Classificazione discriminativa

Vincenzo Bonifaci

IN550 – Machine Learning

Classificazione generativa vs. discriminativa

Approccio generativo

- Stima $\Pr(x, y)$ per poi dedurre $\Pr(y|x)$
- Confronta le $\Pr(y|x)$ per trovare la classe più verosimile

Esempi: QDA, LDA, Naive Bayes

Approccio discriminativo

- Stima $\Pr(y|x)$
- Confronta le $\Pr(y|x)$ per trovare la classe più verosimile

Oppure, evitando completamente le probabilità,

- Costruisci direttamente una funzione da $\mathcal{X}$ a $\mathcal{Y}$

Esempi: Regressione logistica, Percettrone, Support Vector Machines

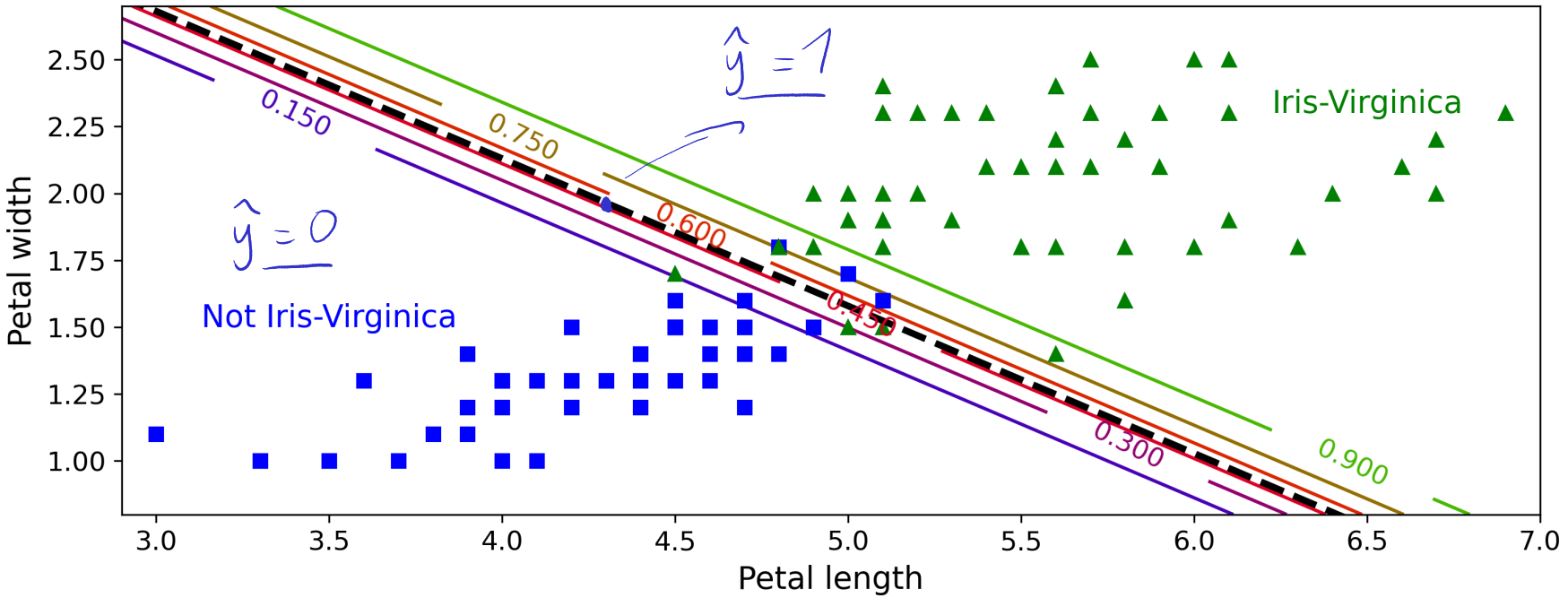
Stima di probabilità per etichette binarie

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 $h(x) \approx \Pr(y = 1|x)$

$y \neq \tilde{0}$ $\quad (\Pr(y=0|x) = 1 - \Pr(y=1|x))$



Un modello lineare per la stima di probabilità?

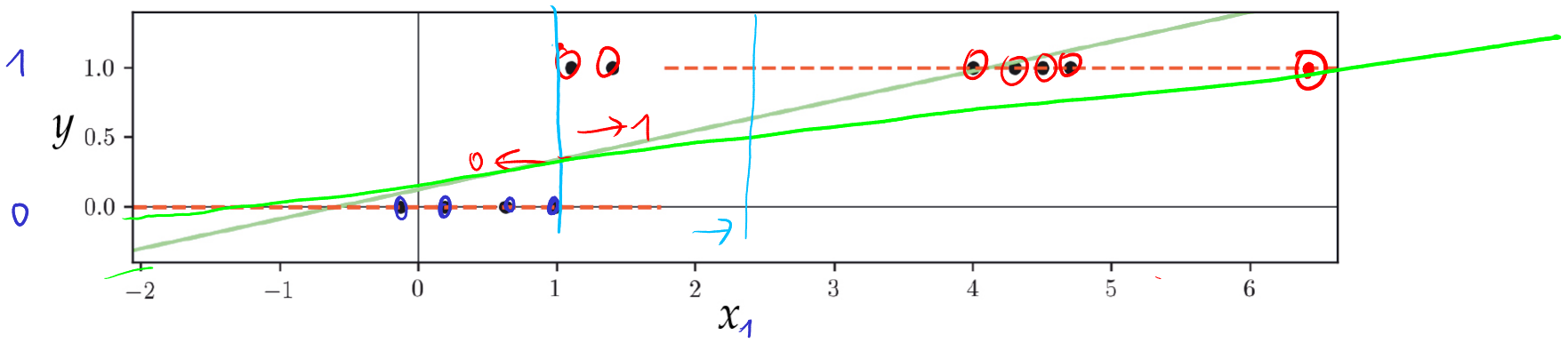
Dato x, vogliamo stimare $\Pr(y = 1|x)$ attraverso una funzione lineare

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

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

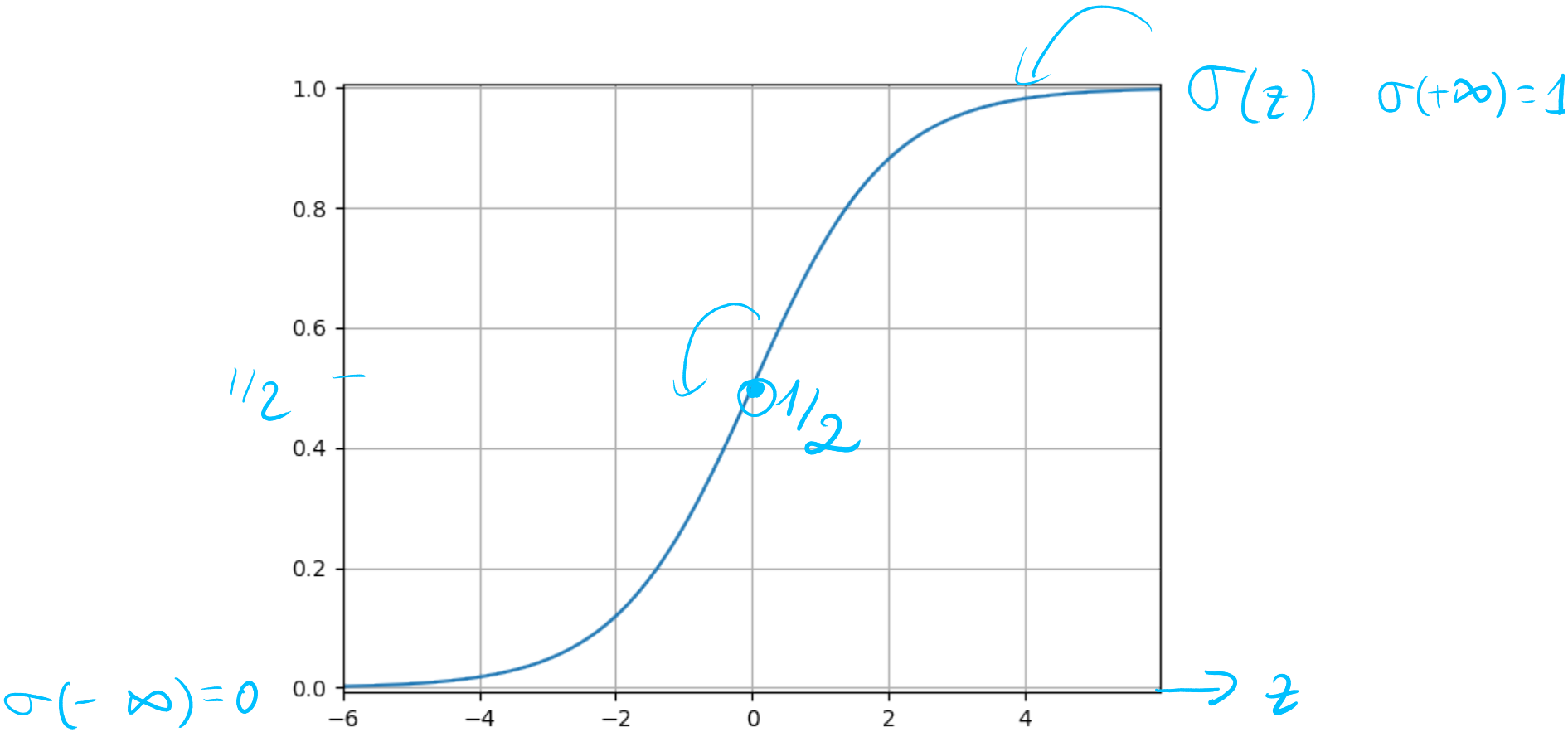
- aumenti quando la funzione lineare aumenta
- sia 50% quando la funzione lineare vale zero

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



La funzione sigmoide (sigmoide logistica)

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



Alcune proprietà della sigmoide

Se $\sigma(z) = (1 + \exp(-z))^{-1}$, allora per ogni $z \in \mathbb{R}$,

$$1 - \sigma(z) = \sigma(-z)$$

Se $\sigma(z) = (1 + \exp(-z))^{-1}$, allora per ogni $z \in \mathbb{R}$,

$$\sigma'(z) = \sigma(z) \cdot (1 - \sigma(z))$$

Regressione logistica binaria (etichette 0/1)

Assumiamo che:

$$\Pr(y = 1|x) = \sigma(w^\top x) = \frac{1}{1 + \exp(-w^\top x)}$$

Ne consegue che:

$$\Pr(y = 0|x) = 1 - \sigma(w^\top x) = \sigma(-w^\top x) = \frac{1}{1 + \exp(w^\top x)}$$

Qualunque sia $y \in \{0, 1\}$, possiamo quindi scrivere

$$\Pr(y|x) = (h(x))^y \, (1 - h(x))^{1-y}$$

dove $h(x) = \sigma(w^\top x)$

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 $\mathbb{R}$:

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

Stima delle probabilità

$\Pr(y = 1 | x)$

MLE nella regressione logistica (etichette 0/1)

Principio di massima verosimiglianza

Dati gli esempi $S = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, scegli $w \in \mathbb{R}^{d+1}$ che massimizza la funzione di *verosimiglianza* [*likelihood*]:

$$\mathcal{L}(w) = \prod_{i=1}^{m} \Pr(y^{(i)} | x^{(i)}; w)$$

Massimizzare $\mathcal{L}(w)$ equivale a minimizzare la *cross-entropia*:

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

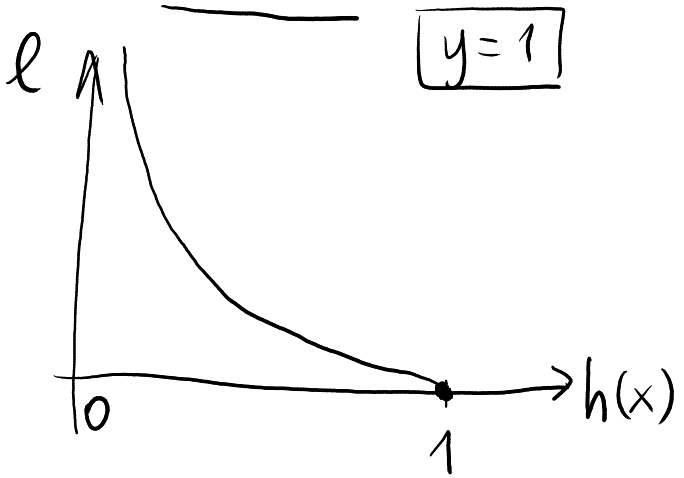
Funzione costo nella regressione logistica (etichette 0/1)

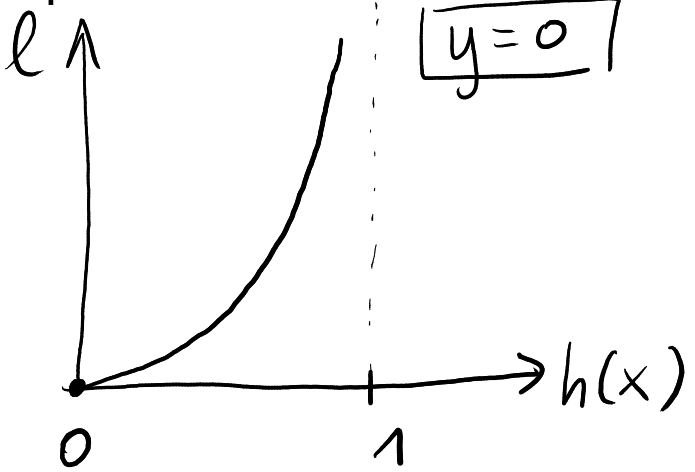
In altre parole stiamo assumendo la seguente funzione costo:

Funzione *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 $(h(x) = \sigma(w^\top x))$





ERM nella regressione logistica (etichette 0/1)

Il principio MLE in questo caso è quindi equivalente al principio Empirical Risk Minimization con la funzione obiettivo cross-entropia

ERM nella regressione logistica

Dati gli esempi $S = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$,
scegli $w \in \mathbb{R}^{d+1}$ che minimizza

$$L_S(w) = -\sum_{i=1}^{m} \log \Pr(y^{(i)} | x^{(i)}; w)$$

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

SGD per la regressione logistica (etichette 0/1)

Possiamo minimizzare il costo con i metodi gradiente

Per esempio con SGD: $w \leftarrow w - \eta \nabla \ell(h_w)$

Calcolando $\nabla \ell(h_w)$ e sfruttando $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ otteniamo

$$\nabla \ell(h_w) = (h(x) - y)x$$

Regola SGD per la regressione logistica (0/1)

$$w \leftarrow w - \eta \cdot (h(x) - y) \cdot x$$

NB. La regola ha la stessa struttura della regola Least Mean Squares (LMS), ma il significato di $h(x)$ è differente

Regressione logistica binaria (etichette ± 1)

Assumiamo che:

$$\Pr(y = +1|x) = \sigma(w^\top x) = \frac{1}{1 + \exp(-w^\top x)}$$

Ne consegue che:

$$\Pr(y = -1|x) = \sigma(-w^\top x) = \frac{1}{1 + \exp(w^\top x)}$$

In generale, per $y \in \{-1, +1\}$, possiamo scrivere

$$\Pr(y|x) = \sigma(y \cdot w^\top x) = \frac{1}{1 + \exp(-y \cdot w^\top x)}$$

Funzione costo nella regressione logistica (etichette ± 1)

La funzione di costo diventa la seguente:

Funzione costo logistica (etichette ± 1)

$$\begin{aligned}\ell(h,(x,y)) &= \log(1+\exp(-y\cdot w^\top x)) \\ &= -\log \Pr(y|x;w)\end{aligned}$$

Rischio empirico nella regressione logistica

Principio di massima verosimiglianza

Dati gli esempi $S = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, scegli $w \in \mathbb{R}^{d+1}$ che massimizza la funzione di *verosimiglianza* [*likelihood*]:

$$\prod_{i=1}^{m} \Pr(y^{(i)} | x^{(i)}; w)$$

Rischio empirico nella regressione logistica

Equivalentemente, passando al logaritmo, vogliamo *minimizzare* il rischio empirico, in linea col principio ERM:

ERM nella regressione logistica

Dati gli esempi $S = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$, scegli $w \in \mathbb{R}^d$ che minimizza

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

Minimizzazione di $L_S(w)$ nella regressione logistica

Rischio empirico nella regressione logistica

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

- Il minimizzatore w^* non è esprimibile in forma chiusa
- Ma $L_S(w)$ è una funzione convessa in w

$\Rightarrow$ Il problema di ottimizzazione corrispondente è convesso

$\Rightarrow$ Possiamo trovare w^* attraverso Gradient Descent o le sue varianti

SGD per la regressione logistica

Per derivare la regola di aggiornamento di SGD ricapitoliamo le assunzioni:

- Ipotesi in $\mathcal{H}_{sig}$: $h_w(x) = \sigma(w^\top x)$
- Costo logistico: $\ell(h_w) = \log(1 + \exp(-y \cdot w^\top x))$

Prendendo le derivate parziali di ℓ, otteniamo

$$\begin{aligned}
\frac{\partial \ell}{\partial w_j}(h_w) &= \frac{1}{1 + \exp(-yw^\top x)} \frac{\partial}{\partial w_j} \left[1 + \exp(-yw^\top x)\right] \\
&= \frac{\exp(-yw^\top x)}{1 + \exp(-yw^\top x)} \frac{\partial}{\partial w_j} \left[-yw^\top x\right] \\
&= -\frac{1}{1 + \exp(+yw^\top x)} \cdot y \cdot x_j \\
&= -\Pr(-y|x; w) \cdot y \cdot x_j
\end{aligned}$$

SGD per la regressione logistica

In forma vettoriale,

$$\nabla \ell(h_w) = -\Pr(-y|x; w) \cdot y \cdot x$$

La regola di aggiornamento SGD è quindi

$$w \leftarrow w + \eta \cdot \Pr(-y|x;\ w) \cdot y \cdot x$$

Regressione logistica multiclasse

Come trattare il caso di K classi con $K > 2$?

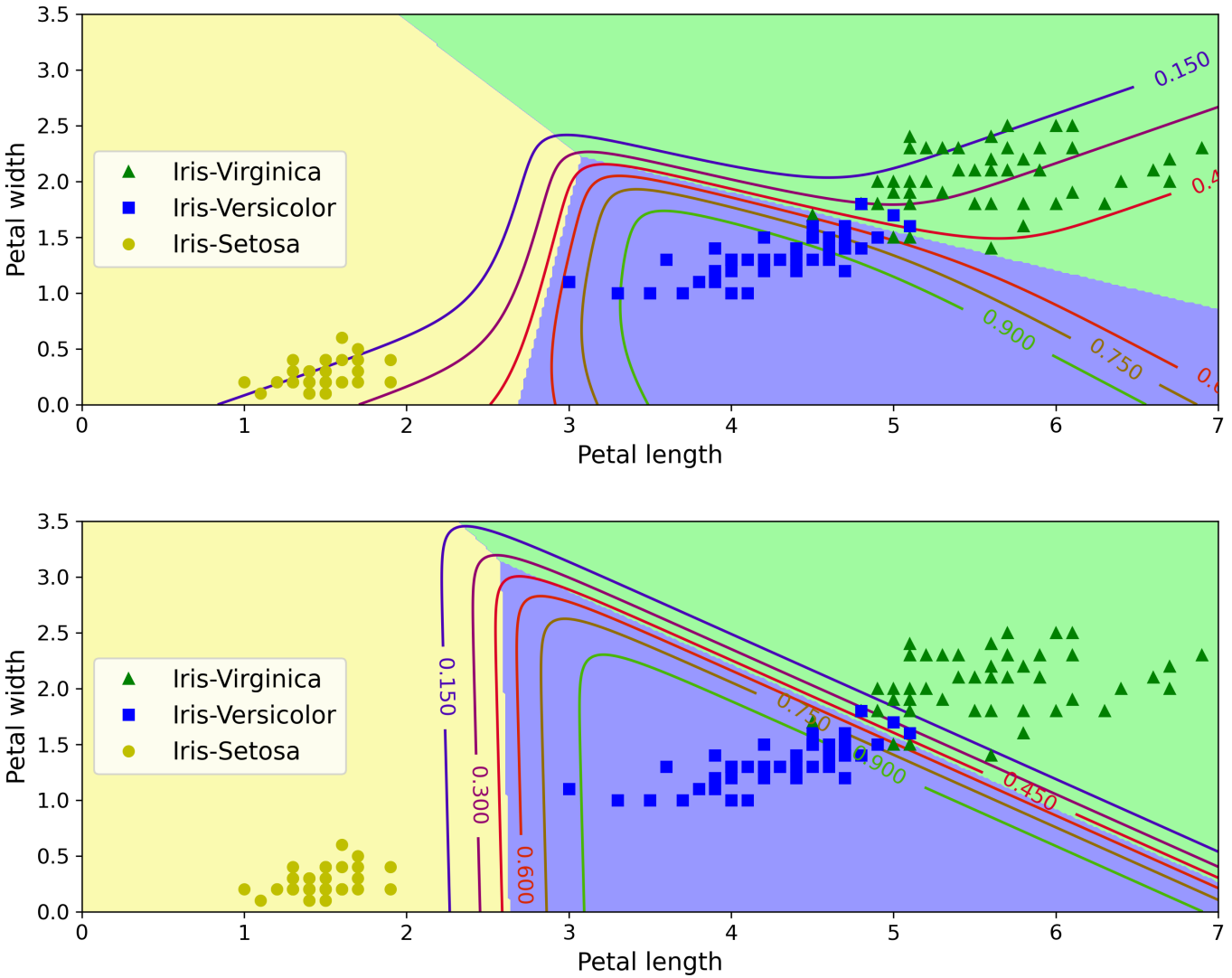
Approccio *One vs. rest*

1. Apprendi K ipotesi $h^{(1)}, \ldots, h^{(K)}$, dove la j-esima ipotesi distingue la classe j dalle altre $K - 1$ classi
2. Dato x, restituisci la classe j che massimizza $h^{(j)}(x)$

Approccio multinomiale (*Softmax regression*)

1. Assumi $\Pr(y = j|x) = \frac{\exp(w^{(j)\top} x)}{\sum_k \exp(w^{(k)\top} x)}$
2. Ottimizza i vettori $w^{(1)}, \ldots, w^{(K-1)}$ per massimizzare la likelihood (senza perdita di generalità, $w^{(K)} = 0$)
3. Dato x, restituisci la classe j che massimizza $\Pr(y = j|x)$

Regressione logistica multiclasse: Esempio



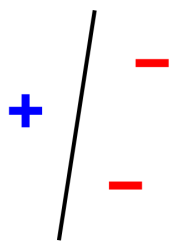
Percettroni e Support Vector Machines

Classificazione binaria e separabilità lineare

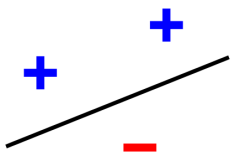
Separabilità lineare

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:

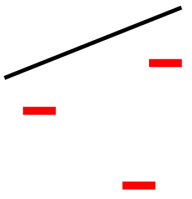
- $w^\top x > 0$ ogniqualvolta x è un esempio di tipo $+$
- $w^\top x < 0$ ogniqualvolta x è un esempio di tipo $-$



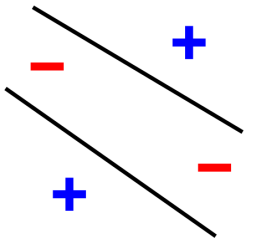
separabile



separabile



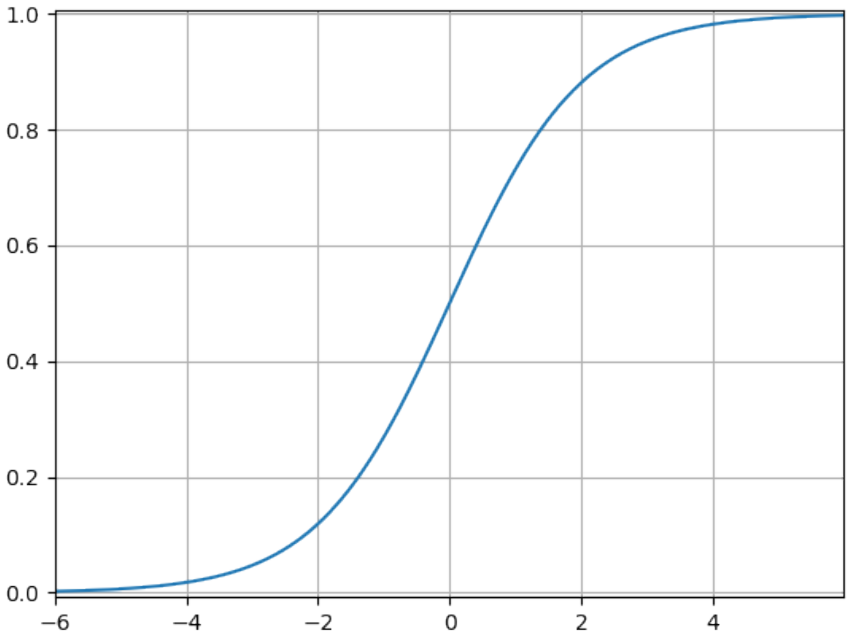
separabile



non separabile

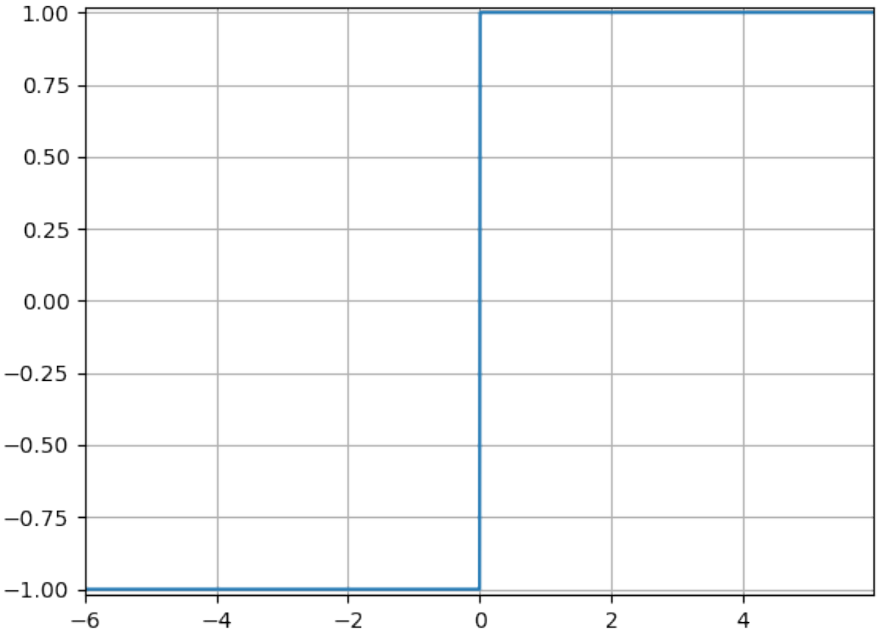
Il regressore logistico

- $x = (1, x_1, \ldots, x_d)$, $y \in [0, 1]$
- $w = (w_0, \ldots, w_d)$
- $h_w(x) = \sigma(w^\top x) = \frac{1}{1+\exp(-w^\top x)}$



Il Percettrone

- $x = (1, x_1, \ldots, x_d)$, $y \in \{-1, +1\}$
- $w = (w_0, \ldots, w_d)$
- $h_w(x) = \text{sign}(w^\top x)$



Se la vera etichetta è y, la predizione è corretta $\Leftrightarrow y \cdot w^\top x > 0$

La classe di ipotesi del Percettrone

Nel *Percettrone*, l'insieme delle ipotesi è l'insieme $\mathcal{H}_P$ delle funzioni ottenute componendo la funzione segno con una funzione lineare da $\mathcal{X}$ a $\mathbb{R}$:

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

Una funzione di costo per il Percettrone

La funzione più naturale (costo 0-1) purtroppo è ardua da ottimizzare!

Cerchiamo di costruire un surrogato:

- Se $y \cdot w^\top x > 0$, poniamo costo $= 0$ (la predizione è corretta)
- Se $y \cdot w^\top x \leq 0$, poniamo costo $= -y \cdot w^\top x$

Hinge Loss (per etichette ± 1)

$$\ell(h, (x, y)) \stackrel{\text{def}}{=} \max(-y \cdot w^\top x, 0)$$

Per ogni (x, y), questa funzione costo è convessa, quindi possiamo usarla con i metodi gradiente!

SGD per il Percettrone

- Se $y \cdot w^\top x > 0$, allora $\ell = 0$ (la predizione è corretta)
- Se $y \cdot w^\top x \leq 0$, allora $\ell = -y \cdot w^\top x$

Calcolando il gradiente del costo, otteniamo:

- Nel primo caso, $\nabla \ell = 0$
- Nel secondo caso, $\nabla \ell = -y \cdot x$

Regola di aggiornamento SGD per il Percettrone

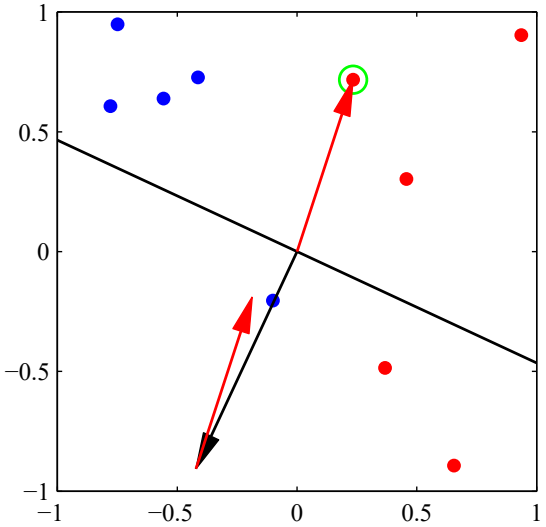
- Se $y \cdot w^\top x > 0$, poni $w^{(t+1)} \leftarrow w^{(t)} - \eta \cdot 0$
- Se $y \cdot w^\top x \leq 0$, poni $w^{(t+1)} \leftarrow w^{(t)} + \eta \cdot y \cdot x$

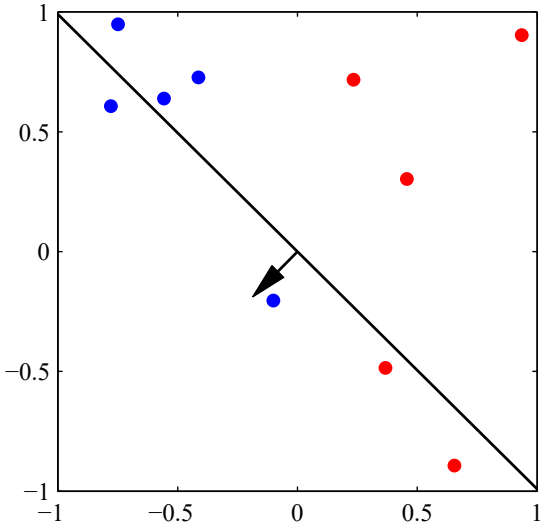
Algoritmo del Percettrone

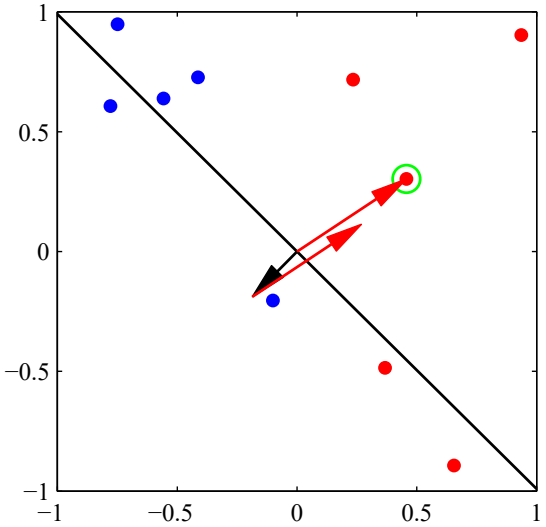
Algoritmo del Percettrone

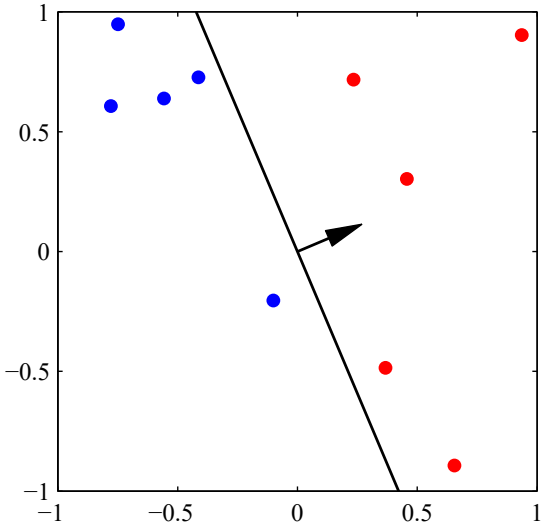
1. Inizializza $w^{(1)} = 0_{(d+1)\times 1}$
2. Per $t = 1, 2, \ldots,$, considera ciclicamente ogni esempio (x, y):
   - Se $y \cdot w^\top x \leq 0$, aggiorna $w^{(t+1)} \leftarrow w^{(t)} + \eta\, y \cdot x$

Algoritmo del Percettrone









Convergenza del Percettrone

Teorema (Convergenza del Percettrone)

Se il training set è linearmente separabile:

- L'algoritmo del Percettrone trova un'ipotesi con rischio empirico pari a zero
- L'algoritmo converge in un numero finito di passi

Forma duale del Percettrone

Algoritmo del Percettrone

1. Inizializza $w = 0_{(d+1)\times 1}$
2. Per $t = 1, 2, \ldots,$ considera ciclicamente ogni esempio (x, y):
   - Se (x, y) è misclassificato, aggiorna $w \leftarrow w + \eta\, y \cdot x$

Quindi l'output dell'algoritmo avrà la forma

$$w = \eta \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

dove α_i è il # di volte che l'esempio i-esimo ha causato un aggiornamento

$\Rightarrow$ Possiamo rappresentare w tramite $\alpha = (\alpha_1, \ldots, \alpha_m)$ (variabili *duali*)

Forma duale del percettrone

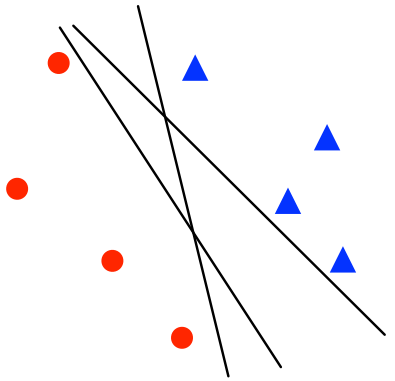
Algoritmo del Percettrone (forma primale)

1. Inizializza $w = 0_{(d+1)\times 1}$
2. Per $t = 1, 2, \ldots,$ considera ciclicamente ogni esempio $(x^{(i)}, y^{(i)})$:
   - Se $(x^{(i)}, y^{(i)})$ è misclassificato, aggiorna $w \leftarrow w + \eta\, y^{(i)} \cdot x^{(i)}$

Algoritmo del Percettrone (forma duale)

1. Inizializza $\alpha = 0_{m\times 1}$
2. Per $t = 1, 2, \ldots,$ considera ciclicamente ogni esempio $(x^{(i)}, y^{(i)})$:
   - Se $(x^{(i)}, y^{(i)})$ è misclassificato, poni $\alpha_i \leftarrow \alpha_i + 1$
3. Restituisci $w = \eta \sum_i \alpha_i y^{(i)} x^{(i)}$

Oltre il Percettrone



Domanda: Possiamo selezionare il separatore più "robusto"?

Separazione robusta

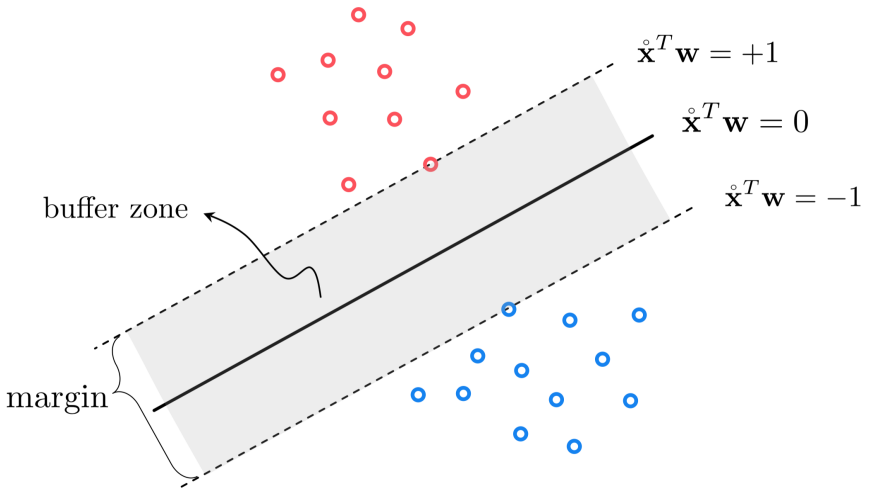
Input: m esempi $(x, y) \in \mathbb{R}^{d+1} \times \{-1, +1\}$
Output: $w \in \mathbb{R}^{d+1}$ tale che $y \cdot w^\top x > 0$ per ogni esempio (x, y)

Scalando w, è equivalente a richiedere

$$y \cdot w^\top x \geq 1 \qquad \text{per ogni esempio } (x, y)$$

Margine di un separatore lineare

$$y \cdot w^\top x \geq 1 \qquad \text{per ogni esempio } (x, y)$$



Il *margine* è $2/\left\|(w_1, \ldots, w_d)\right\|$

$\Rightarrow$ massimizzare il margine $\equiv$ minimizzare $\sqrt{w_1^2 + w_2^2 + \ldots + w_d^2}$

Separazione a massimo margine

Hard-margin Support Vector Machine

$$\min_{w} w_1^2 + w_2^2 + \ldots + w_d^2$$

$$\text{t.c. } y \cdot w^\top x \geq 1 \text{ per ogni esempio } (x, y)$$

È un problema di minimizzazione convessa vincolata:

- funzione obiettivo convessa
- vincoli lineari

$\Rightarrow$ è risolvibile in modo efficiente

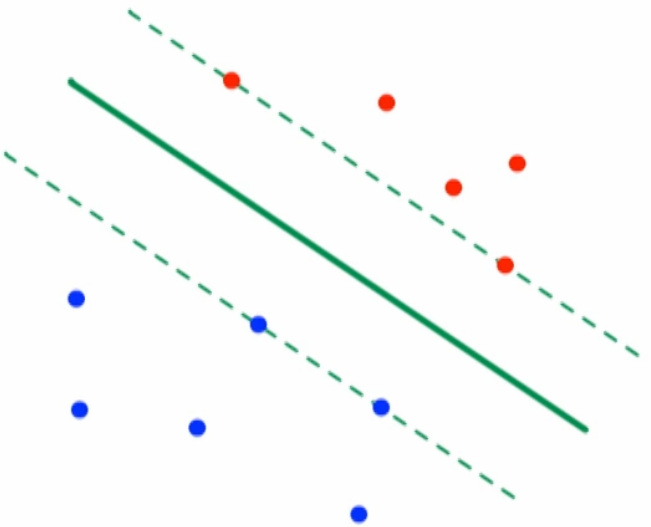
Importante: ha soluzione solo se gli esempi sono linearmente separabili

Vettori di supporto

Si può dimostrare che la soluzione ottima ha la forma

$$w^* = \sum_{i=1}^{m} \alpha_i \cdot y^{(i)} \cdot x^{(i)}$$

e quindi dipende solo dai *vettori di supporto*: gli $x^{(i)}$ giacenti sul margine



Ma cosa fare se il dataset non è linearmente separabile?

Il caso non separabile

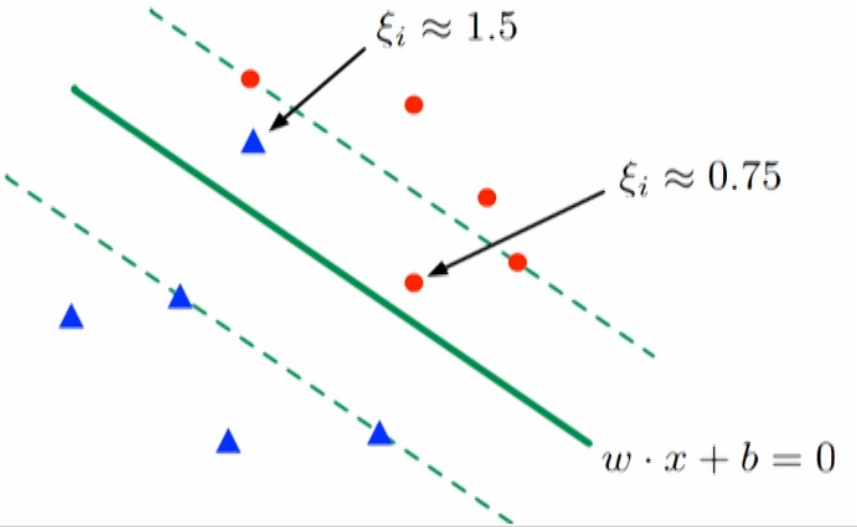
Introduciamo delle variabili di *slack* ("allentamento"):

Soft-Margin Support Vector Machine

$$\min_{w} w_1^2 + w_2^2 + \ldots + w_d^2 + C \sum_{i=1}^{m} \xi_i$$

$$\text{t.c. } y^{(i)} \cdot w^\top x^{(i)} \geq 1 - \xi_i \text{ per } i = 1, 2, \ldots, m$$

$$\xi \geq 0$$



Compromesso tra margine e slack

Che ruolo gioca l'iperparametro C?

Soft-Margin Support Vector Machine

$$\min_{w} w_1^2 + w_2^2 + \ldots + w_d^2 + C \sum_{i=1}^{m} \xi_i$$

$$\text{t.c. } y^{(i)} \cdot w^\top x^{(i)} \geq 1 - \xi_i \text{ per } i = 1, 2, \ldots, m$$

$$\xi \geq 0$$

$C = 0$: restituisce $w^* = 0$ (gli allentamenti non sono penalizzati)

$C \to \infty$: equivalente a Hard-margin SVM