

Introduzione alla Classificazione

Programmazione di Applicazioni Data Intensive

Laurea in Ingegneria e Scienze Informatiche
DISI – Università di Bologna

Gianluca Moro

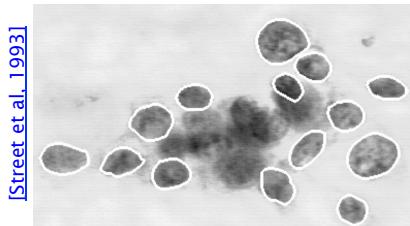
Dipartimento di Informatica – Scienza e Ingegneria
Università di Bologna
Via Venezia, 52 – I-47521 Cesena (FC)
Gianluca.Moro@Unibo.it

(draft)

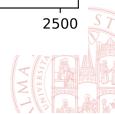
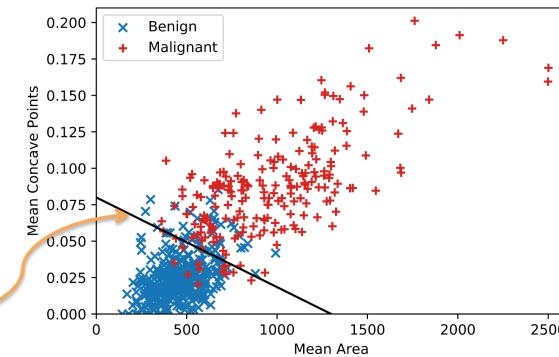


Introduzione alla Classificazione

Cosa Significa Classificare ?

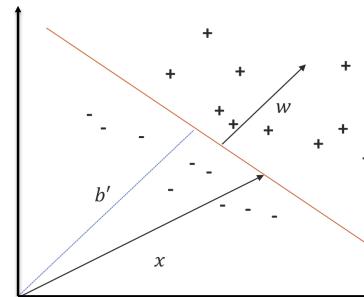


- A sinistra cellule cancerogene benigne e maligne descritte da alcune variabili
 - area, perimetro, consistenza, num. concavità, varianza scala di grigi ...
 - e per ognuna anche media, max, varianza
- a dx ogni punto è una cellula
 - rossa maligna, blu benigna: problema a 2 classi (2 insiemi)
 - qui con 2 variabili: num. medio delle concavità (y), area media delle concavità (x)
- **classificare == individuare una funzione che massimizzi la separazione tra le classi**



Classificazione Lineare con Iperpiani

- Metodi che assumono l'esistenza di separazioni lineari dei dati:
 - Individuazione di iperpiani di separazione delle classi con *programmazione lineare* oppure con soluzioni iterative, es. *Perceptron*
- un iper piano $w \cdot x + b = 0$ è definito da due parametri:
 - w è il vettore unitario perpendicolare all'iperpiano (retta rossa, in 2D $w_1x_1 + w_2x_2 + b = 0$)
 - b è la distanza dell'iperpiano dall'origine
 - $w \cdot x$ è il prodotto scalare tra vettori, i.e. proiezione di x nella direzione di w
 - i vettori x (istanze) che risiedono sulla retta, ossia tali che $w \cdot x + b = 0$, distano quindi b dall'origine i.e. $w \cdot x = b$
 - i punti x tali che $w \cdot x + b > 0$ sono sopra l'iperpiano (distanza maggiore di b), quelli sotto $w \cdot x + b < 0$



Gianluca Moro - DISI, Università di Bologna



Classificazione Lineare: Perceptron (1957)

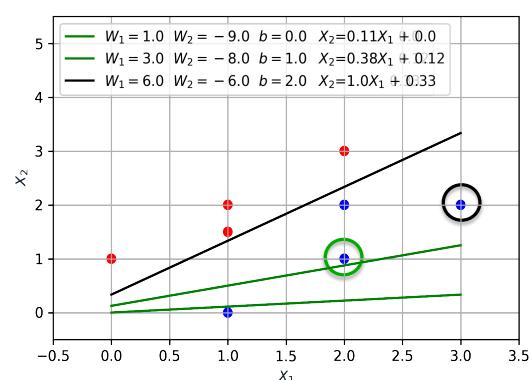
Esempio in 2D: Trovare la retta con parametri w_1, w_2, b tali che

$w_1x_1 + w_2x_2 + b > 0$ per i punti blu

$w_1x_1 + w_2x_2 + b < 0$ per i punti rossi

$\text{class}(x)$: +1 se x è blu, -1 se rosso

```
# Perceptron naive
# b = w_0x_0    x_0=1 for each instance x
w = [0, 1, -9] # initial random value
repeat
  classification_errors == False
  for each instance x
    if class(x) < 0 and w · x ≥ 0 # ERROR!
      w = w - x; classification_errors = True
    else if class(x) > 0 and w · x ≤ 0 # ERROR!
      w = w + x; classification_errors = True
  until classification_errors == True
```

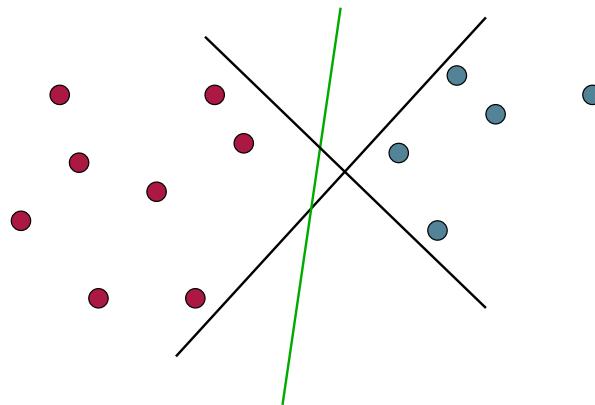


converge solo se i dati sono separabili linearmente
Progenitore delle Reti Neurali

Gianluca Moro - DISI, Università di Bologna



Quale Iperpiano?

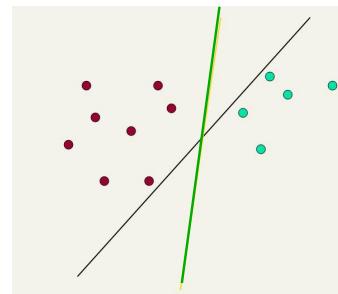


In generale molte possibili soluzioni per w_1, w_2, b



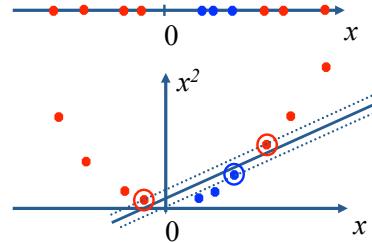
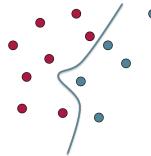
Quale Iperpiano ?

- Molte possibili soluzioni per w_1, w_2, b
- Alcuni metodi individuano un iperpiano di separazione non ottimale [in base ad un qualche criterio di ottimalità]
 - E.g., Perceptron, Regressione Logistica, *Regressione lineare con soglia*
- Altri individuano un iperpiano di separazione ottimale
 - Support Vector Machines
- Quali dati influenzano la ricerca dell'iperpiano ?
 - Tutti i punti
 - Perceptron, Regressione Logistica, Naïve Bayes, Regressione lineare con soglia
 - Solo i “punti difficili”, i.e. vicini al decision boundary
 - Support vector machines



Classificatori Non Lineari

- problemi con dati non separabili linearmente
- numerosi approcci
 - Support Vector Machines, Reti Neurali, kNN, Decision Tree, Gradient Boosting, XGboost,
 - ...
- soluzioni che trasformano lo spazio dei dati in modo che le classi diventino separabili linearmente
 - Support Vector Machines, Reti Neurali ...
- soluzioni intrinsecamente non lineari
 - kNN, Decision Tree, Gradient Boosting, XGboost, lighGBM, Catboost ...
- più variabili hanno i dati, più chance ci sono di separare le classi linearmente



11



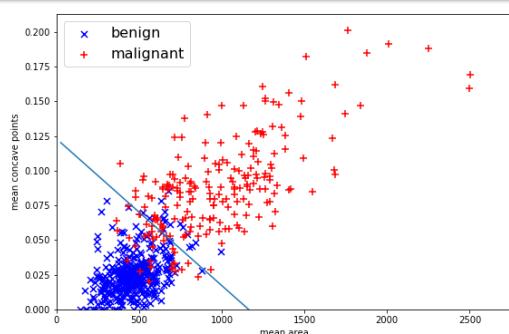
Formalizziamo l'Esempio Iniziale a 2 Classi

- variabili di input

$$x_1 = \text{mean_area}$$

$$x_2 = \text{mean_concave_points}$$
- variabile da predire **discreta**

$$y = \begin{cases} -1 & \text{if } \text{benign} \\ +1 & \text{if } \text{malign} \end{cases}$$
- separazione lineare delle due classi
 - la retta di separazione è $b + w_1 x_1 + w_2 x_2 = 0$ con w_1, w_2, b da apprendere
 - le coppie $\mathbf{x} = (x_1, x_2)$ tali che $b + w_1 x_1 + w_2 x_2 < 0$ sono cellule benigne
 - quelle tali che $w_1 x_1 + w_2 x_2 - b \geq 0$ sono maligne, ossia:



$$y = \begin{cases} -1 & \text{if } b + \mathbf{w} \cdot \mathbf{x} < 0 \\ +1 & \text{if } b + \mathbf{w} \cdot \mathbf{x} \geq 0 \end{cases}$$

12



Classificazione di Istanze con Iperpiano

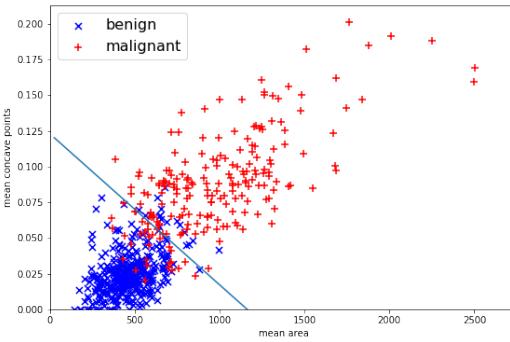
- variabili di input

$$x_1 = \text{mean_area} \quad x_2 = \text{mean_concave_points}$$

$$y = \begin{cases} -1 & \text{if } b + w \cdot x < 0 \\ +1 & \text{if } b + w \cdot x \geq 0 \end{cases}$$

- Iperpiano individuato (i.e. retta)

- $0.007\text{mean_area} +$
 $67.443\text{mean_concave_points}$
 $-8.287 = 0$



- Classifichiamo due nuove cellule

- $\mathbf{x}^{(1)} = (\text{mean_area} = 500, \text{mean_concave_points} = 0.025)$
 $0.007 \cdot 500 + 67.443 \cdot 0.025 - 8.287 = -3.035$ cellula benigna
- $\mathbf{x}^{(2)} = (\text{mean_area} = 500, \text{mean_concave_points} = 0.075)$
 $0.007 \cdot 500 + 67.443 \cdot 0.075 - 8.287 = 0.271$ cellula maligna (border line)



Come Trovare l’Iperpiano di Separazione ?

- ogni istanza è etichettata con -1 o +1 $y = \begin{cases} -1 & \text{if } \text{benign} \\ +1 & \text{if } \text{malign} \end{cases}$
- forma compatta equivalente di separazione $-y(b + w \cdot x) < 0$
 - l’istanza correttamente classificata da valore negativo, altrimenti positivo
 - e.g. con $y = -1$ e $b + w \cdot x = -1$ si ha $1 \cdot -1 = -1$ idem con $y = 1$
- $-y(b + w \cdot x) < 0$ è equivalente a $\max(0, -y(b + w \cdot x)) = 0$
 - l’espressione $\max(0, -y(b + w \cdot x))$ restituisce 0 se l’istanza è classificata correttamente, altrimenti il valore è positivo
- perciò minimizzando, rispetto a b , w , la somma di questa espressione sulle m istanze di training, si minimizza l’errore

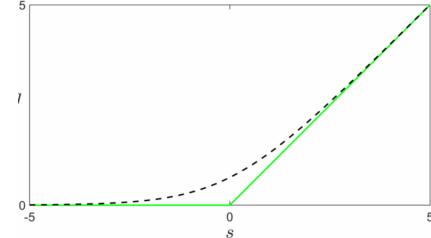
$$\underset{b, w}{\text{minimize}} \sum_{i=1}^m \max(0, -y_i \cdot h_w(x_i)) \text{ dove } h_w(x_i) = b + w \cdot x_i$$

- la funzione è continua e convessa ma non derivabile, perciò il metodo di discesa del gradiente è inapplicabile, inoltre ha un min fittizio per $b=w=0$



Iperpiano di Separazione: Logistic Loss

- sostituiamo $\max(0, s)$ *in verde* con una funzione derivabile
- una sua approssimazione è nota come $\text{softmax}(0, s) = \log(1+e^s)$
 - convessa, continua e derivabile
- perciò minimizzando la somma rispetto a b e w della softmax su tutte le istanze, si minimizza l'errore



$$\underset{b,w}{\text{minimize}} \sum_{i=1}^m \log\left(1+e^{-y_i \cdot h_w(x_i)}\right)$$

- Questa formulazione è nota come *logistic loss* a cui si aggiunge la regolarizzazione dei parametri w con peso λ

$$\underset{b,w}{\text{minimize}} \sum_{i=1}^m \log\left(1+e^{-y_i \cdot h_w(x_i)}\right) + \lambda \|w\|_2^2$$

Gianluca Moro - DISI, Università di Bologna

15



Regressione Logistica (i)

- la derivata della logistic loss in forma compatta (i.e. senza b)

$$\tilde{x} = (1, x) \quad \tilde{w} = (b, w) \quad \nabla \sum_{i=1}^m \log\left(1+e^{-y_i \cdot h_{\tilde{w}}(\tilde{x}_i)}\right) + \lambda \|w\|_2^2 = -\sum_{i=1}^m \frac{y_i \tilde{x}_i}{1+e^{y_i \cdot h_{\tilde{w}}(\tilde{x}_i)}} + \lambda w = 0$$

- determinazione dei parametri w migliori con discesa del gradiente

$$w = w - \eta \sum_{i=1}^m -\frac{y_i \tilde{x}_i}{1+e^{y_i \cdot h_{\tilde{w}}(\tilde{x}_i)}} - 2\lambda w = -\sum_{i=1}^m \sigma(-y_i h_{\tilde{w}}(\tilde{x}_i)) y_i \tilde{x}_i - 2\lambda w$$

$$\text{dove } \sigma(t) = \frac{1}{1+e^{-t}}$$

- $\sigma(t)$ è nota come *Regressione Logistica* ed il classificatore risultante è il seguente (versione moderna del Perceptron)

$$\sigma(x) = \frac{e^{h_w(x)}}{1+e^{h_w(x)}} = \frac{1}{1+e^{-(b+w \cdot x)}} \quad \text{dove } b + w \cdot x \text{ è l'iperpiano individuato}$$



Regressione Logistica: Caratteristiche

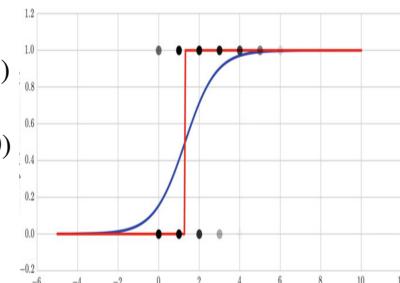
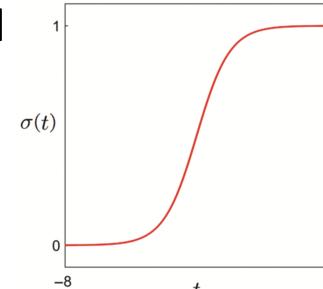
- $\sigma(t) = \frac{1}{1+e^{-(b+w\cdot t)}}$ Dominio e Codominio: $R \rightarrow [0, 1]$
- il risultato è interpretabile come probabilità di appartenenza di ogni istanza t ad una delle classi
- approssima una funzione a gradino

se x è sull'iperpiano allora $w \cdot x + b = 0 \rightarrow \sigma(x) = \frac{1}{1+e^0} = 0.5$

se x è t.c. $w \cdot x + b > 0 \rightarrow \sigma(x) = \frac{1}{1+e^{-(b+w \cdot x > 0)}} > 0.5$ (Classe 1)

se x è t.c. $w \cdot x + b < 0 \rightarrow \sigma(x) = \frac{1}{1+e^{-(b+w \cdot x < 0)}} < 0.5$ (Classe 0)

- tutto ciò vale sia con più variabili di input dove $x = (x_1, \dots, x_n)$, sia con più di 2 classi



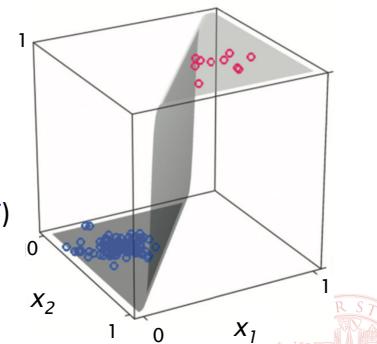
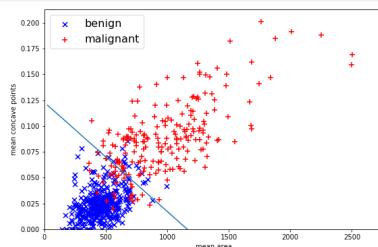
17



Gianluca Moro - DISI, Università di Bologna

Regressione Logistica: Esempio di Classificazione

- Classificazione di cellule in benigne e maligne, Iperpiano individuato
 - $0.007 \cdot \text{mean_area} + 67.443 \cdot \text{mean_concave_points} - 8.287 = 0$
- Classifichiamo due cellule
 - $\mathbf{x}^{(1)} = (\text{mean_area} = 500, \text{mean_concave_points} = 0.025)$
 - $0.007 \cdot 500 + 67.443 \cdot 0.025 - 8.287 = -3.035$ (benigna)
 - $\sigma(500, 0.025) = \frac{1}{1+e^{-(0.007 \cdot 500 + 67.443 \cdot 0.025 - 8.287)}} = 0.046$
 - $\mathbf{x}^{(2)} = (\text{mean_area} = 500, \text{mean_concave_points} = 0.075)$
 - $0.007 \cdot 500 + 67.443 \cdot 0.075 - 8.287 = 0.337$ (maligna)
 - $\sigma(500, 0.075) = \frac{1}{1+e^{-(0.007 \cdot 500 + 67.443 \cdot 0.075 - 8.287)}} = 0.567$



18



Gianluca Moro - DISI, Università di Bologna

Valutazione del modello di Classificazione

Matrice di Confusione

		classe predetta	
		a	b
classe reale	a	veri positivi TP	falsi negativi FN
	b	falsi positivi FP	vero negativi TN

Precision e Recall
ancora più
importanti quando
le classi sono
sbilanciate

$$\begin{array}{ll} 1 \text{ precision}(a) = \text{TP}/(\text{TP}+\text{FP}) & \text{precision}(b) = \text{TN}/(\text{TN}+\text{FN}) \\ 2 \text{ recall}(a) = \text{TP}/(\text{TP}+\text{FN}) & \text{recall}(b) = \text{TN}/(\text{TN}+\text{FP}) \end{array}$$

$$3 \text{ accuracy} = \text{TP}+\text{TN}/(\text{TP}+\text{FN}+\text{TN}+\text{FP})$$

$$\begin{array}{l} 4 \text{ F1-measure}(a) = 2*\text{precision}(a)*\text{recall}(a)/(\text{precision}(a)+\text{recall}(a)) \\ \text{F1-measure}(b) = 2*\text{precision}(b)*\text{recall}(b)/(\text{precision}(b)+\text{recall}(b)) \end{array}$$

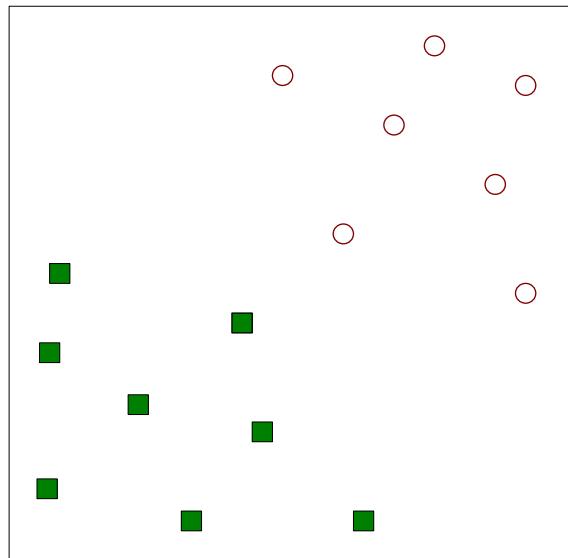
19

Gianluca Moro - DISI, Università di Bologna



Ritorniamo alla classificazione con Iperpiano in due dimensioni

- Quali rette (i.e. funzioni) separano i dati delle 2 classi ?
- in più dimensioni abbiamo iperpiani di separazione

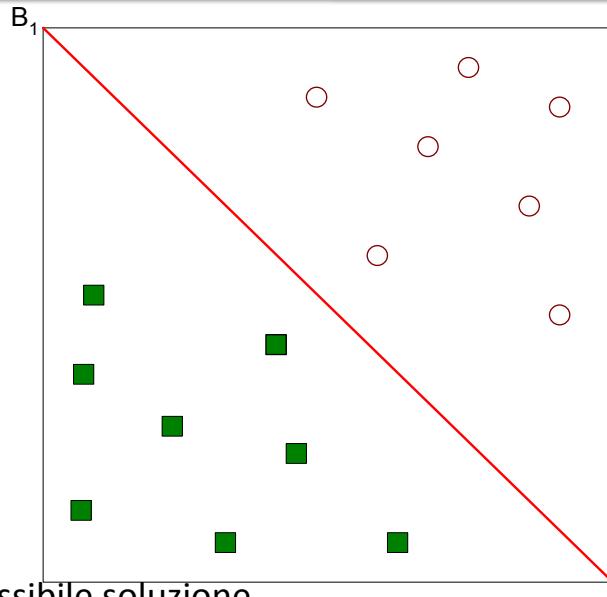


Gianluca Moro - DISI, Università di Bologna

21



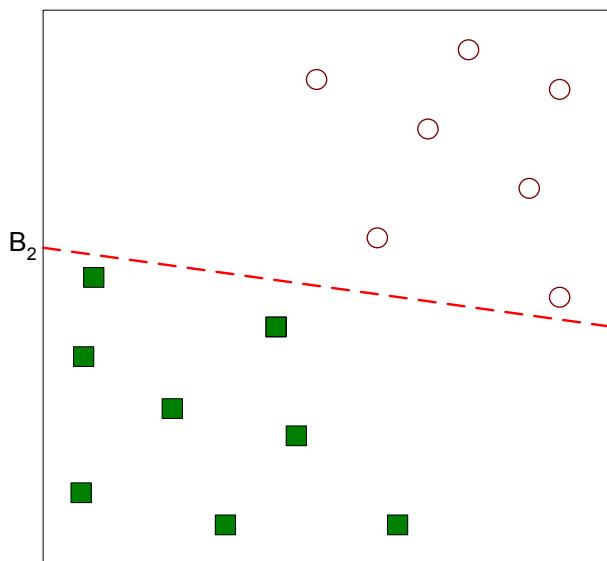
Questa ?



- Una possibile soluzione



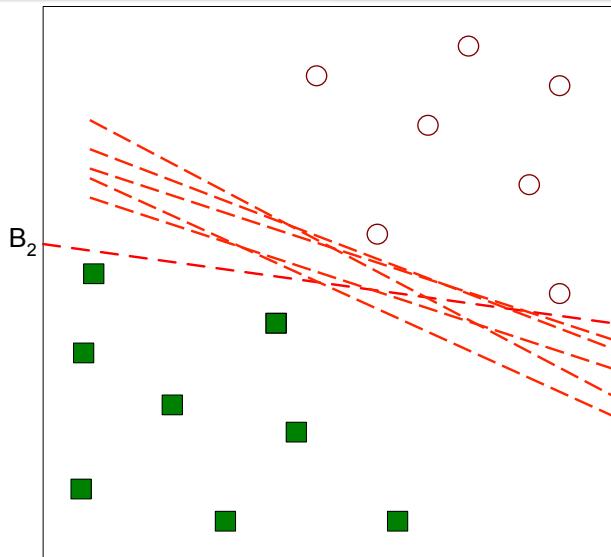
Oppure Questa ?



- Un'altra possibile soluzione



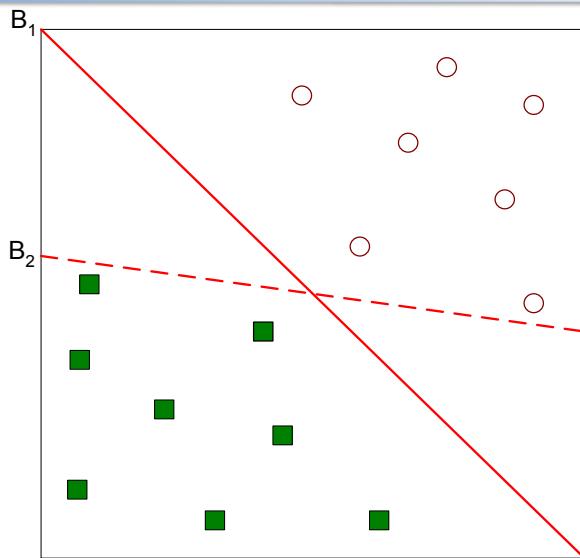
Oppure una tra queste ?



- Altre possibili soluzioni



Come stabilire quale sia la migliore ?



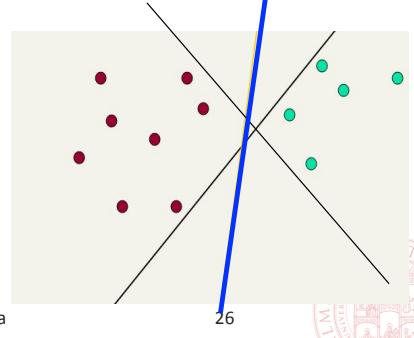
- Quale è la migliore, B1 o B2 ? Come si definisce il concetto di *migliore* ?



Classificatori lineari: Quale Iperpiano ?

- Le possibili soluzioni in 2D sono rette di separazione che soddisfano l'equazione $w_1x_1 + w_2x_2 + b = 0$ per w_1, w_2, b
- Diversi metodi trovano iperpiani di separazione in generale non ottimali
 - Un esempio è il Perceptron alla base delle prime reti neurali
- Idea: trovare una soluzione tale che
 - sia massimizzata la distanza tra l'iperpiano ed i "punti difficili", ossia più vicini al decision boundary

la retta è un decision boundary tra le 2 classi:
 $w_1x_1 + w_2x_2 + b = 0$

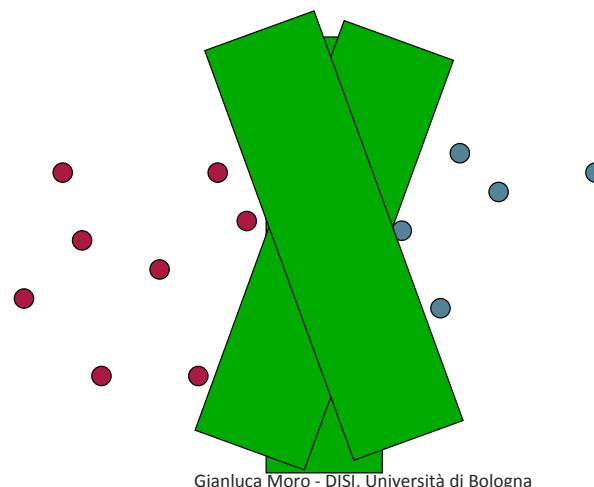


Gianluca Moro - DISI, Università di Bologna



Sviluppiamo l'Idea

- Il numero di soluzioni diminuisce se cerchiamo una separazione lineare con il maggiore margine possibile tra le istanze delle due classi



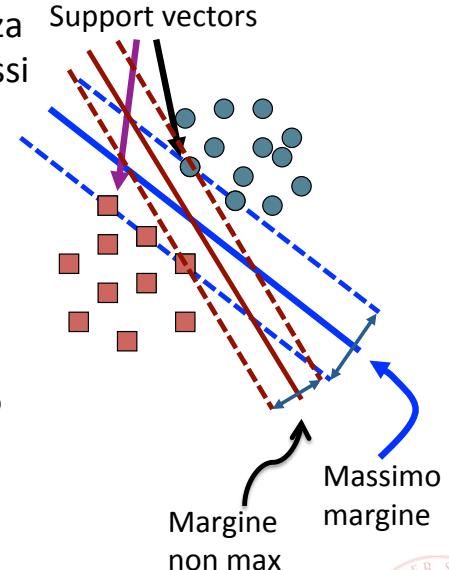
Gianluca Moro - DISI, Università di Bologna

27



Iperpiano di Separazione Migliore

- individuare l'iperpiano che massimizza il margine tra le istanze delle due classi → **minore overfitting**
- L'iperpiano migliore è definito dai punti “difficili” chiamati *support vectors*
 - punti più vicini al decision boundary
 - Se mancassero i restanti punti (i.e. tutti i non support vector) l'iperpiano calcolato sarebbe il medesimo
- Risolubile come problema di ottimizzazione quadratica → **SUPPORT VECTOR MACHINES**



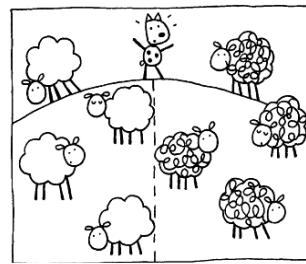
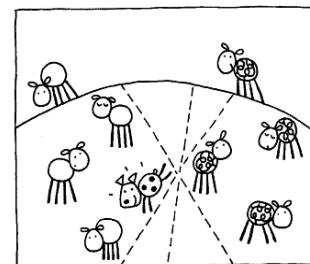
Gianluca Moro - DISI, Università di Bologna

28



SVM: Support Vector Machines

- Metodo sviluppato da V. Vapnik (e suoi co-autori) negli anni 70 in Russia e diventato noto internazionalmente solo nel 1992
- Definito originariamente per la classificazione binaria
 - i.e. istanze appartenenti in modo esclusivo a due classi
- Obiettivo: individuare la separazione lineare ottimale (secondo un criterio geometrico) tra le istanze delle due classi
 - Adatto a domini ad elevata dimensionalità
 - Ritenuto tra i metodi più efficaci per il testo
 - più efficace di altri metodi con training set piccoli



Gianluca Moro - DISI, Università di Bologna

29



Definizione del Problema

- Sia $D=\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_r, y_m)\}$ l'insieme delle r istanze di **training**
- dove $\mathbf{x}_i = (x_1, \dots, x_n)$ è il **vettore dell'istanza i-esima** nello spazio di valori reali $X \subseteq R^n$
- y_i è la **classe** di appartenenza di \mathbf{x}_i , con $y_i \in \{1, -1\}$
1: *classe esempi positivi* -1: *classe esempi negativi*
- Individuare \mathbf{w} e b t.c. l'iperpiano $\mathbf{w}^T \cdot \mathbf{x} + b = 0$ massimizzi la separazione tra i support vector delle 2 classi
- Il classificatore risultante è una funzione lineare

$$f(\mathbf{x}_i) = y_i \text{ dove } y_i = \begin{cases} 1 & \text{se } \mathbf{w} \cdot \mathbf{x}_i + b \geq 0 \\ -1 & \text{se } \mathbf{w} \cdot \mathbf{x}_i + b < 0 \end{cases}$$

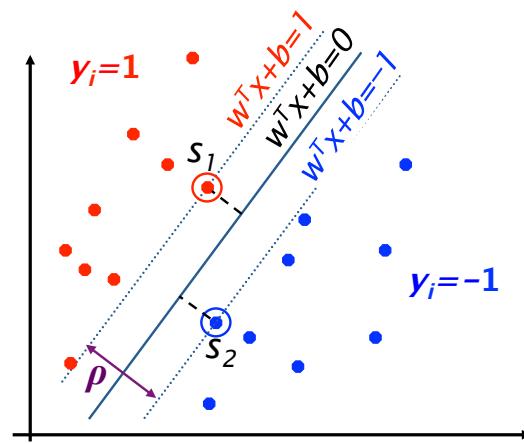
Gianluca Moro - DISI, Università di Bologna

40



Separazione Lineare

- i punti x_i sull'iperpiano di separazione soddisfano l'uguaglianza $\mathbf{w}^T \cdot \mathbf{x}_i + b = 0$
 - i punti \mathbf{x}_i t.c. $\mathbf{w}^T \cdot \mathbf{x}_i + b > 1$ sono le istanze con $y_i = 1$ e
 - i punti \mathbf{x}_i t.c. $\mathbf{w}^T \cdot \mathbf{x}_i + b < -1$ sono le istanze con $y_i = -1$
 - b è proporzionale alla distanza dello iperpiano dall'origine
- siano s_1 e s_2 due support vector delle 2 classi (punti più vicini all'iperpiano)
- definiamo i 2 iperpiani paralleli a $\mathbf{w}^T \cdot \mathbf{x} + b = 0$ passanti per s_j
- $\mathbf{w}^T \cdot s_1 + b = 1 \quad \mathbf{w}^T \cdot s_2 + b = -1$, ρ è il margine (distanza) tra loro
- Obiettivo: calcolare s_j, \mathbf{w}, b che massimizzino ρ



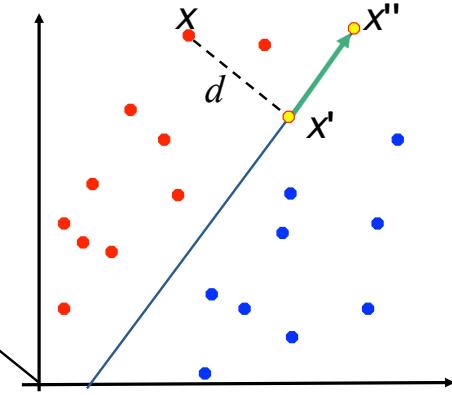
Gianluca Moro - DISI, Università di Bologna

31



Distanza di un Punto da un Iperpiano

- La distanza euclidea d del punto x dall'iperpiano di separazione è $d = \|x - x'\|$ norma-2 del vettore
- se 2 vettori sono *ortogonali*, il loro prodotto scalare è 0
- siano x' e x'' 2 punti sull'iperpiano di separazione $\rightarrow w^T \cdot x' + b = 0$ e $w^T \cdot x'' + b = 0$
- il vettore differenza è $w^T x' - w^T x'' = 0$ ed è parallelo all'iperpiano di separazione ma $w^T \cdot (x' - x'') = 0 \rightarrow$ i 2 vettori sono ortogonali $\rightarrow w$ è ortogonale all'iperpiano
- $w/\|w\|$ è il vettore unitario, perciò $d w/\|w\| = x - x' \rightarrow x' = x - d \cdot w/\|w\|$ ma x' soddisfa $w \cdot x' + b = 0 \rightarrow w^T \cdot (x - d w/\|w\|) + b = 0 \rightarrow w^T \cdot x - d w^T w/\|w\| + b = 0$ e poiché $\|w\| = \sqrt{w^T w}$ allora
- $\rightarrow w^T \cdot x - d \|w\| + b = 0 \rightarrow d = (w^T x + b)/\|w\|$



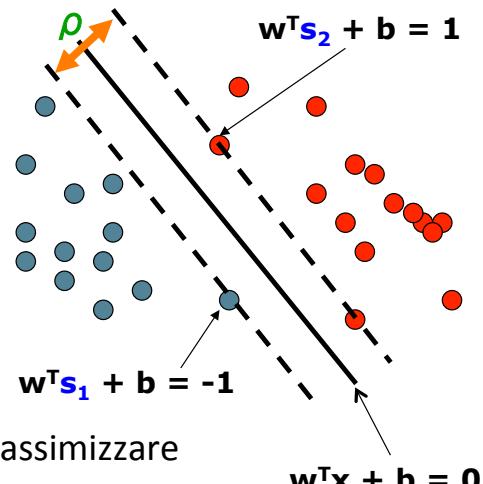
Gianluca Moro - DISI, Università di Bologna

32



Calcolo analitico del margine ρ

- iperpiano di separazione**
 $w^T x + b = 0$
 - iperpiani paralleli**
 $\min_{i=1,\dots,n} |w^T x_i + b| = 1$
 - gli x_i che soddisfano l'equazione sono i support vector
 - siano s_1, s_2 due support vector
 - la somma della loro distanza dall'iperpiano è il margine ρ da massimizzare
- $$\frac{|w^T s_1 + b|}{\|w\|} + \frac{|w^T s_2 + b|}{\|w\|} = \frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|} = \rho$$



Gianluca Moro - DISI, Università di Bologna

33

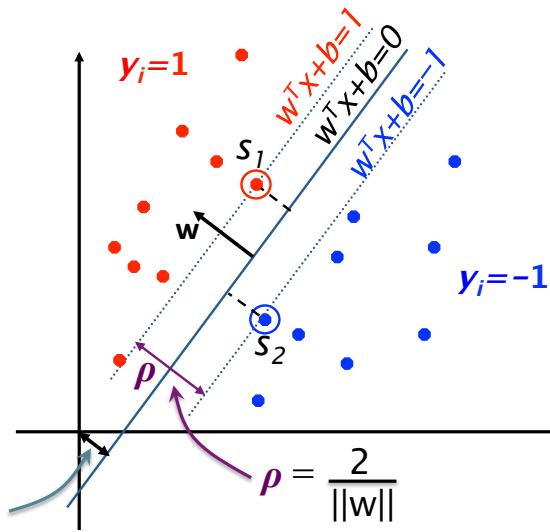


Iperpiano di Separazione: la variabile b

- iperpiano di separazione
 $w^T x + b = 0$
- quanto dista dall'origine ?
- la distanza di un punto x_i dall'iperpiano è

$$\frac{w^T x_i + b}{\|w\|}$$

- Le componenti del punto nell'origine valgono 0
- $\rightarrow w^T x_i = 0$, perciò dista $\frac{b}{\|w\|}$
- b determina la distanza dell'iperpiano dall'origine



Gianluca Moro - DISI, Università di Bologna

34



SVM Lineare: il problema di ottimizzazione

Calcolare w e b tali che sia massimizzato il margine

$$\rho = \frac{2}{\|w\|} \quad \text{con i vincoli, per ogni } \{(x_i, y_i)\}$$

$$\begin{aligned} w^T x_i + b &\geq 1 \quad \text{se } y_i = 1 \\ w^T x_i + b &\leq -1 \quad \text{se } y_i = -1 \end{aligned} \quad \rightarrow \text{in forma compatta} \quad y_i (w^T x_i + b) \geq 1$$

- Formulazione equivalente: $\max \frac{2}{\|w\|} = \min \frac{\|w\|}{2} = \min \sqrt{\sum_{j=1}^n w_j^2} \rightarrow \min \frac{w^T w}{2}$

Calcolare w e b tali che sia minimizzata

$$\Phi(w) = \frac{w^T w}{2} \quad \text{con i vincoli, per ogni } \{(x_i, y_i)\}, \quad y_i (w^T x_i + b) \geq 1$$

minimizzare $\|w\|^2$
è matematicamente
più semplice di
 $\|w\|$

Gianluca Moro - DISI, Università di Bologna

35



Ottimizzazione quadratica:

Metodo di Lagrange

Calcolare \mathbf{w} e b tali che sia minimizzata

$$\Phi(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{w}}{2} \quad \text{con i vincoli, per ogni } \{(\mathbf{x}_i, y_i)\}, \\ y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

- Ottimizzazione di una funzione obiettivo quadratica con vincoli lineari -> **problema di ottimizzazione convesso**
- Metodo di **Lagrange**: Ridefinizione della **funzione obiettivo** incorporando **ogni vincolo nella funzione obiettivo con associato un moltiplicatore $\alpha_i \geq 0$**

Calcolare $\alpha_1 \dots \alpha_N$ tali che sia minimizzata la funzione

$$L_p = \frac{\mathbf{w}^T \mathbf{w}}{2} - \sum_{i=1}^r \alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1]$$

con quali vincoli ?



Metodo di Lagrange: Definizione dei vincoli

- Minimizzare L_p definita con moltiplicatori di **Lagrange α_i** ,

$$L_p = \frac{\mathbf{w}^T \mathbf{w}}{2} - \sum_{i=1}^r \alpha_i [y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1] \quad \text{formulazione primaria}$$

- ogni vincolo $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1$ è incorporato come $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0$
- La minimizzazione penalizza implicitamente le soluzioni per \mathbf{w} e b che violano i vincoli perché queste fanno aumentare L_p
 - e.g. poiché ogni $\alpha_i \geq 0$, se ogni $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 < 0$ allora la sommatoria è negativa e il segno meno la cambia in positivo sommandola a $\mathbf{w}^T \mathbf{w}/2$
- Per calcolare il minimo, poniamo le derivate prime di $L_p = 0$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i \quad \frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^r \alpha_i y_i = 0$$

- Se i vincoli incorporati in L_p fossero equazioni, invece che disequazioni, risolvendole insieme alle derivate ricaveremmo **α_i, \mathbf{w}, b** e la soluzione



Generalizzazione del Metodo di Lagrange: Condizioni di Karush-Kuhn-Tucker

- Per risolvere un problema di ottimizzazione non lineare con **vincoli di disuguaglianza** è necessario, ma non sufficiente, soddisfare le seguenti condizioni:

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^r \alpha_i y_i x_i \quad \frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^r \alpha_i y_i = 0$$

derivate
prime di L_p

$$\forall i = 1, \dots, r \\ \alpha_i \geq 0 \quad y_i(w^T x_i + b) - 1 \geq 0 \quad \alpha_i(y_i(w^T x_i + b) - 1) = 0$$

- moltiplicatori $\alpha \geq 0$
- vincoli originali
- relative equazioni

- i moltiplicatori α_i sono nulli per tutti i punti x_i tali che $y_i(w^T x_i + b) - 1 \neq 0$
 - infatti $\alpha_i(y_i(w^T x_i + b) - 1) = 0$ perciò $\alpha_i = 0$
- I restanti x_i t.c. $y_i(w^T x_i + b) - 1 = 0$ sono **support vector** con $\alpha_i > 0$
 - se fossero tutti $\alpha_i = 0$ allora anche w sarebbe 0 in base a $\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^r \alpha_i y_i x_i$

Gianluca Moro - DISI, Università di Bologna

38



Formulazione del Problema Duale (i)

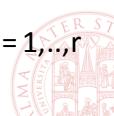
- Le condizioni di Karush-Kuhn-Tucker sono invece necessarie e sufficienti con **funzioni obiettivo convesse** e **vincoli lineari**
- Tuttavia il problema di ottimizzazione rimane ancora complicato a causa della presenza dei vincoli di disuguaglianza
- Se eliminassimo le variabili w e b potremmo eliminare anche i relativi vincoli di disuguaglianza
- Per questi problemi esiste una formulazione duale: sostituendo le derivate prime nella **formulazione primaria**

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^r \alpha_i y_i x_i \quad \frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^r \alpha_i y_i = 0 \quad L_p = \frac{w^T w}{2} - \sum_{i=1}^r \alpha_i [y_i(w^T x_i + b) - 1]$$

- si ottiene $L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j x_i^T x_j$ con vincoli $\sum_{i=1}^r \alpha_i y_i = 0$
 $\alpha_i \geq 0 \quad \forall i = 1, \dots, r$

Gianluca Moro - DISI, Università di Bologna

39



Formulazione del Problema Duale (ii)

- *Formulazione duale*

$$\underset{\alpha}{\text{Max}} : L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j x_i^T x_j$$

con i vincoli

$$\begin{aligned} \sum_{i=1}^r \alpha_i y_i &= 0 \\ \alpha_i &\geq 0 \quad \forall i = 1, \dots, r \end{aligned}$$

- Formulazione nota come **Wolfe dual**

- Per problemi convessi con vincoli lineari, ha la proprietà che il massimo di L_D si ha con gli stessi valori di w , b e α_i che producono il minimo nella *formulazione primaria* L_P
- La soluzione si basa su tecniche numeriche nell'ambito della programmazione quadratica che esulano dagli scopi del corso

- Risolvendo L_D si ottengono i valori α_i da cui si ricavano w e b :

- w si ottiene sostituendo α_i nella derivata prima di L_P
- b si ottiene sostituendo w in $\alpha_i(y_i(w^T x_i + b) - 1) = 0 \rightarrow \alpha_i(y_i w^T x_i + y_i b) = \alpha_i$
 $b = 1/y_k - w^T x_k$ per ogni x_k tale che $\alpha_k > 0$ i.e. per ogni **support vector**

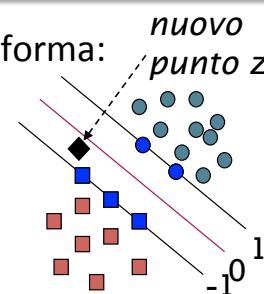


SVM: La Funzione di Classificazione Finale

- La funzione di classificazione perciò ha questa forma:

$$f(z) = \text{sign}(w^T z + b) \quad \text{dove}$$

$$w^T z = \sum_{k \in \text{support vector}} y_k \alpha_k x_k^T z \quad b = \frac{1}{y_k} - w^T x_k = y_k - w^T x_k$$

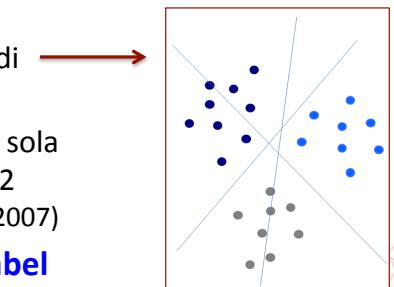
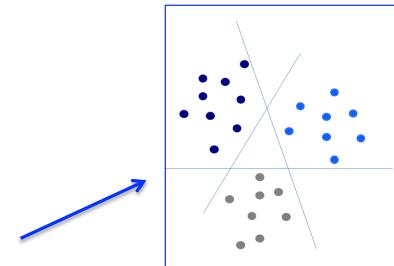


- data un'istanza z , $f(z)$ esegue il prodotto scalare tra z e i support vector $x_k \Rightarrow$ non occorre l'iperpiano di separazione
- se $f(z)$ ha segno positivo allora z è classificata come positiva, altrimenti negativa
- secondo la formulazione duale, il costo per determinare la funzione di classificazione include quello del prodotto scalare di ogni coppia delle istanze di training



SVM: Classificazione Multi-Classe

- Metodo di classificazione binaria
 - i.e. esistono 2 classi ed ogni istanza appartiene ad una e una sola classe
- Tre modalità per applicare SVM a classificazioni con K classi:
 - **1-molti:** K data set a 2 classi, ciascuno contiene a turno una classe e l'unione delle restanti classi → generazione di K SVM
 - **1-1:** si genera una SVM per ogni coppia di classi → $K(K-1)/2$ data set e SVM
 - riformulazione del metodo SVM con una sola funzione obiettivo invece che K o $K(K-1)/2$ (Crammer 2002, Lee-Wabha 2004, Nanculef 2007)
- **1-molti** adatta anche a problemi **multi-label**



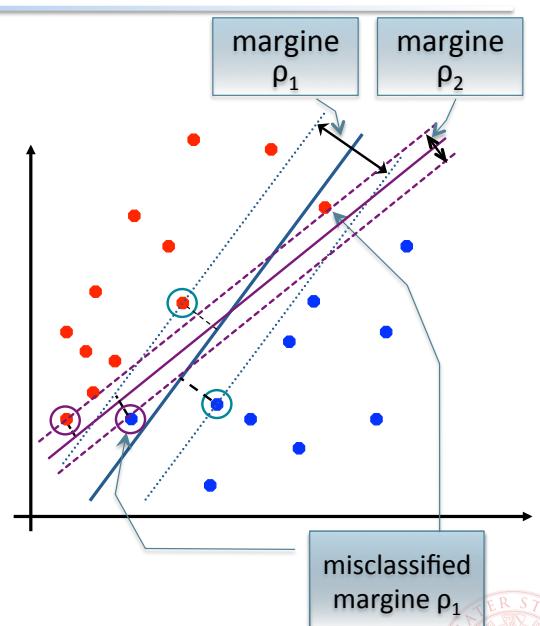
Gianluca Moro - DISI, Università di Bologna

42



Margine Soft e Istanze Misclassified

- maggiore è il margine, minore è l'overfitting
 - accuratezza più affidabile su nuove istanze
- margine ρ_2 senza errori
- margine $\rho_1 > \rho_2$ ma con errori
- **qual è l'iperpiano migliore ?**
- trade off tra **massimizzazione del margine** e **minimizzazione del numero di errori**
→ MARGINE SOFT
- Introduzione della tolleranza agli errori nel training set ma con penalizzazione



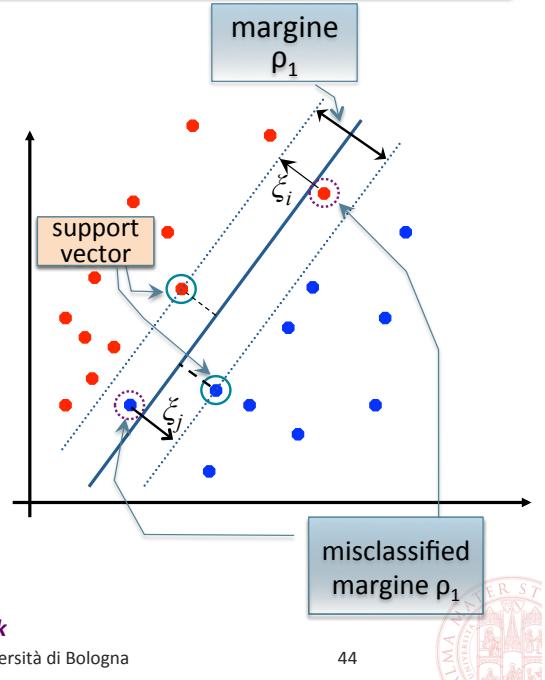
Gianluca Moro - DISI, Università di Bologna

43



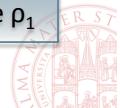
Istanze Misclassified e Variabili Slack

- La formulazione illustrata non ammette errori nel training set
- si introducono altre variabili ξ , chiamate *slack* per “spostare” le istanze misclassified
- Le variabili ξ “trasformano” gli errori, i.e. i punti, all’interno del margine in support vector
- massimizzare solo il margine con queste trasformazioni porta a soluzioni ottime degeneri
 - il margine è così ampio che contiene tutte le istanze di training
- minimizzare anche la somma dei ξ_k**



Gianluca Moro - DISI, Università di Bologna

44



Margine Soft: Ottimizzazione con variabili slack

- Formulazione precedente:

Calcolare \mathbf{w} e b tali che sia minimizzata

$$\Phi(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{w}}{2} \quad \text{con i vincoli, per ogni } \{(\mathbf{x}_i, y_i)\}, \\ y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

La soluzione ha questa forma: $f(\mathbf{x}) = \text{sign}(\sum y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b)$

- Formulazione con variabili slack:

Calcolare \mathbf{w} e b tali che sia minimizzata

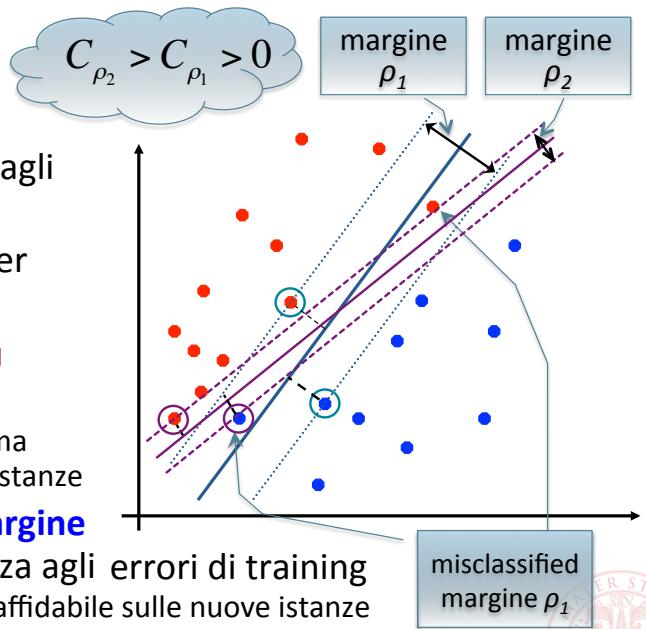
$$\Phi(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_i \xi_i \quad \text{con i vincoli, per ogni } \{(\mathbf{x}_i, y_i)\}, \\ y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ \text{con } \xi_i \geq 0 \text{ per ogni } i$$

- Il parametro $C > 0$ consente di pesare/controllare l’overfitting



Margine Soft e il Parametro C

- $\min \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_i \xi_i$
- aumentando il valore di C ,** aumenta il peso assegnato agli errori di training
- **il margine si restringe** per ridurre gli errori di training
- più il margine è piccolo, più aumenta l'overfitting**
 - più accuratezza sul training ma meno affidabile sulle nuove istanze
- riducendo C aumenta il margine** perché aumenta la tolleranza agli errori di training
→ accuratezza inferiore ma più affidabile sulle nuove istanze



Gianluca Moro - DISI, Università di Bologna

46



Metodo di Lagrange con Variabili Slack

- Minimizzare L_p definita con moltiplicatori di Lagrange α_i, μ_i

$$L_p = \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_{i=1}^r \xi_i - \sum_{i=1}^r \alpha_i [y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^r \mu_i \xi_i \quad \text{formulazione primaria con variabili Slack}$$

- un moltiplicatore $\alpha_i \geq 0$ per ogni vincolo $y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i$, incorporato come $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i \geq 0$, ed uno $\mu_i \geq 0$ per ogni ξ_i
- La minimizzazione penalizza implicitamente le soluzioni per \mathbf{w} e b che violano i vincoli perché queste fanno aumentare L_p
 - e.g. se ogni $y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i < 0$, la sommatoria, con ogni $\alpha_i \geq 0$, diventa negativa ed il risultato con segno meno si somma a $\mathbf{w}^T \mathbf{w}/2$
- Per calcolare il minimo, come nel caso senza variabili slack, poniamo le derivate prime di $L_p = 0$

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^r \alpha_i y_i \mathbf{x}_i \quad \frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^r \alpha_i y_i = 0 \quad \frac{\partial L_p}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \Rightarrow C = \alpha_i + \mu_i$$

Gianluca Moro - DISI, Università di Bologna

48



Variabili Slack:

Condizioni di Karush-Kuhn-Tucker (KKT)

- Come nel caso senza variabili slack, le seguenti condizioni sono necessarie e sufficienti per risolvere il problema:

derivate prime di L_p

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^r \alpha_i y_i x_i \quad \frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^r \alpha_i y_i = 0 \quad \frac{\partial L_p}{\partial \xi_i} = 0 \Rightarrow C = \alpha_i + \mu_i$$

$$\begin{aligned} \forall i=1,\dots,r \quad \alpha_i &\geq 0, \quad \mu_i \geq 0, \quad \xi_i \geq 0, \quad y_i(w^T x_i + b) - 1 + \xi_i \geq 0 \\ \alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) &= 0, \quad \mu_i \xi_i = 0 \quad C > 0 \end{aligned}$$

- moltiplicatori $\alpha, \mu \geq 0$
- vincoli originali
- relative equazioni

- $\forall x_i$ t.c. $y_i(w^T x_i + b) - 1 + \xi_i \neq 0 \rightarrow$ moltiplicatore $\alpha_i = 0$
 - I restanti x_i t.c. $y_i(w^T x_i + b) - 1 + \xi_i = 0$ sono **support vector** con $\alpha_i > 0$
 - anche i punti misclassified interni al margine, i.e. $\xi_i > 0$, hanno $\alpha_i > 0$, infatti, poiché $\mu_i \xi_i = 0$ allora $\mu_i = 0$, perciò $\alpha_i = C > 0$



Variabili Slack: Problema Duale (i)

- Come nel caso senza variabili slack, sostituiamo in L_p

$$L_p = \frac{w^T w}{2} + C \sum_{i=1}^r \xi_i - \sum_{i=1}^r \alpha_i [y_i(w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^r \mu_i \xi_i$$

- le relative derivate prime

$$\frac{\partial L_p}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^r \alpha_i y_i x_i \quad \frac{\partial L_p}{\partial b} = 0 \Rightarrow \sum_{i=1}^r \alpha_i y_i = 0 \quad \frac{\partial L_p}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i = \mu_i$$

- si ottiene (con $\alpha_i \geq 0 \quad \forall i=1,\dots,r$)

$$\begin{aligned} L_D &= \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j x_i^T x_j + C \sum_{i=1}^r \xi_i - \sum_{i=1}^r \alpha_i [y_i(\sum_{j=1}^r \alpha_j y_j x_i^T x_j + b) - 1 + \xi_i] - \sum_{i=1}^r (C - \alpha_i) \xi_i \\ &= \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j x_i^T x_j - \left(\sum_{i=1}^r \sum_{j=1}^r \alpha_i y_i \alpha_j y_j x_i^T x_j + b \sum_{i=1}^r \alpha_i y_i - \sum_{i=1}^r \alpha_i \right) + C \sum_{i=1}^r \xi_i - \left(C \sum_{i=1}^r \xi_i - \sum_{i=1}^r \alpha_i \xi_i \right) - \sum_{i=1}^r \alpha_i \xi_i \\ &= \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{i=1}^r \sum_{j=1}^r y_i y_j \alpha_i \alpha_j x_i^T x_j \end{aligned}$$

la formulazione duale non contiene le variabili w, b e $\xi \rightarrow$ possiamo ricavare gli α



Variabili Slack: Problema Duale (ii)

- *Formulazione duale*

$$\underset{\alpha}{\text{Max}} : L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j x_i^T x_j$$

con i vincoli

$$\begin{aligned} \sum_{i=1}^r \alpha_i y_i &= 0 \\ \alpha_i &\geq 0 \quad \forall i = 1, \dots, r \end{aligned}$$

- E' identica alla formulazione precedente, eccetto l'intervallo di variabilità dei moltiplicatori α_i

- infatti $\frac{\partial L_p}{\partial \xi_i} = 0 \Rightarrow C - \alpha_i = \mu_i$ ma $\alpha_i, \mu_i \geq 0 \rightarrow 0 \leq \alpha_i \leq C$

- Risolvendo L_D si ottengono i valori α_i da cui si ricavano w e b :

- w : si ottiene sostituendo α_i nella derivata di L_p posta a zero $w = \sum_{i=1}^r \alpha_i y_i x_i$
- b : si ottiene sostituendo w in $y_i(w^T x_i + b) - 1 + \xi_i = 0 \rightarrow b = (1 - \xi_i) / (y_i - w^T x_i)$
- ma ξ_i è ignoto, tuttavia $\forall 0 < \alpha_k < C, \mu_k > 0$ perché $C - \alpha_k = \mu_k \rightarrow \xi_k = 0$ perché dalla condizione di KKT $\mu_k \xi_k = 0 \rightarrow$ ogni x_k è un support vector
- perciò $y_k(w^T x_k + b) - 1 + \xi_k = 0 = y_k(w^T x_k + b) - 1 \rightarrow b = 1 / (y_k - w^T x_k)$

Gianluca Moro - DISI, Università di Bologna

51

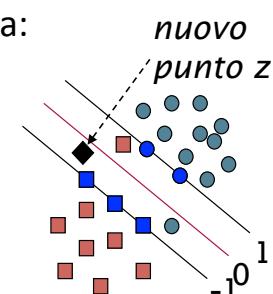


SVM con Variabili Slack: La Funzione di Classificazione Finale

- La funzione di classificazione ha la stessa forma:

$$f(z) = \text{sign}(w^T z + b) \quad \text{dove}$$

$$w^T z = \sum_{k \in \text{support vector}} y_k \alpha_k x_k^T z \quad b = \frac{1}{y_k} - w^T x_k = y_k - w^T x_k$$



- i.e. la classificazione avviene come nella formulazione senza variabili slack

- data un'istanza z , $f(z)$ esegue il prodotto scalare tra z e i support vector $x_k \Rightarrow$ non occorre l'iperpiano di separazione
- se $f(z)$ ha segno positivo allora z è classificata come positiva, altrimenti negativa \rightarrow la classificazione lineare con margine soft è efficiente quanto la classificazione senza variabili slack

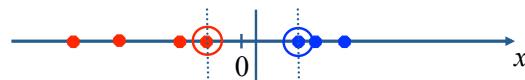
Gianluca Moro - DISI, Università di Bologna

52

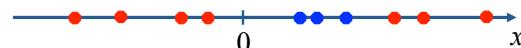


SVM non lineari

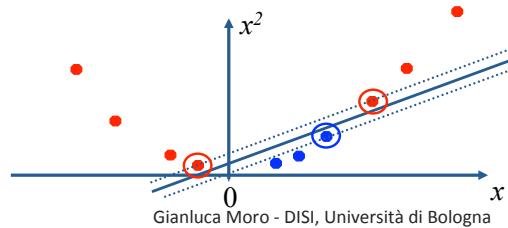
- I metodi precedenti individuano soluzioni per dati linearmente separabili (anche con misclassified):



- Come trattare dati non linearmente separabili, nemmeno con variabili slack, in modo efficace ?



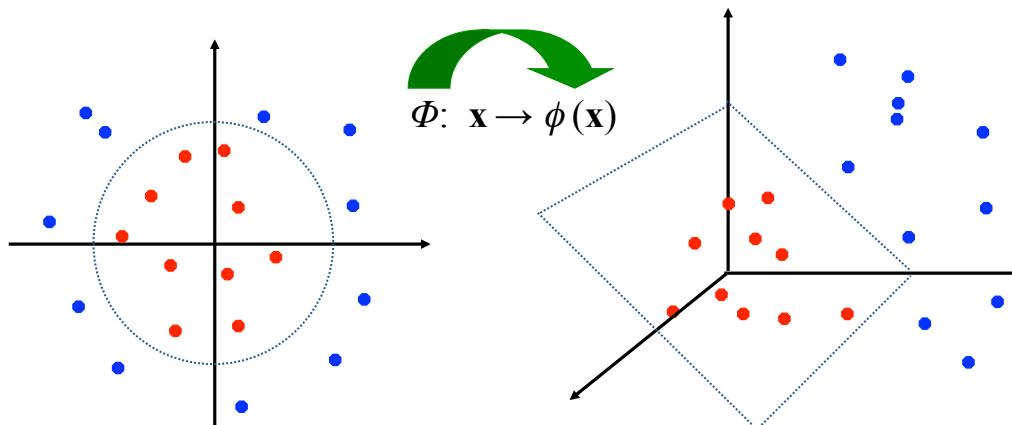
- Mapping dei dati in uno spazio con maggiori dimensioni



53

SVM Non Lineari: Spazio dei Dati

- Mapping dello spazio dei dati, **con una funzione non lineare**, in uno spazio con più dimensioni dove il training set è separabile con **SVM lineari**



Ottimizzazione nello Spazio Trasformato

- Dopo il mapping, il training set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_r, y_r)\}$ diventa $\{(\phi(\mathbf{x}_1), y_1), (\phi(\mathbf{x}_2), y_2), \dots, (\phi(\mathbf{x}_r), y_r)\}$

Calcolare \mathbf{w} e b tali che sia minimizzata

$$\Phi(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{w}}{2} + C \sum_i \xi_i \quad \text{con i vincoli, } \forall \{(\mathbf{x}_i, y_i)\},$$

$$y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \quad \text{con } \xi_i \geq 0 \quad \forall i=1,\dots,r$$

Problema Duale

$$\underset{\alpha}{\operatorname{Max}} : L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j \phi(\mathbf{x}_i^T) \phi(\mathbf{x}_j)$$

$$\text{Vincoli} \quad \sum_{i=1}^r \alpha_i y_i = 0 \quad \text{e} \quad 0 \leq \alpha_i \leq C \quad \forall i=1,\dots,r$$

prodotto scalare
dei vettori
trasformati

Funzione di Classificazione $f(z) = \operatorname{sign} \left(\sum_{k \in SV} y_k \alpha_k \phi(\mathbf{x}_k^T) \phi(z) + b \right)$ con SV= support vector



SVM non lineari: Esempio di Trasformazione

- Supponiamo di definire in un spazio di dati bidimensionale la trasformazione quadratica da 2 a 3 variabili:

$$(\mathbf{x}_1, \mathbf{x}_2) \mapsto (\mathbf{x}_1^2, \mathbf{x}_2^2, \sqrt{2}\mathbf{x}_1\mathbf{x}_2)$$

- L'istanza di training $(2, 3)$ della classe -1 , i.e. $((2, 3), -1)$, è mappata, dallo spazio bidimensionale nello spazio delle feature come $((4, 9, 8.5), -1)$
- In generale questo approccio è soggetto al problema della “maledizione della dimensionalità” → soluzione non scalabile
 - In problemi reali, trasformazioni utili, anche partendo da un numero di attributi ragionevole nello spazio originale, generano un num. di dimensioni nello spazio delle feature, in generale, intrattabile*



SVM non lineari: “Maledizione della Dimensionalità”

- Generalizziamo, ad esempio, la **trasformazione quadratica** precedente di un vettore **n-dimensionale**:

$$(x_1, \dots, x_n) \xrightarrow{2} (x_1^2, x_2^2, \dots, x_n^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \dots, \sqrt{2}x_{i+1}x_{i+2}, \dots, \sqrt{2}x_{i+1}x_n, \dots, \sqrt{2}x_{n-1}x_n)$$

$n(n+1)/2$
dimensioni

- il prodotto scalare dei vettori x e z trasformati è il seguente:

$$\phi(x_1, \dots, x_n) \cdot \phi(z_1, \dots, z_n) = \sum_{i=1}^n x_i^2 z_i^2 + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^n x_i x_j z_i z_j$$

- **num. operazioni = $O(\frac{1}{2} n(n+1))$** , n = dim. dello spazio iniziale
 - $n + n(n-1)/2 = (2n + n(n-1))/2 = n(2+n-1)/2 = n(n+1)/2$
 - In generale trasformazioni con polinomi di grado g costano $\approx O(n^g/g!)$



SVM non Lineari: Trasformaz. Polinomiali di Grado e Dimensionalità Arbitrarie

- Trasformazioni polinomiali di grado g di vettori n-dimensional

$$(x_1, \dots, x_n) \xrightarrow{g} (\dots, \underbrace{\sqrt{g!} \prod_{i=1}^n \frac{x_i^{k_i}}{\sqrt{k_i!}}, \dots}, \text{ t.c. } \forall k_1 + \dots + k_n = g)$$

num. componenti = $\binom{n-1+g}{n-1}$

con $0 \leq k_i \leq g$, $k_i \in \mathbb{N}$, $\forall 1 \leq i \leq n$

- il num. di componenti del vettore trasformato corrisponde anche al numero delle dimensioni del prodotto $\phi(x) \cdot \phi(z)$
- Esempio:

- con vettori in uno spazio iniziale a 100 dimensioni e trasformazione di grado 4, il num. di dimensioni nello spazio trasformato è 4 421 275



Trasformazioni e Costi Computazionali

- Nella formulazione duale il num. di prodotti scalari è $r^2/2$

$$\underset{\alpha}{\text{Max}} : L_D = \sum_{i=1}^r \alpha_i - \frac{1}{2} \sum_{j=1}^r \sum_{i=1}^r y_i y_j \alpha_i \alpha_j \phi(x_i^\top) \phi(x_j)$$

Vincoli $\sum_{i=1}^r \alpha_i y_i = 0$ e $0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, r$

- Con polinomi di **grado g** ed **n dimensioni** il costo è $O(r^2 n^g / 2g!)$
 - e.g. 100 dim., grado 4, 1000 istanze $\rightarrow 1000^2/2 \times 4421275 \approx 2.21 \times 10^{12}$
- Il costo di classificazione di ogni istanza con l'iperpiano ottimo

$$f(z) = \text{sign} \left(\sum_{k \in SV} y_k \alpha_k \phi(x_k^\top) \phi(z) + b \right) \text{ con } SV = \text{support vector}$$

è $O(|SV|^2 n^g / 2g!)$ E.g. come sopra: $\geq 100 \times 4421275 \approx 4.42 \times 10^8$

- Se il risultato di $\phi(x) \cdot \phi(z)$ si ottenessse direttamente da x e z
non occorrerebbero i vettori trasformati $\phi(x)$ e la funzione ϕ



SVM non lineari: Funzioni Kernel

- Kernel Polinomiali: $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^g$
 - Calcoliamo il kernel con grado $g = 2$ con uno spazio iniziale a 2 dimensioni: $\mathbf{x} = (x_1, x_2)$ e $\mathbf{z} = (z_1, z_2)$.
- $$\begin{aligned}
 (\mathbf{x} \cdot \mathbf{z})^2 &= (x_1 z_1 + x_2 z_2)^2 \\
 &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\
 &= (x_1^2, x_2^2, \sqrt{2}x_1 x_2) \cdot (z_1^2, z_2^2, \sqrt{2}z_1 z_2) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})
 \end{aligned}$$
- costo computazionale $O(n)$
 $n = \text{num. dimensioni}$
- costo comput. $O(\frac{1}{2} n(n+1))$

- $\rightarrow (\mathbf{x} \cdot \mathbf{z})^2$ corrisponde al prodotto scalare nello spazio trasformato dove $\phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$
- Kernel Trick:** sostituzione di $\phi(\mathbf{x}) \cdot \phi(\mathbf{z})$ con $K(\mathbf{x}, \mathbf{z})$ nella formulazione duale e nella funzione di classificazione
 - e.g. con $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^g$



Confronto Costi Computazionali con Kernel

- Kernel polinomiale di grado g di vettori n-dimensional

$$K(x, z) = (x^T z)^g = \left(\sum_{k=1}^n x_k z_k \right)^g = \phi(x) \cdot \phi(z)$$

- num. dim. del vettore trasformato $\phi(x) = \binom{n-1+g}{n-1} = \frac{1}{g!} \prod_{i=0}^{g-1} (n+i)$

- invece il num. di termini con funzione Kernel è $O(n)$

Grado del polinomio di trasformazione	$\phi(x) \cdot \phi(z)$	$\phi(x)$: num. dimensioni con n dimensioni iniziali	E.g. con 100 dimensioni ed $r^2/2$	Con Kernel Trick	E.g. con 100 dimensioni
2	$(x \cdot z)^2$	$n(n+1)/2$	$2525 r^2$	$n r^2/2$	$50 r^2$
3	$(x \cdot z)^3$	$n(n+1)(n+2)/6$	$85850 r^2$	$n r^2/2$	$50 r^2$
4	$(x \cdot z)^4$	$n(n+1)(n+2)(n+3)/24$	$2.21 \times 10^6 r^2$	$n r^2/2$	$50 r^2$



Come riconoscere le funzioni kernel ?

- Funzione kernel:
 - è una funzione con dominio $R^m \times R^m$ il cui risultato corrisponde al prodotto scalare dei vettori del dominio in un qualche spazio delle feature R^n con $n > m$
 - $K(a, b) = \phi(a) \cdot \phi(b)$
 - non tutte le funzioni sono kernel, e.g. $K(a,b) = (a \cdot b + 1)^3$ è kernel ?
- Come riconoscere se una funzione è una funzione kernel ?
- Teorema di Mercer:**
 - $K(x,y)$ è kernel se e solo è una funzione *semi-definita positiva*, i.e., per ogni funzione $f(x)$ il cui $\int f^2(x) dx$ è finito, deve valere la seguente:

$$\int K(x, y) f(x) f(y) dx dy \geq 0$$



SVM: Esempi di Funzioni Kernel

- Kernel non lineari possono separare data set altrimenti non separabili
- Efficienti grazie al kernel trick
- Limite: possono generare modelli affetti da overfitting

$$\text{Proprietà delle Funzioni Kernel} \quad K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})$$

$$\text{Polinomiale:} \quad K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + \theta)^g$$

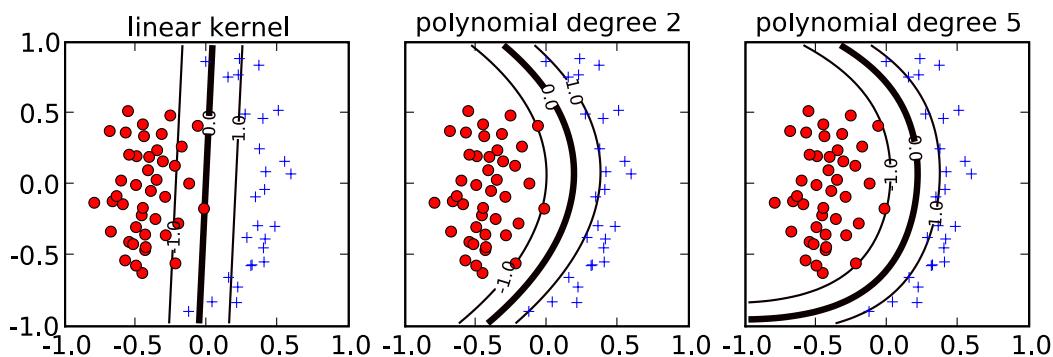
$$\text{Gaussian Radial Basis:} \quad K(\mathbf{x}, \mathbf{y}) = e^{(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)} \text{ con } \gamma = \frac{1}{2\sigma^2}$$

$$\text{Sigmoidale:} \quad K(\mathbf{x}, \mathbf{y}) = \tanh(k \mathbf{x} \cdot \mathbf{y} - \delta)$$

dove $0 \leq \theta \in R, g \in N, k, \delta \in R$



Iperparametri SVM: Grado del Polinomio



- nel data set in figura, il decision boundary con un kernel lineare, e.g. polinomio di grado 1, genera dei misclassified
- lasciando invariato il parametro C e aumentando il grado del polinomio, aumenta la curvatura del decision boundary
 - in questo caso migliora la separazione tra le due classi



Iperparametri SVM: Kernel Gaussiano

- funzione di classificazione con Kernel Gaussiano

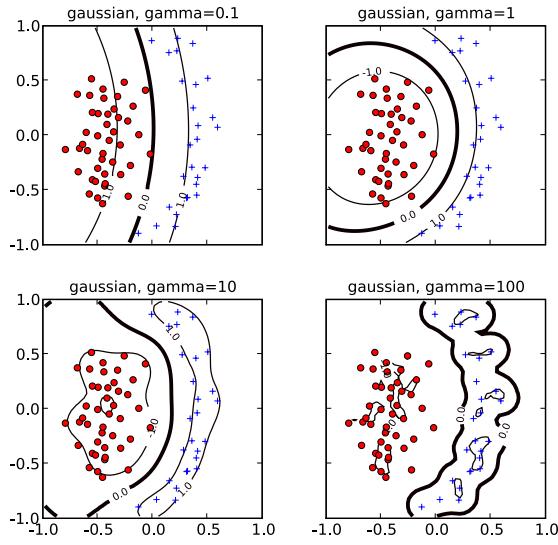
$$f(z) = \text{sign} \left(\sum_{k \in SV} y_k \alpha_k e^{(-\gamma \|x_k^T - z\|^2)} + b \right)$$

con $SV = \text{support vectors}$, $\gamma = \frac{1}{2\sigma^2}$

corrisponde ad una somma di gaussiane centrate sui support vectors

- aumentando gamma, con C invariante, aumenta la flessibilità del decision boundary

- poiché diminuiscono le devstd delle gaussiane e quindi anche le reciproche influenze date dalla sommatoria → caso estremo ogni punto è decision boundary



Iperparametri Kernel Gaussiano: Gamma e C

- $f(z) = \text{sign} \left(\sum_{k \in SV} y_k \alpha_k e^{(-\gamma \|x_k^T - z\|^2)} + b \right)$

con $SV = \text{support vectors}$, $\gamma = \frac{1}{2\sigma^2}$

- lo spazio di ricerca del modello migliore (e.g. accurato) con kernel polinomiale e gaussiano è bidimensionale
 - parametri C e γ oppure gamma che variano su scala logaritmica

- proprietà:

- le curve di livello nel grafico collegano combinazioni dei parametri gamma e C che producono accuratezze equivalenti
- infatti C e gamma influenzano entrambi la flessibilità del decision boundary

