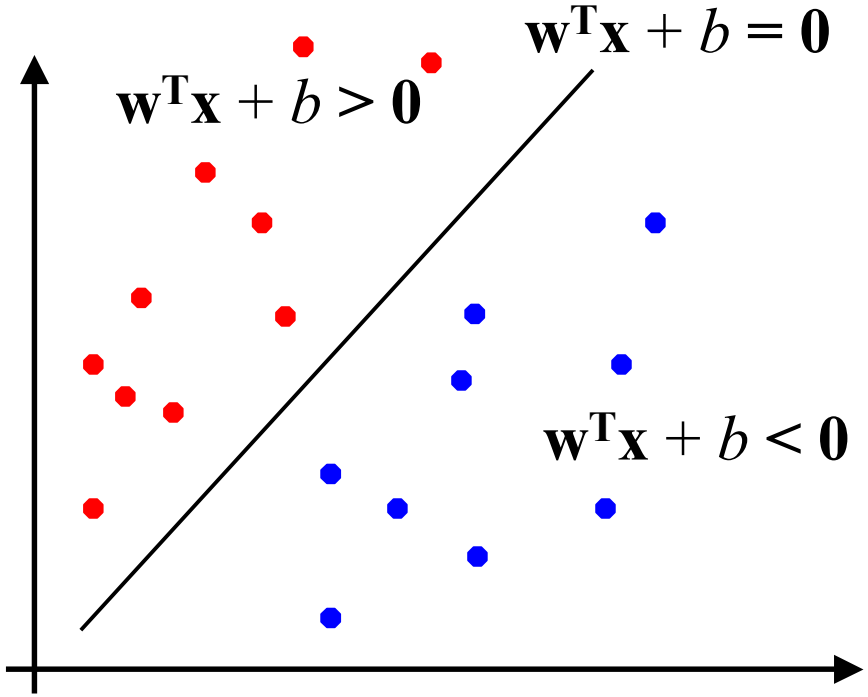
Support Vector Machines

Macchine a vettori di supporto

Separatori Lineari (Percettrone)

- La classificazione binaria può essere vista come un problema di separazione di classi nello spazio delle *feature*



$$\sum_{i=1}^{m} w_i x_i + b = \mathbf{w}^T\mathbf{x} + b$$

$\mathbf{w}$ vettore dei pesi, $\mathbf{w}^T$ trasposta di $\mathbf{w}$

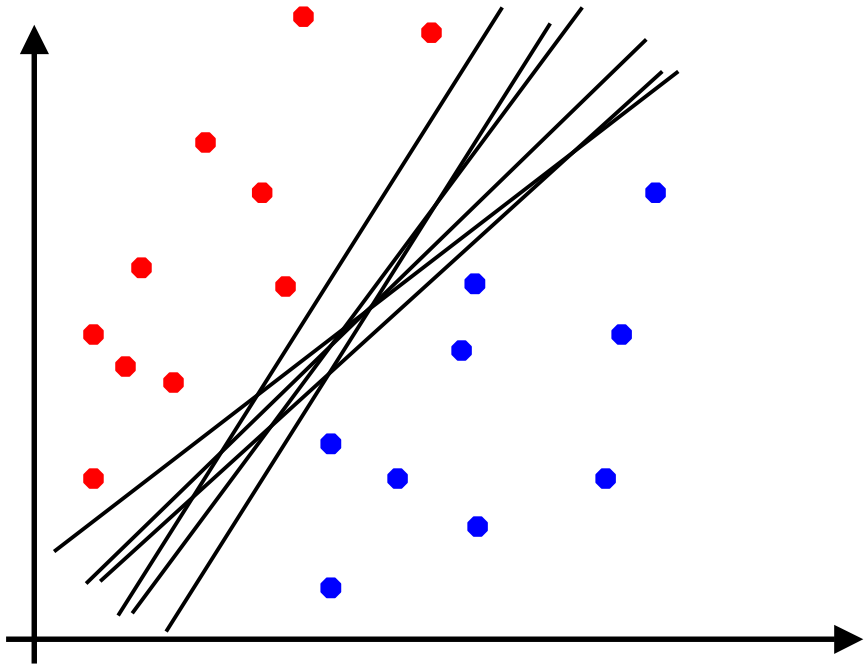
$\mathbf{x}$ vettore associato ad una istanza di X

$\mathbf{w}^T\mathbf{x}$ prodotto scalare

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b) = \begin{cases} +1 \\ \\ -1 \end{cases}$$

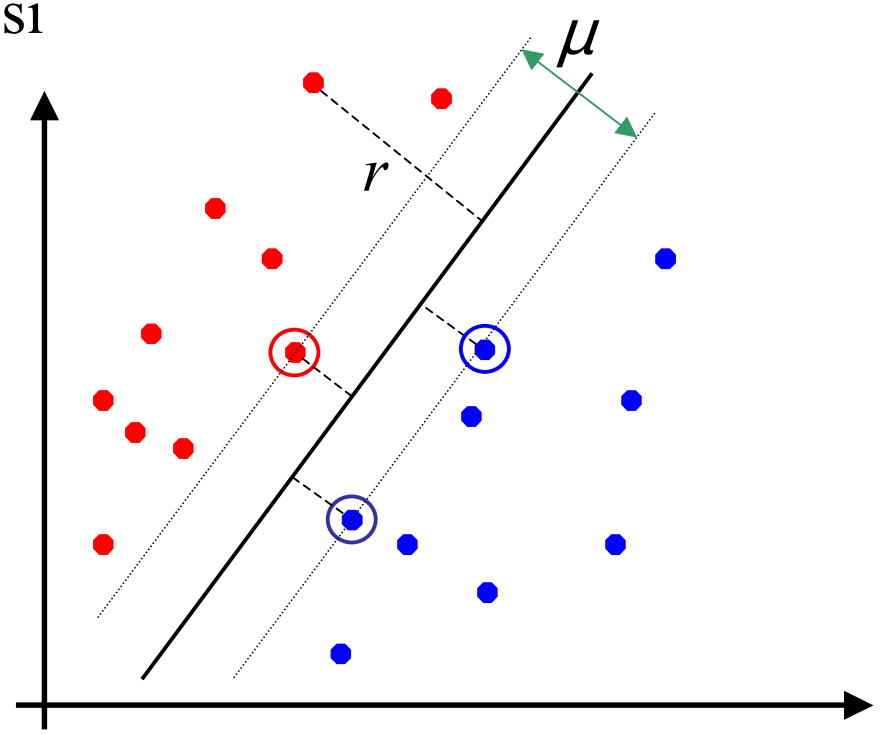
Separatori Lineari

- Quale separatore è ottimo?



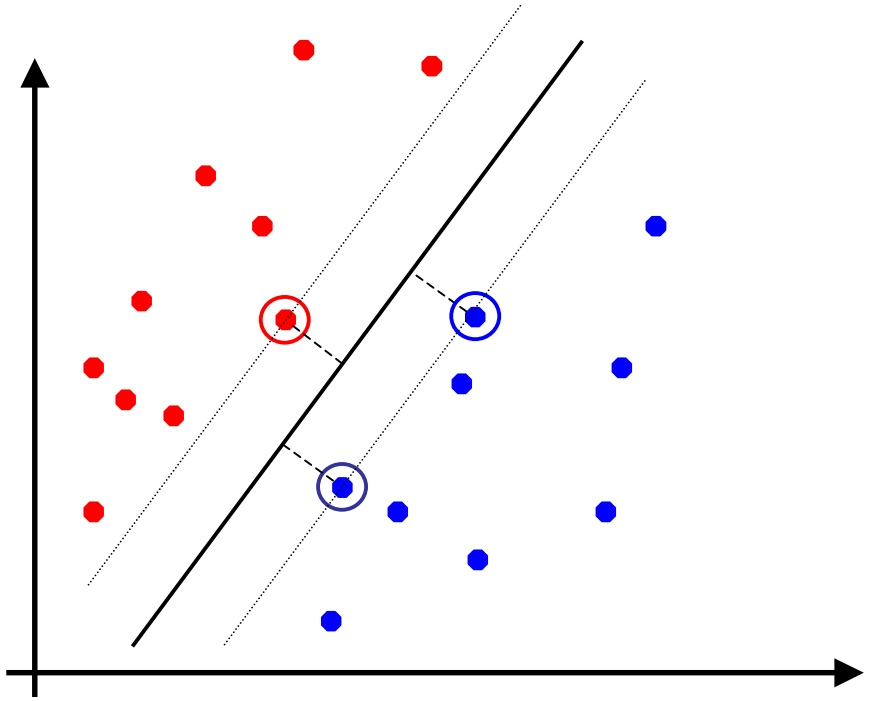
Margine di Classificazione

- La distanza di un esempio dall'iperpiano di separazione è r
- Gli esempi più vicini all'iperpiano si chiamano **support vectors**.
- **Il margine** μ dell'iperpiano di separazione è la distanza minima fra le due classi

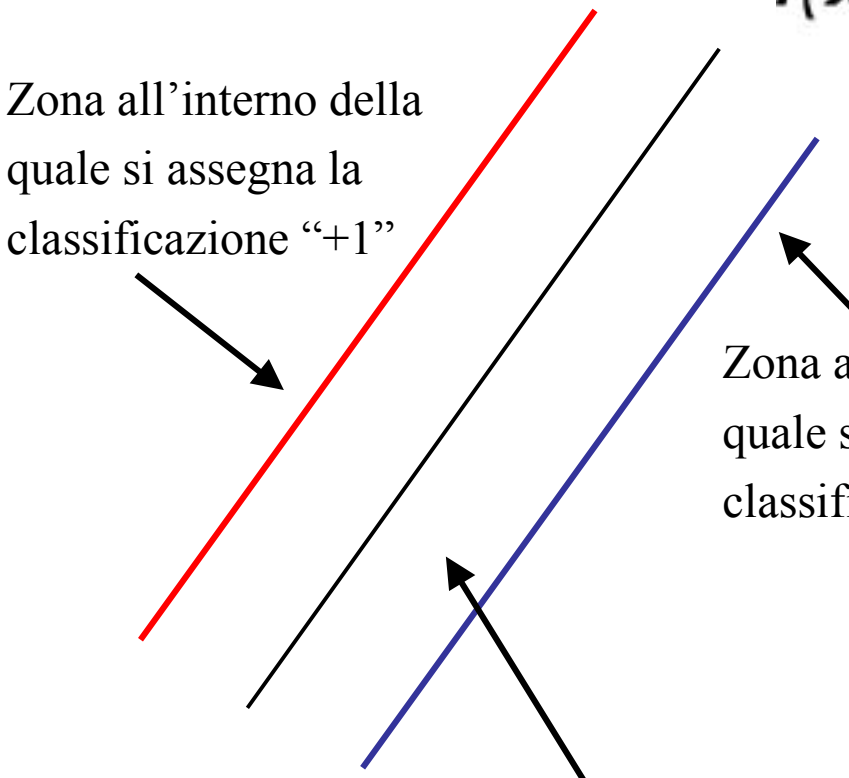


Classificazione con il margine massimo

- Massimizzare il margine corrisponde ad individuare l'iperpiano ottimo *h*
- Questo implica che solo alcuni esempi sono importanti per l'apprendimento, i **vettori di supporto**. Gli altri possono essere ignorati



$$f(\mathbf{x}, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$



Zona all'interno della quale si assegna la classificazione "+1"

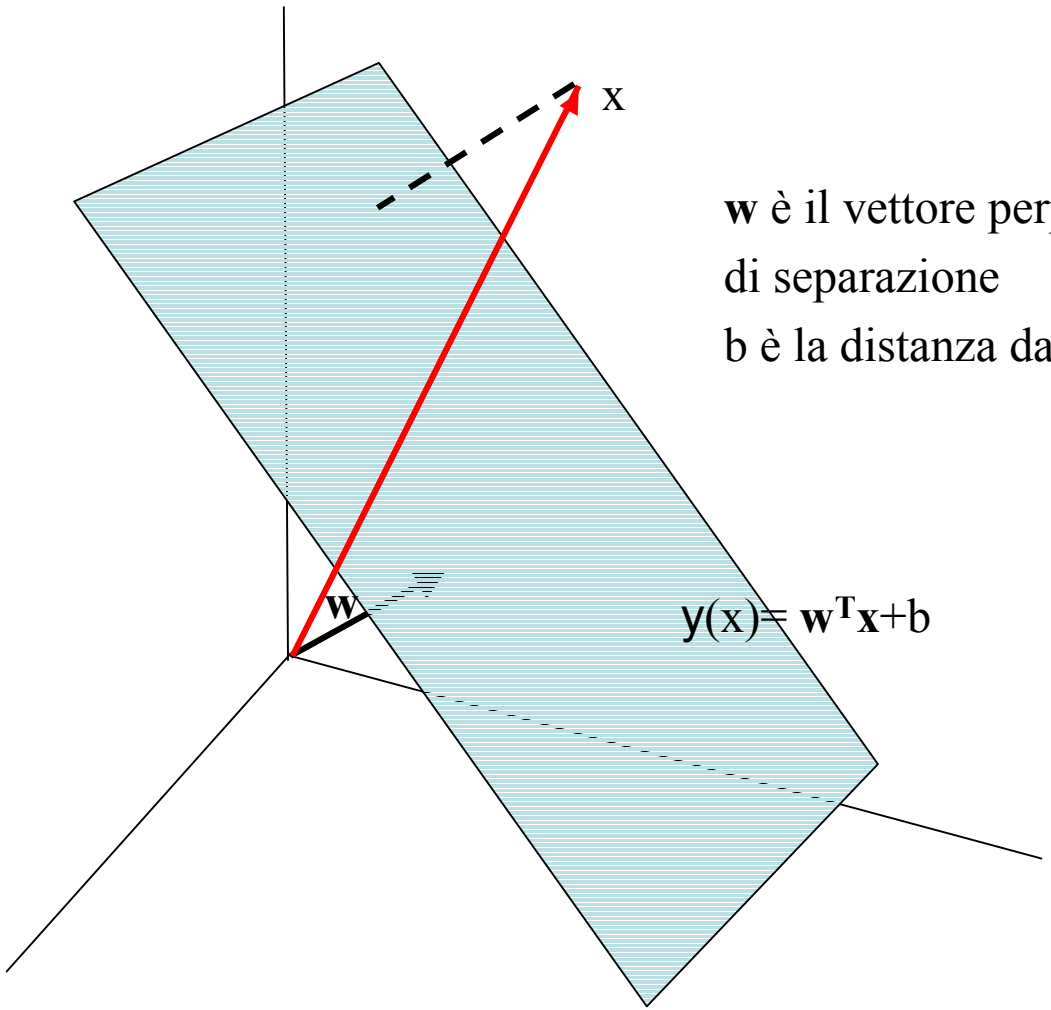
Zona all'interno della quale si assegna la classificazione "-1"

Zona di incertezza

$$\mathbf{w}^T \cdot \mathbf{x} + b \geq 1$$

$$\mathbf{w}^T \cdot \mathbf{x} + b \leq -1$$

$$-1 < \mathbf{w}^T \cdot \mathbf{x} + b < 1$$



w è il vettore perpendicolare al piano di separazione

b è la distanza dall'origine

SVM lineare

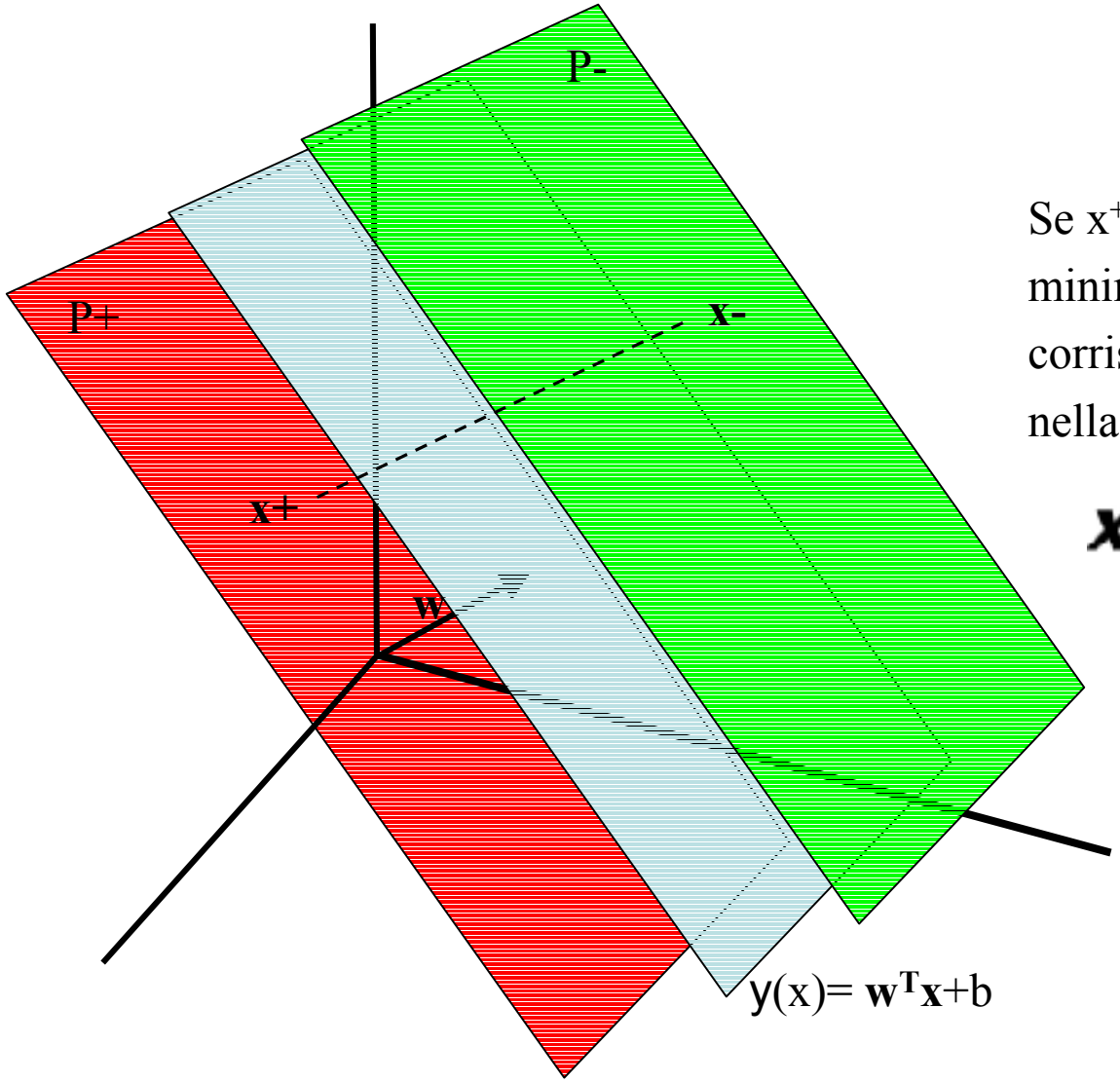
- Se assumiamo che i dati di addestramento $D=\{(\mathbf{x_i}, y_i)\}$ si trovino a distanza almeno 1 dall'iperpiano, valgono le seguenti condizioni per $\mathbf{x_i}$ in D:

$$\mathbf{w^T x_i} + b \geq 1 \quad \text{se } f(\mathbf{x_i}) = +1$$

$$\mathbf{w^T x_i} + b \leq -1 \quad \text{se } f(\mathbf{x_i}) = -1$$

- Per i vettori di supporto, la diseguaglianza diventa una eguaglianza;
- Indichiamo con ρ la distanza fra i piani P^+: $\mathbf{w^T x_i} + b = 1$ e P^-: $\mathbf{w^T x_i} + b = -1$
- Sia x^+ un punto di P^+ e x^- un punto di P^- a distanza minima da x^+
- $\rho = \| \mathbf{x^+} - \mathbf{x^-} \|$, $(\mathbf{x^+} - \mathbf{x^-}) = \lambda \mathbf{w}$

Perché?



Se x^+ e x^- sono a distanza minima , muoversi da x^+ a x^- corrisponde ad un percorso nella direzione di **w**

$$x^+ = x^- + \lambda w$$

Per riassumere:

$$w^T x^+ + b = +1$$

$$w^T x^- + b = -1$$

$$x^+ = x^- + \lambda w$$

Mettendo assieme:

- Abbiamo dunque:

$$x^+ - x^- = \lambda w \Rightarrow x^+ = x^- + \lambda w$$

$$\boldsymbol{w}^T \cdot \boldsymbol{x}^+ + b = 1, \quad \boldsymbol{w}^T \cdot \boldsymbol{x}^- + b = -1$$

$$\Rightarrow \quad (\boldsymbol{w}^T \cdot (\boldsymbol{x}^- + \lambda \boldsymbol{w})) + b = 1 \Rightarrow (\boldsymbol{w}^T \cdot \boldsymbol{x}^- + b) + \lambda \boldsymbol{w}^T \cdot \boldsymbol{w} = \boldsymbol{1}$$

$$-1 + \lambda \boldsymbol{w}^T \cdot \boldsymbol{w} = 1 \Rightarrow \lambda = \frac{2}{\boldsymbol{w}^T \cdot \boldsymbol{w}}$$

$$\rho = \| \lambda \cdot \boldsymbol{w} \| = \frac{2}{\|\boldsymbol{w}\|}$$

- Per massimizzare il margine, dobbiamo minimizzare $\|\mathbf{w}\|$
  - Questo è l'obiettivo di SVM!

SVM lineare (2)

- Il problema di ottimizzazione quadratica che ne risulta è:

Trova $\mathbf{w}$ e b tali che

$\rho = \frac{2}{\|\mathbf{w}\|}$ è massimo; e per ogni $\{(\mathbf{x_i}, y_i)\} \in D$

$\mathbf{w^T x_i} + b \geq 1$ se $y_i = 1$; $\mathbf{w^T x_i} + b \leq -1$ se $y_i = -1$

- Una formulazione migliore basata sulla "programmazione quadratica" è:

Trova $\mathbf{w}$ e b t.c.

$\mathbf{1/2\ w^T w}$ ($=\|\mathbf{w}\|^2$) è minimizzata (1/2 utilizzato per convenienza matematica);

e per ogni $\{(\mathbf{x_i}, y_i)\}$: $y_i (\mathbf{w^T x_i} + b) \geq 1$

Risolvere il problema di ottimizzazione

- Si deve ottimizzare una funzione quadratica soggetta a vincoli lineari: $y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq 1$
- I problemi di ottimizzazione quadratica sono problemi di programmazione matematica ben noti, per i quali esistono vari algoritmi.
- La soluzione comporta l'utilizzo di Lagrangiani

Trovare i minimi di una funzione

- Dall'analisi matematica, data una funzione

$$f(x_1,...,x_k)$$

- l'identificazione dei minimi della funzione è ottenuta mediante il gradiente di *f*
- Esempio:

$$f(x,y) = 2x^2 + y^2$$

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (4x, 2y)$$

$$4x = 0$$

$$2y = 0$$

- *f* ha un minimo in (0, 0)

Minimi di una funzione vincolata

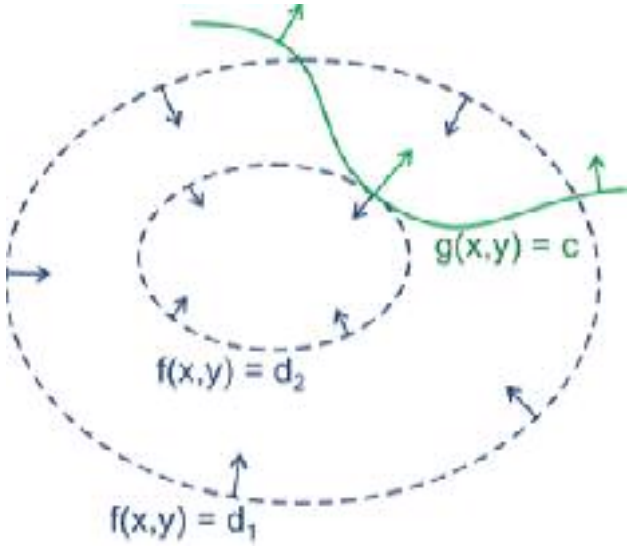
- Ma se la funzione è vincolata da una serie di vincoli:

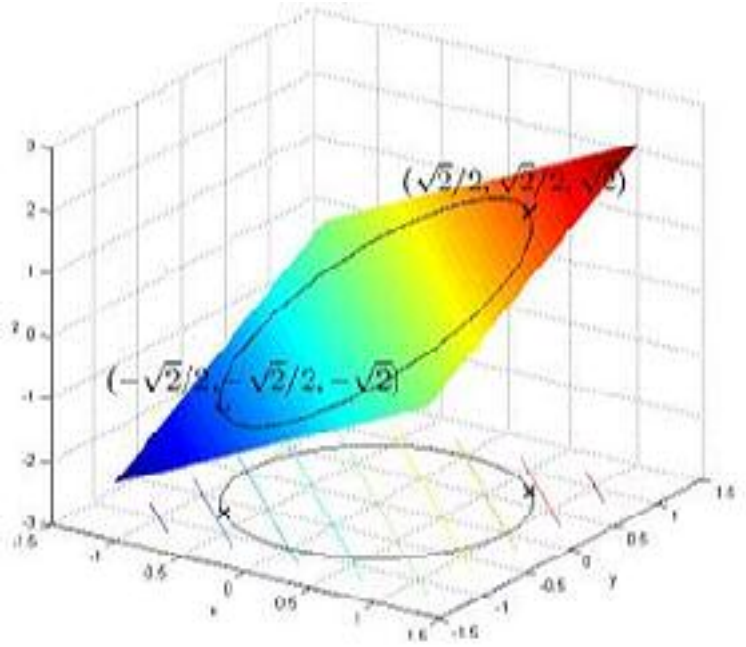
$$g_1(x_1,...,x_k)$$
$$\vdots$$
$$g_n(x_1,...,x_k)$$

- Dobbiamo trovare i valori estremi di *f* per i soli punti ristretti dai vincoli *g*
- I valori minimi di *f* sono ottenuti quando le superfici si “toccano”

Graficamente:

- Due esempi:





$$f(x,y) = x^2 + y^2$$
$$g(x,y) = c$$

$$f(x,y) = x + y$$
$$g(x,y) = x^2 + y^2 - 1$$

Lagrangiani

- Data una funzione da ottimizzare f ed un insieme di condizioni g_1, ..., g_n, un **Lagrangiano** è una funzione $L(f,g_1, ..., g_n, \alpha_1, ..., \alpha_n)$ che "incorpora" le condizioni nel problema di ottimizzazione

$$L(f,\alpha_1,...,\alpha_n) = f(x_1,...,x_k) - \sum_{i=1}^{n} \alpha_i g_i(x_1,...,x_k)$$

Lagrangiani

- Ad esempio, se f = $w^2/2$ e $g_1 = wx - 1 \geq 0$, allora

$$L(w,\alpha) = \frac{w^2}{2} - \alpha(wx - 1), \quad \alpha \geq 0$$

- Si calcolano le *derivate* rispetto alle variabili del lagrangiano (**w** e α in questo esempio) e si impone siano =0
- Risolvendo il sistema di equazioni ottenuto, si ricavano i valori che soddisfano il problema
- L deve essere massimizzata rispetto alle **variabili primarie** (**w** nell'esempio), ma minimizzata rispetto alle **variabili duali** (α)

Calcola le derivate parziali di α nell'esempio precedente

$$L(f(w),\alpha) = \frac{w^2}{2} - \alpha(wx-1), \;\; \alpha \geq 0$$

$$\frac{\partial(L)}{\partial w} = w - \alpha x = 0 \;\; \Rightarrow w = \alpha x$$

$$\frac{\partial(L)}{\partial \alpha} = -wx = 0$$

Torniamo al problema di SVM

- **Minimizzare** il seguente *Lagrangiano*:

(x_i, y_i) learning set

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y_i((\mathbf{w}^T \mathbf{x_i}) + b) - 1), \quad \alpha_i \geq 0$$

- Imponiamo dapprima:

$$\frac{\partial(L)}{\partial b} = 0, \quad \frac{\partial(L)}{\partial w} = 0 \quad da \quad cui :$$

$$(1) \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (2) \ \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$

- **La condizione $\alpha \geq 0$ porta a selezionare solo un sotto-insieme di vettori, per i quali questa condizione deve essere verificata ($\alpha > 0$), ovvero i Support Vectors (i vettori di "frontiera"), da cui**:

$$\boldsymbol{w} = \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i$$

Langrangiani:
problema primario e problema duale

Problema primario

$$\min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha}) = \min_{\mathbf{w},b}\left(\frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_i (\alpha_i y_i \mathbf{x_i}^T)\cdot\mathbf{w} - b\sum \alpha_i y_i + \sum_i \alpha_i\right) =$$

Problema duale

$$\max_{\boldsymbol{\alpha}=(\alpha_1,\ldots\alpha_m)} \min_{\mathbf{w},b} L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\sum_i (\alpha_i y_i \mathbf{x_i}^T)\sum_j (\alpha_j y_j \mathbf{x_j}) - \sum_i (\alpha_i y_i \mathbf{x_i}^T)\sum_j (\alpha_j y_j \mathbf{x_j}) - b\cdot 0 + \sum_i \alpha_i) =$$

$$\max\left(\sum_i \alpha_i - \frac{1}{2}\sum_{i,j=1..n} \alpha_i\alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}\right)$$

RIASSUMIAMO

1) Formulazione del problema di ottimizzazione:

$$\min_{\vec{w},b} \frac{1}{2}||\vec{w}||^2$$

soggetto a: $\forall i \in \{1,\ldots,n\} : y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \geq 0$

2) Espressione del problema con un Lagrangiano:

$$L(\vec{w},b,\alpha) = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{n}\alpha_i(y_i(\vec{w}\cdot\vec{x}_i + b) - 1)$$

3)Ottieni la soluzione (teorema di Kuhn-Tucker)

$$\frac{\partial L(\vec{w},b,\alpha)}{\partial \vec{w}} = 0 \quad \Leftrightarrow \quad \vec{w^*} = \sum_{i=1}^{n} y_i \alpha_i^* \vec{x}_i \text{ E1}$$

$$\frac{\partial L(\vec{w},b,\alpha)}{\partial b} = 0 \quad \Leftrightarrow \quad \sum_{i=1}^{n} y_i \alpha_i^* = 0 \text{ E2}$$

$$\alpha_i^*[y_i(\vec{w}^*\cdot\vec{x}_i + b^*) - 1] = 0 \;\; i = 1,\ldots,n$$

I vettori per cui $\alpha^*>0$ sono detti vettori di supporto.

4) Ottieni la formulazione duale eliminando le variabili primarie w,b in 2) (formule E1 e E2)

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \vec{x}_i\cdot\vec{x}_j$$

soggetto a: $\forall i \in \{1,\ldots,n\} : \alpha_i \geq 0$ e $\sum_{i=1}^{n} y_i\alpha_i = 0.$

Funzione di decisione

- La funzione di decisione era:

$$f(x) = \text{sgn}(\mathbf{w}^T \cdot \mathbf{x} + b)$$

- ma avendo la condizione:

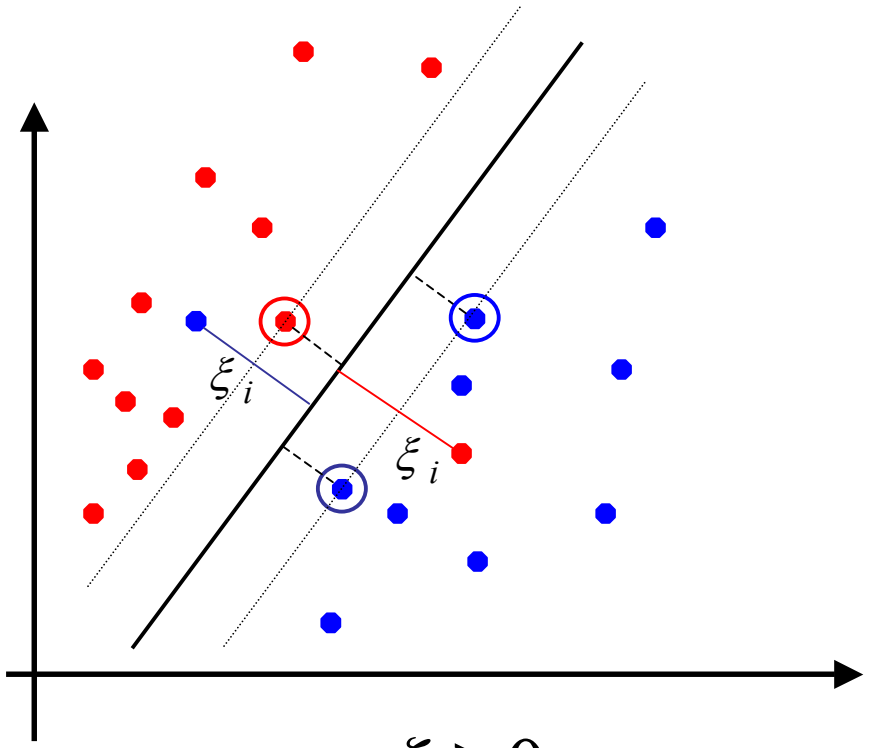
$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$$

- otteniamo:

$$f(x) = \text{sgn}(\sum_{i=1}^{m} \alpha_i y_i \mathbf{x_i}^T \cdot \mathbf{x} + b)$$

Margini “Soft”

- Se il set di addestramento **non è linearmente separabile**?
- *Si introducono le slack variables* ξ_i che consentono la classificazione errata di qualche punto.



Soft Margin Classification

- Problema lineare:

  Trova $\mathbf{w}$ e b t.c.
  $\mathbf{1/2}\ \mathbf{w}^{\mathrm{T}}\mathbf{w}$ è minimizzata e per ogni $\{(\mathbf{x_i}, y_i)\}$
  $y_i\ (\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + \mathrm{b}) \geq 1$

- Con le slack variables:

  Trova $\mathbf{w}$ e b t.c.
  $\mathbf{1/2}\ \mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum \xi_i$ è minimizzata e per ogni $\{(\mathbf{x_i}, y_i)\}$
  $y_i\ (\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \geq 1 - \xi_i$ e $\xi_i \geq 0$ per ogni i

- Il parametro C controlla l'overfitting.

Sommario di SVM lineare

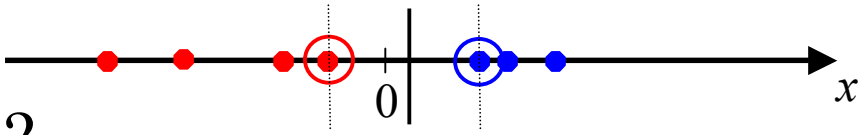
- Il classificatore (funzione obiettivo) è un iperpiano di separazione.
- I "punti" (esempi) più importanti sono i vettori di support ("sostengono" l'iperpiano, mantenedolo in equilibrio)
- Algoritmi di ottimizzazione quadratica identificano quali punti rappresentano il "supporto"
- Nella formulazione del problema e nella soluzione appaiono i prodotti scalari:

Trova $\alpha_1 \ldots \alpha_n$ t.c.
$\mathbf{W}(\alpha) = \sum \alpha_i - \mathbf{1/2} \sum\sum \alpha_i \alpha_j y_i y_j \mathbf{x_i}^T \mathbf{x_j}$ **è** massimizzata e
(1) $\sum \alpha_i y_i = 0$
(2) $0 \le \alpha_i \le C$ per ogni α_i

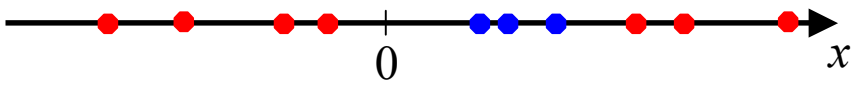
$f(\mathbf{x}) = \text{sgn}(\sum \alpha_i y_i \mathbf{x_i}^T \mathbf{x} + b)$

Non-linear SVMs

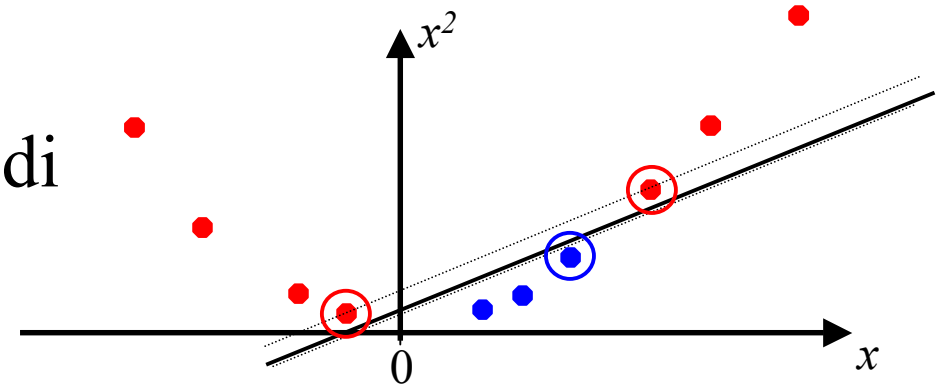
- Se i dataset sono separabili con un po' di rumore le cose funzionano:



- Ma se il rumore è eccessivo?

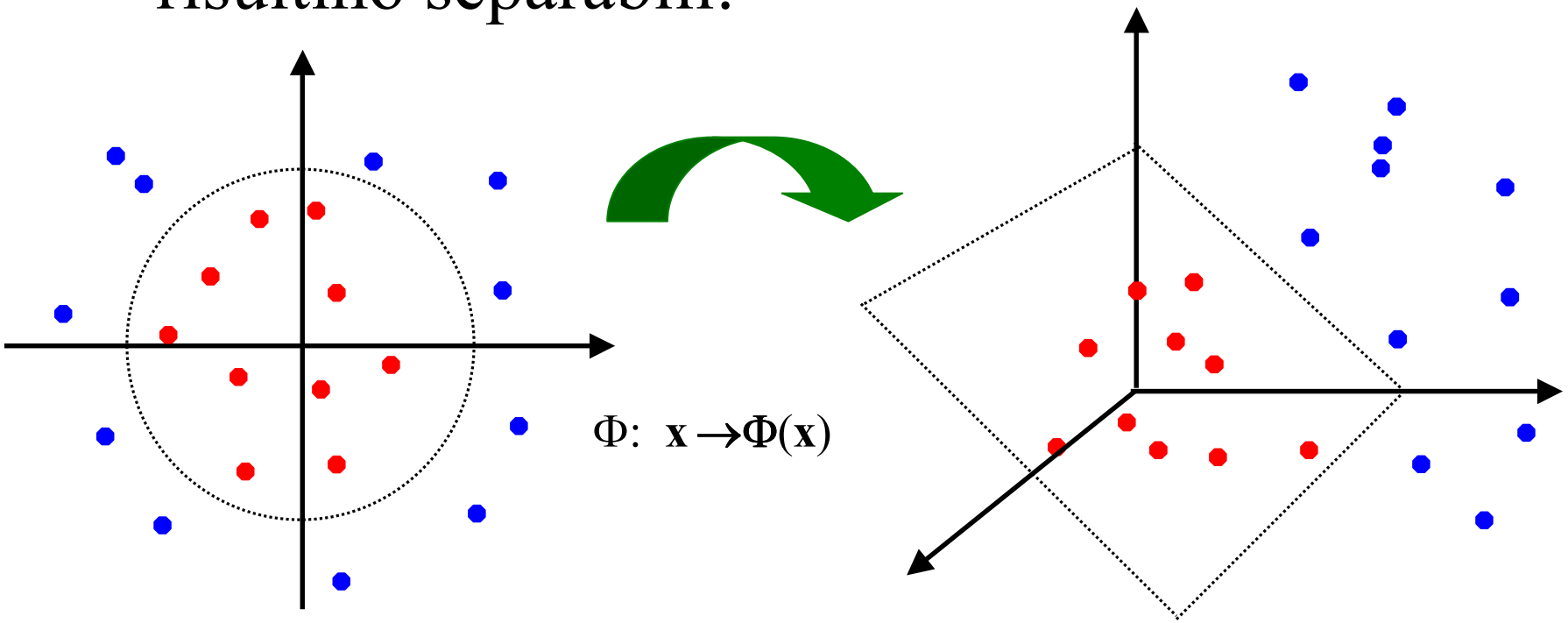


- Si può proiettare il problema in uno spazio di dimensioni maggiori:



Non-linear SVMs: Feature spaces

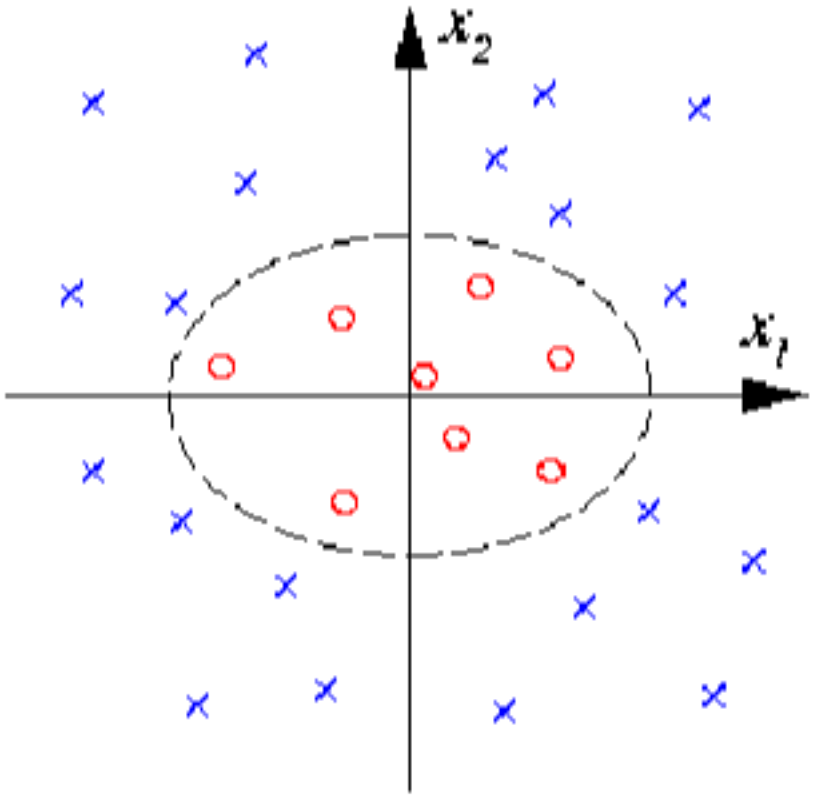
- Proiettare in uno spazio nel quale i dati risultino separabili:

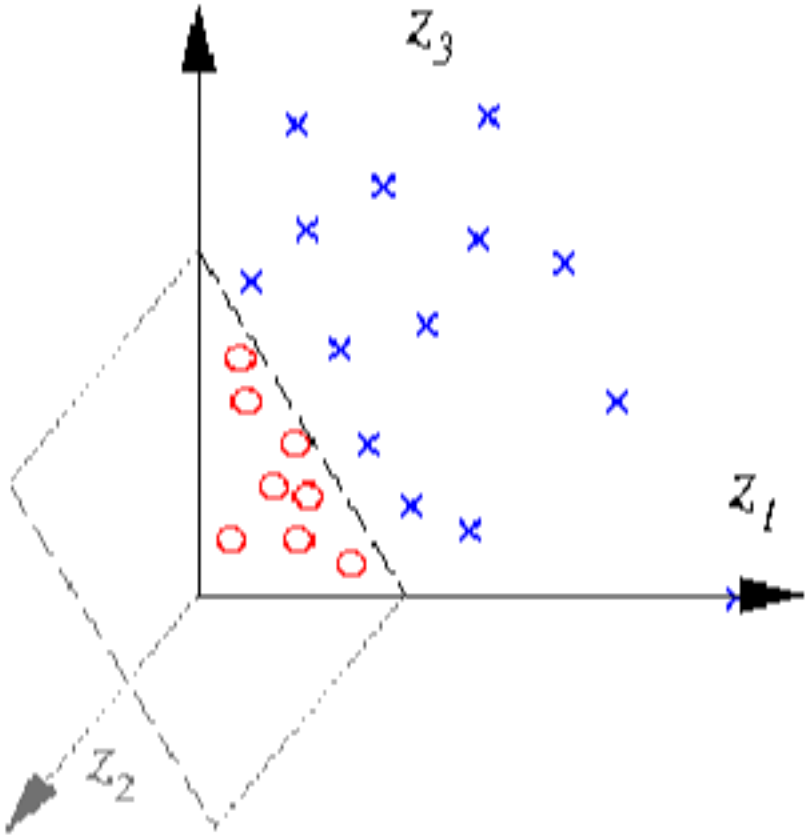


Esempio di funzione Φ

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$





Funzioni Kernel

- Una funzione *kernel* è una funzione che corrisponde ad un prodotto scalare in uno spazio esteso
- Il classificatore lineare si basa sul prodotto scalare fra vettori dello spazio delle istanze X (quindi, non esteso): $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$
- Se ogni punto è traslato in uno spazio di dimensioni maggiori attraverso una trasformazione $\Phi : \mathbf{x} \rightarrow \Phi(\mathbf{x})$ il prodotto scalare diventa:

$$K(\mathbf{x}_i,\mathbf{x}_j)= \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)=\mathbf{x'}_i^T\mathbf{x'}_j$$

dove $\mathbf{x'}_i$ e $\mathbf{x'}_j$ indicano trasformazioni non lineari

Funzioni kernel: un esempio

- Abbiamo vettori a 2 dimensioni $\mathbf{x}=(x_1, x_2)$
  Sia $K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^2$,
  Dobbiamo mostrare che $K(\mathbf{x_i},\mathbf{x_j}) = \Phi(\mathbf{x_i})^{\mathbf{T}}\Phi(\mathbf{x_j})$:

  $K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^2$, $= 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 +$
  $2x_{i1}x_{j1} + 2x_{i2}x_{j2}=$
  $= (1, x_{i1}^2, \sqrt{2}\, x_{i1}x_{i2}\,, x_{i2}^2\,, \sqrt{2}x_{i1}\,, \sqrt{2}x_{i2})^{\mathbf{T}}\, (1, x_{i1}^2, \sqrt{2}\, x_{i1}x_{i2}\,,$
  $x_{i2}^2\,, \sqrt{2}x_{i1}\,, \sqrt{2}x_{i2}) = \Phi\,(\mathbf{x_i})^{\mathbf{T}}\Phi\,(\mathbf{x_j})$, dove

  $$\Phi\,(\mathbf{x}) = (1, x_{i1}^2, \sqrt{2}\, x_{i1}x_{i2}\,, x_{i2}^2\,, \sqrt{2}x_{i1}\,, \sqrt{2}x_{i2})$$

Quali funzioni sono Kernel?

- Per alcune funzioni $K(\mathbf{x_i},\mathbf{x_j})$ verificare che $K(\mathbf{x_i},\mathbf{x_j})= \Phi(\mathbf{x_i})^{\mathrm{T}}\Phi(\mathbf{x_j})$ è complesso.
- Teorema di Mercer:
  - *Ogni funzione la cui matrice associata Gram K (ovvero t.c. $K_{ij} := K(\boldsymbol{x_i},\boldsymbol{x_j})$) è positiva semidefinita è un kernel, ovvero:*

$$\sum_{i,j} c_i \bar{c}_j K_{ij} \geq 0, \quad \forall c_i \in C$$

Esempi di Funzioni Kernel

- Lineare: $K(\mathbf{x_i},\mathbf{x_j})= \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$
- Polinomiale potenza di p:
  $K(\mathbf{x_i},\mathbf{x_j})= (1+ \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^p$
- Gaussiana (*radial-basis function network*):
  $$K(\mathbf{x_i}, \mathbf{x_j}) = e^{-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2}}$$
- Percettrone a due stadi:
  $K(\mathbf{x_i},\mathbf{x_j})= \tanh(\beta_0 \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j} + \beta_1)$

Applicazioni

- SVMs sono attualmente fra i migliori classificatori in una varietà di problemi (es. elaborazione del linguaggio e genomica).
- Il tuning dei parametri SVM è un'arte: la selezione di uno specifico kernel e i parametri viene eseguita in modo empirico (tenta e verifica: *trial and test*)

Applicativi

- SVM^{light} - *http://svmlight.joachims.org*
- bsvm - *http://www.csie.ntu.edu.tw/~cjlin/bsvm/*
- libsvm - *http://www.csie.ntu.edu.tw/~cjlin/libsvm/*

- **Differenze:** *funzioni Kernel utilizzabili, tecnica di ottimizzazione, possibilità di classificazione multipla, interfacce di utente*