

Possiamo anche calcolare il prodotto interno fra $\phi_K(\boldsymbol{x}) = K(\boldsymbol{x},\cdot)$ e un qualunque $f \in \mathcal{H}_K$,

$$\big\langle f, K(\boldsymbol{x},\cdot)\big\rangle = \sum_{i=1}^{N} \alpha_i \big\langle K(\boldsymbol{x}_i,\cdot), K(\boldsymbol{x},\cdot)\big\rangle = \sum_{i=1}^{N} \alpha_i\, K(\boldsymbol{x}_i,\boldsymbol{x}) = f(\boldsymbol{x})\ .$$

Da ciò si vede che f calcola il prodotto interno fra se stessa e ϕ_K. Questa proprietà si chiama *reproducing* e per questa ragione $\mathcal{H}_K$ è anche chiamato un *reproducing kernel Hilbert space* (RKHS).

Consideriamo un training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$ linearmente separabile in $\mathcal{H}_K$ per un qualche kernel K. Ovvero, esiste un $f \in \mathcal{H}_K$ tale che $\min_t y_t f(\boldsymbol{x}_t) \geq 1$. Il teorema di convergenza del Perceptrone (variante kernel) ci dà un maggiorante sul numero di aggiornamenti pari a

$$\|f\|^2 \max_t K(\boldsymbol{x}_t, \boldsymbol{x}_t) \ .$$