Introduzione al Machine Learning: Regressione logistica

Vincenzo Bonifaci

Regressione logistica

La *regressione logistica* è un metodo (di tipo ERM) per la stima di probabilità

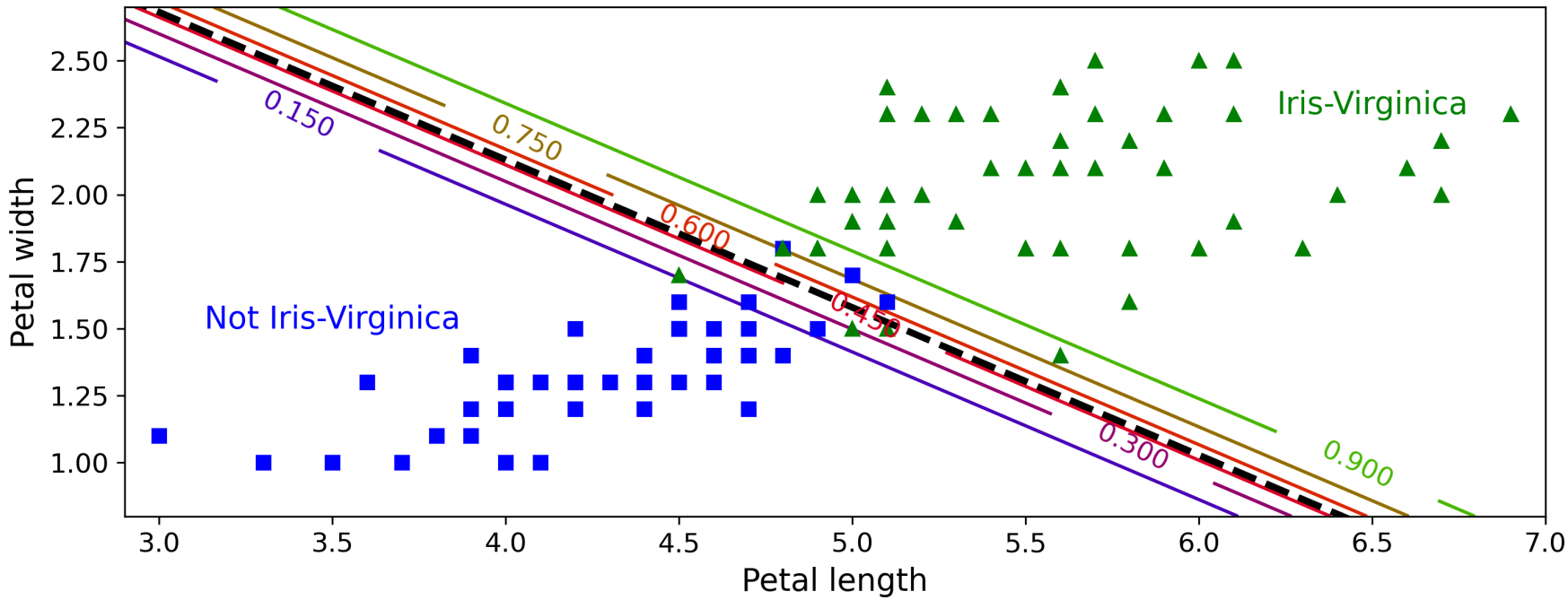
Dato input x, cerca di stimare la probabilità condizionata che l'etichetta y abbia un certo valore piuttosto che un altro ($\Pr(y|x)$)

La regressione logistica può essere usata anche per la classificazione: dato x, restituisce il valore di y più probabile

Stima di probabilità per etichette binarie

Stima della probabilità condizionata (etichette binarie)

Dato: un insieme di esempi (x, y) con $x \in \mathbb{R}^{d+1}$ e $y \in \{0, 1\}$
Trova: una funzione $h : \mathcal{X} \to [0, 1]$ con $h(x) \approx \Pr(y = 1|x)$



Un modello lineare per la stima di probabilità?

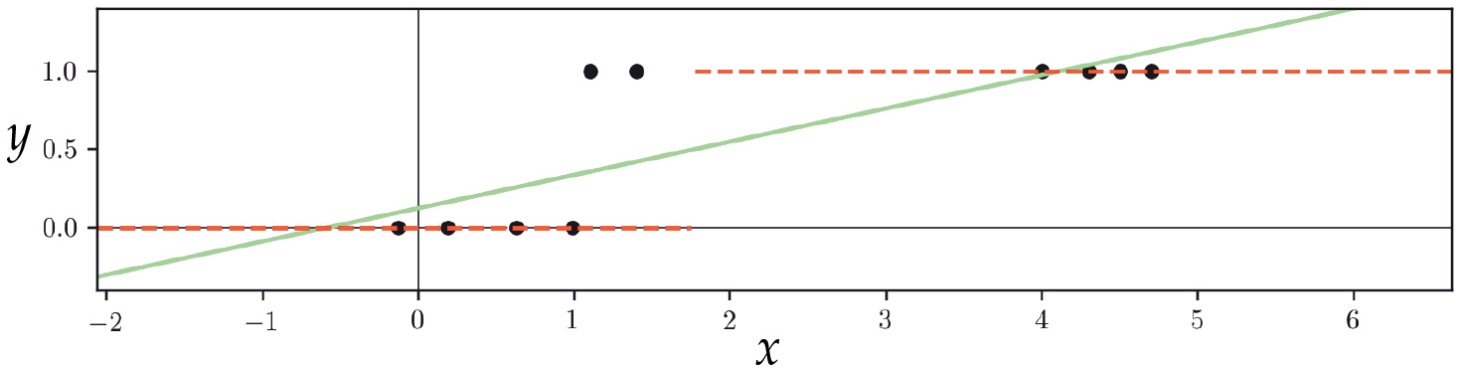
Dato x, vogliamo stimare $\Pr(y = 1|x)$ attraverso una funzione lineare

$$w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d = w^\top x$$

Vorremmo che $\Pr(y = 1|x)$:

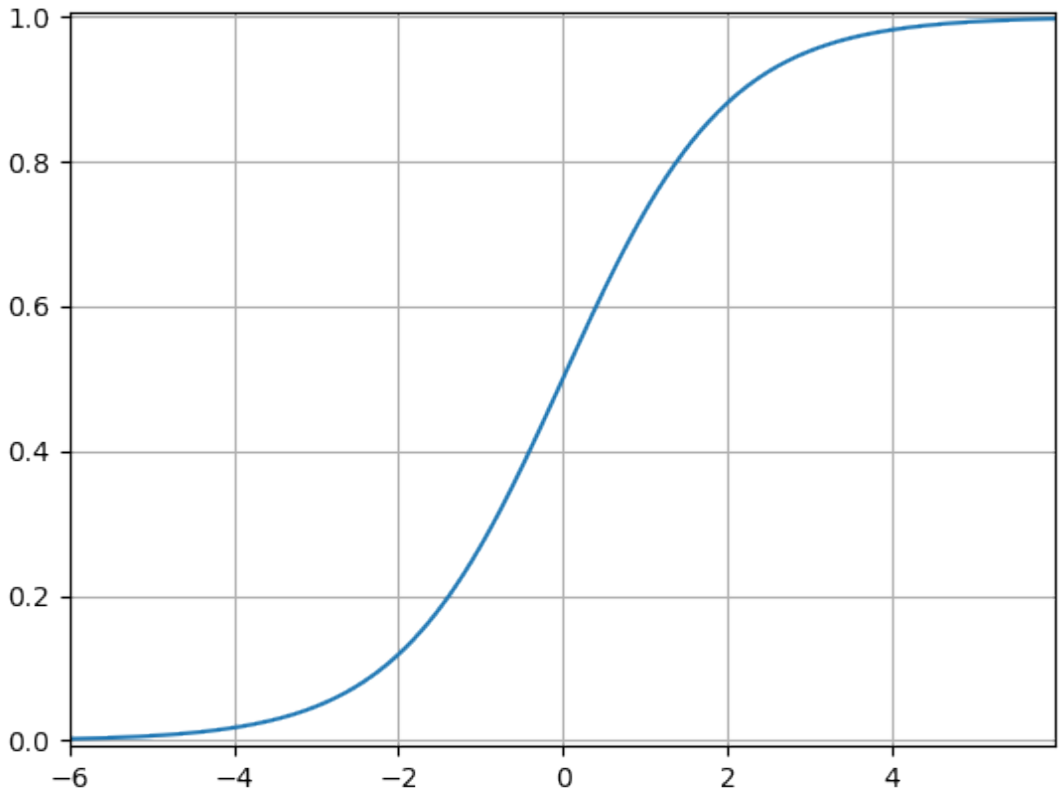
- aumenti quando la funzione lineare aumenta
- sia 50% quando la funzione lineare vale zero

Come convertire $w^\top x$ in una probabilità?



La funzione sigmoide (sigmoide logistica)

$$\sigma(z) = \frac{1}{1 + e^{-z}} \qquad \in [0, 1]$$



La classe di ipotesi della regressione logistica binaria

Nella *regressione logistica*, l'insieme delle ipotesi è l'insieme $\mathcal{H}_{sig}$ delle funzioni ottenute componendo la sigmoide con una funzione lineare da $\mathcal{X} \subseteq \mathbb{R}^{d+1}$ a $\mathbb{R}$:

$$h \in \mathcal{H}_{sig} \quad \Leftrightarrow \quad h(x) = \sigma(w^\top x) \text{ per qualche } w \in \mathbb{R}^{d+1}$$

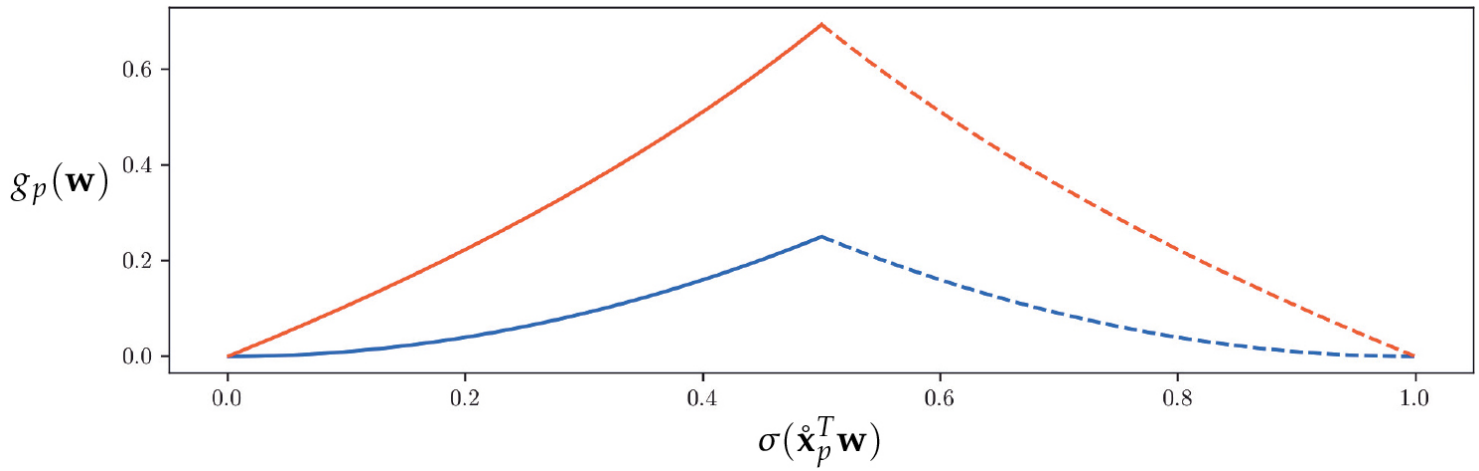
Funzione costo nella regressione logistica (etichette 0/1)

La regressione logistica si basa sulla seguente funzione costo:

Funzione costo *cross-entropia* (etichette 0/1)

$$\ell(h,(x,y)) = \begin{cases} -\log h(x) & \text{se } y = 1 \\ -\log(1-h(x)) & \text{se } y = 0 \end{cases}$$

È una funzione convessa nel vettore w dei parametri



Possiamo minimizzare il rischio empirico con i metodi gradiente

Rischio empirico nella regressione logistica (etichette 0/1)

ERM nella regressione logistica (etichette 0/1)

Dato il dataset S, trova $w \in \mathbb{R}^{d+1}$ che minimizza

$$\sum_{i=1}^{m} \left[-y^{(i)} \log \hat{y}^{(i)} - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

dove $\hat{y}^{(i)} = h(x^{(i)}) = \sigma(w^\top x^{(i)})$

Il problema di ottimizzazione corrispondente è convesso

$\Rightarrow$ Possiamo trovare w attraverso metodi del gradiente

Si possono usare anche metodi del secondo ordine (la funzione è sia convessa che ovunque differenziabile)

Funzione costo nella regressione logistica (etichette ± 1)

Con etichette ± 1, la funzione di costo è la seguente:

Funzione costo *log-loss* o *softmax* (etichette ± 1)

$$\ell(h, (x, y)) = \log(1 + \exp(-y \cdot w^\top x))$$

Rischio empirico nella regressione logistica (etichette ± 1)

ERM nella regressione logistica (etichette ± 1)

Dato il dataset S, trova $w \in \mathbb{R}^{d+1}$ che minimizza

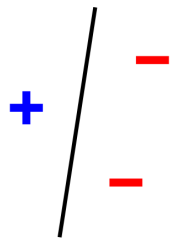
$$\sum_{i=1}^{m} \log(1 + \exp(-y^{(i)} \cdot w^\top x^{(i)}))$$

Classificazione binaria e separabilità lineare

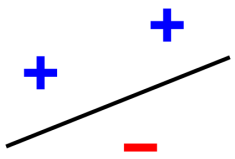
Separabilità lineare

Un insieme di esempi (x, y) con etichette di due tipi ($+$ e $-$) è *linearmente separabile* se esiste $w \in \mathbb{R}^{d+1}$ tale che:

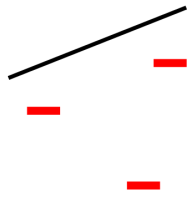
- $w^\top x > 0$ ogniqualvolta x è un esempio di tipo $+$
- $w^\top x < 0$ ogniqualvolta x è un esempio di tipo $-$



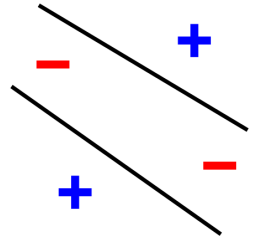
separabile



separabile



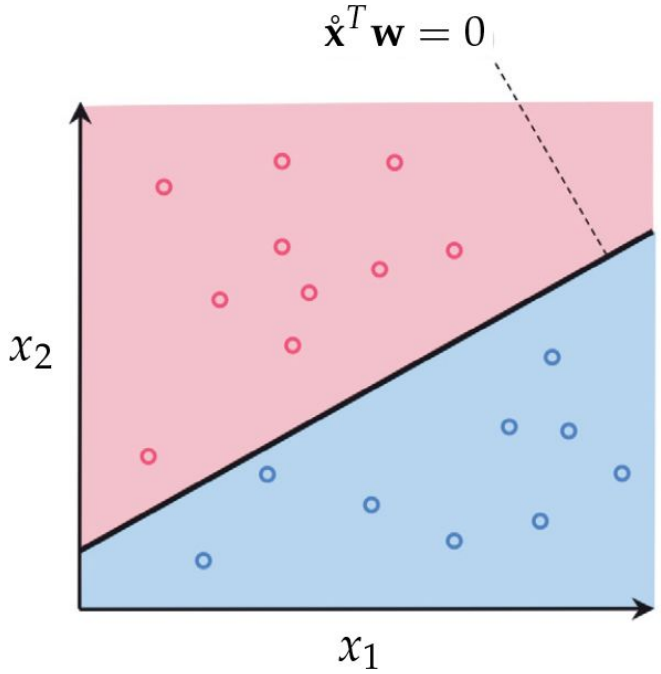
separabile



non separabile

Valori $y \cdot w^\top x$ e separabilità lineare

Supponiamo che il vettore w separi perfettamente gli esempi positivi e negativi:



Questo significa che per ogni (x, y),

- $y = +1$ e $w^\top x > 0$, oppure
- $y = -1$ e $w^\top x < 0$

In altre parole, $-y \cdot w^\top x$ è sempre < 0

Costo softmax e separabilità lineare

Se w separa perfettamente gli esempi positivi e negativi:

$$-y \cdot w^\top x < 0 \qquad \text{per ogni } (x, y)$$

allora qualunque sia il costo softmax

$$\log\left[1 + \exp(-y \cdot w^\top x)\right] > 0$$

possiamo diminuirlo semplicemente usando $2w$ invece di w:

$$\log\left[1 + \exp(-y \cdot 2w^\top x)\right] < \log\left[1 + \exp(-y \cdot w^\top x)\right]$$

Questo significa che il minimo della funzione si ha prendendo $\|w\| \to \infty$!

Il divergere di w può causare instabilità numerica negli algoritmi

Regressione logistica regolarizzata

Per evitare il divergere di w, si può considerare il problema di ottimizzazione vincolata

$$\begin{aligned} \underset{w \in \mathbb{R}^{d+1}}{\text{minimize}} \quad & \sum_{i=1}^{m} \log \left[1 + \exp(-y^{(i)} \cdot w^\top x^{(i)}) \right] \\ \text{subject to} \quad & \|\omega\|_2^2 \leq 1 \end{aligned}$$

che (per qualche $\lambda > 0$) equivale alla seguente formulazione

Regressione logistica con regolarizzazione ℓ_2

$$\underset{w \in \mathbb{R}^{d+1}}{\text{minimize}} \sum_{i=1}^{m} \log \left[1 + \exp(-y^{(i)} \cdot w^\top x^{(i)}) \right] + \lambda \|\omega\|_2^2$$

NB. Qui $\omega = (w_1, w_2, \ldots, w_d)$ è il vettore w senza la componente w_0

Regressione logistica in `scikit-learn`

Possibilità 1: usare la classe `LogisticRegression`

Regressione logistica	Iperparametri	Interfaccia `scikit-learn`
Non regolarizzata	nessuno	`LogisticRegression(penalty='none')`
Regolarizzata	$C\ (= \frac{1}{2\lambda})$	`LogisticRegression(penalty='l2', C)`

Possibilità 2: invocare Stochastic Gradient Descent con funzione costo *log*:

Regressione logistica (± 1)	Iperparametri	Interfaccia `scikit-learn`
Regolarizzata	$\alpha\ (= \lambda), \eta, T$	`SGDClassifier(loss='log', alpha, penalty='l2', max_iter...)`

Classificazione binaria: metriche di qualità

Doppio ruolo delle funzioni di costo

La funzione di costo per misurare la qualità delle predizioni nei problemi di classificazione è in genere la funzione costo 0-1:

$$\ell(h,(x,y)) = \begin{cases} 0 & \text{se } h(x) = y \\ 1 & \text{se } h(x) \neq y \end{cases}$$

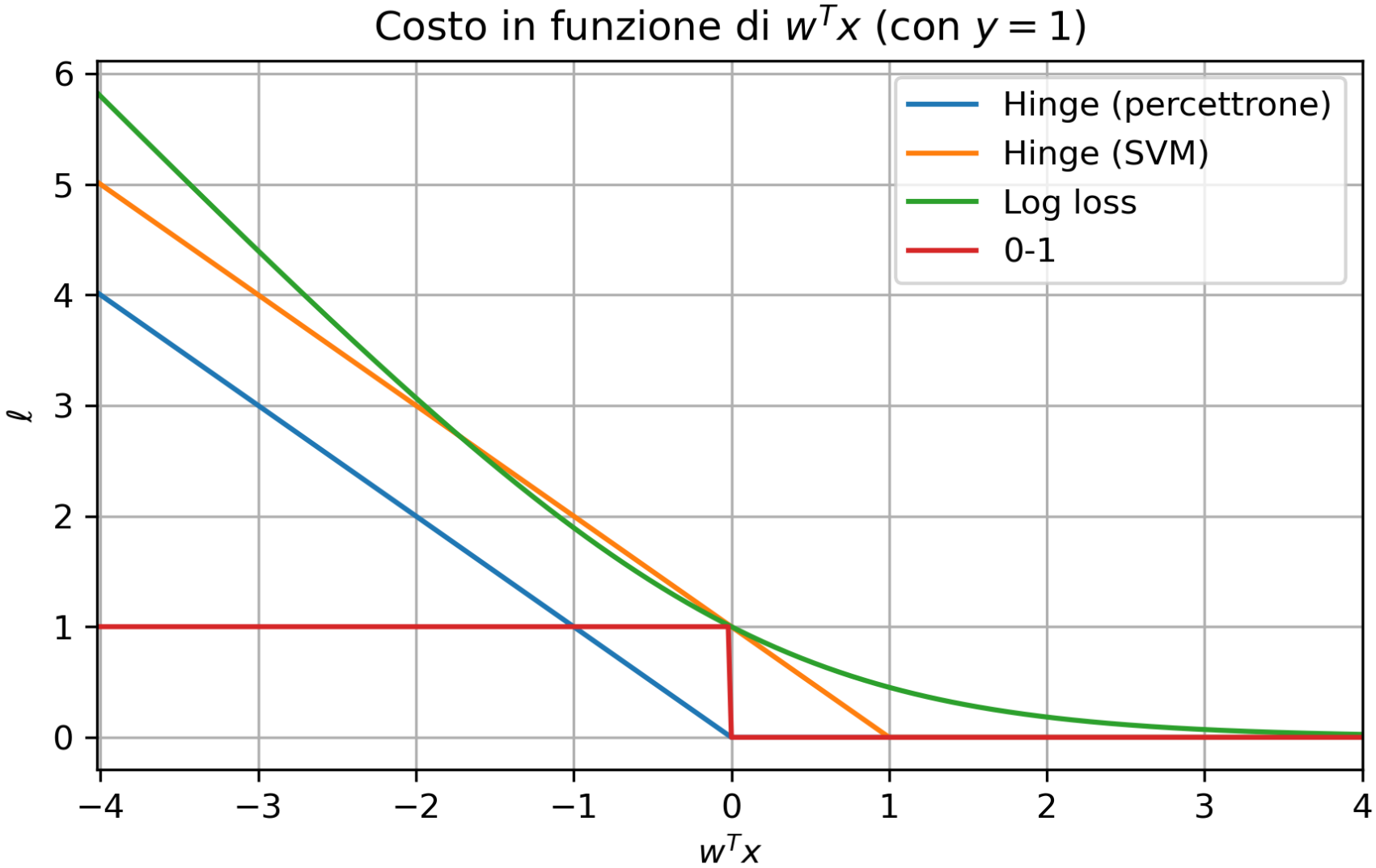
$\Rightarrow$ *Accuratezza*: frazione di nuovi esempi correttamente classificati

La funzione di costo per apprendere il modello è un suo *surrogato*, tipicamente convesso, per motivi computazionali:

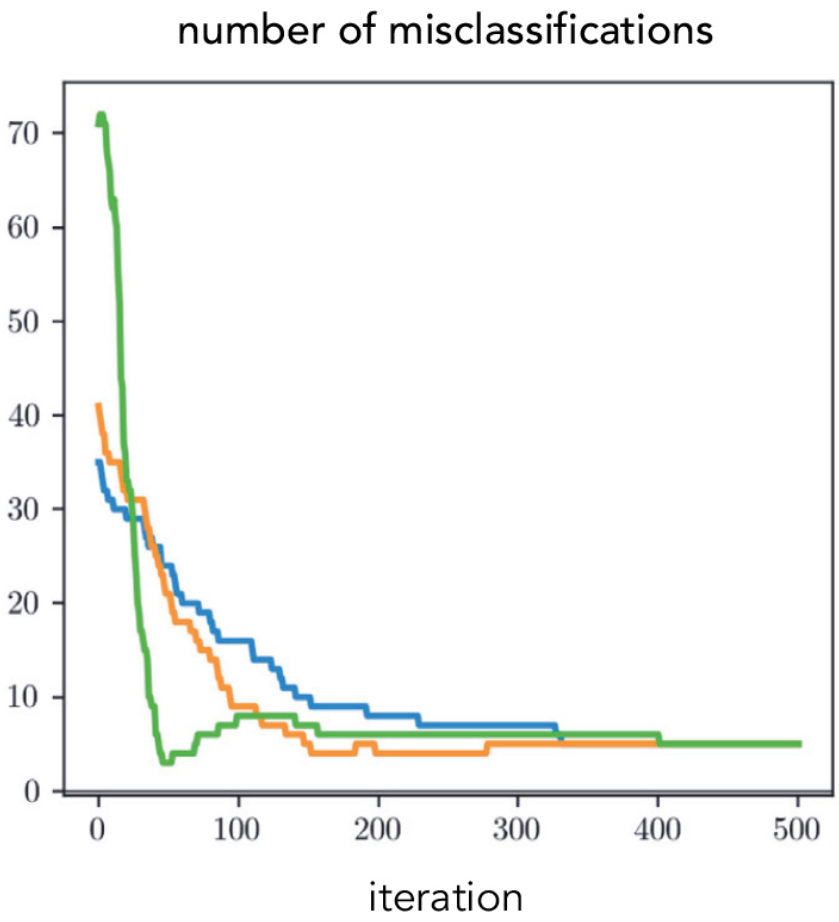
- Log-loss / softmax (regressione logistica)
- Hinge loss (percettrone)...

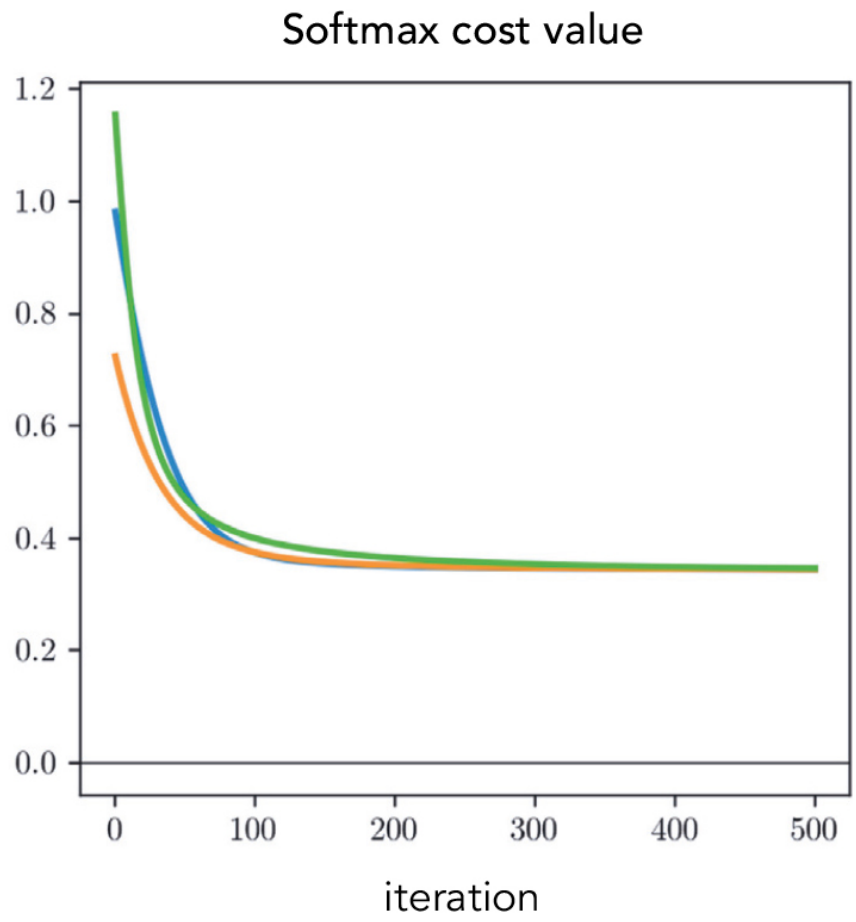
Il modello va validato sulla funzione originale, non sul surrogato
$\Rightarrow$ utilizziamo l'(in)accuratezza come metrica durante validazione e test

Costi surrogati vs. costo 0-1



Esempio


number of misclassifications
70
60
50
40
30
20
10
0
0
100
200
300
400
500
iteration


Softmax cost value
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
iteration

Inconvenienti dell'accuratezza come metrica

L'accuratezza a volte può presentare inconvenienti

- se diversi tipi di misclassificazione hanno costo molto diverso
- se c'è uno sbilanciamento tra le classi, in cui i casi positivi (o quelli negativi) sono estremamente rari

Esempio: diagnosi di una malattia rara ($<$1 per mille della popolazione)
Un classificatore che restituisce sempre `NO` ha accuratezza 99.9%, ma in effetti è del tutto inutile

Matrice di confusione

Per problemi con forte sbilanciamento, è utile separare i tipi di errore attraverso una *matrice di confusione*

	$y = 1$	$y = 0$
$\hat{y} = 1$	*Veri Positivi* Abbiamo urlato al lupo! Abbiamo salvato il villaggio.	*Falsi Positivi* Errore: il lupo non c'era. Abbiamo innervosito tutti.
$\hat{y} = 0$	*Falsi Negativi* C'era un lupo, ma non lo abbiamo stanato. Ha mangiato tutto il pollame.	*Veri Negativi* Nessun lupo, nessun allarme. Tutto tranquillo.

Accuratezza e matrice di confusione

Accuratezza [accuracy]

$$\mathcal{A} = \frac{VP + VN}{VP + VN + FP + FN}$$

Sensibilità, specificità e precisione

Sensibilità [sensitivity, recall]

$$\text{sensibilità} \stackrel{\text{def}}{=} \frac{VP}{VP + FN} = \mathcal{A}_{y=1}$$

Specificità [specificity]

$$\text{specificità} \stackrel{\text{def}}{=} \frac{VN}{VN + FP} = \mathcal{A}_{y=0}$$

Precisione [precision]

$$\text{precisione} \stackrel{\text{def}}{=} \frac{VP}{VP + FP} = \mathcal{A}_{\hat{y}=1}$$

Accuratezza bilanciata

Accuratezza bilanciata

$$\begin{aligned}\mathcal{A}_{\text{balanced}} &\stackrel{\text{def}}{=} \frac{\mathcal{A}_{y=1} + \mathcal{A}_{y=0}}{2} \\ &= \frac{1}{2}\,\text{sensibilità} + \frac{1}{2}\,\text{specificità} \\ &= \frac{1}{2}\frac{VP}{VP + FN} + \frac{1}{2}\frac{VN}{VN + FP}\end{aligned}$$

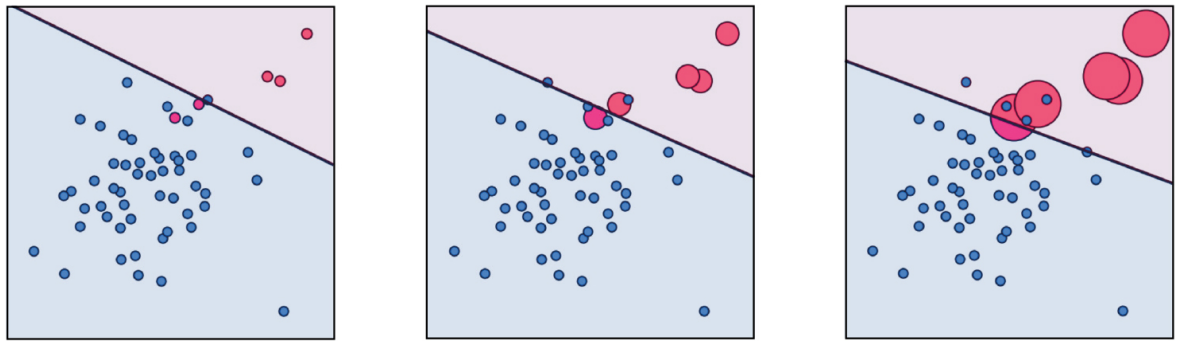
Attenzione. La formula nel libro di testo (Watt et al.) non è corretta (confonde la sensibilità con la precisione)

Pesatura degli esempi

Per regolare l'influenza degli esempi di una classe sbilanciata, possiamo assegnare ad ogni esempio un *peso* β_i, come visto per la regressione

Per esempio, nella regressione logistica possiamo minimizzare

$$\sum_{i=1}^{m} \beta_i \log(1 + \exp(-y^{(i)} \cdot w^\top x^{(i)}))$$



Ad esempio, si può prendere β_i inversamente proporzionale alla taglia della classe $y^{(i)}$