Support Vector Machines

Classificazione con SVM

N. Torelli

2023

University of Trieste

Introduzione

Introduzione

- Con il termine **Support Vector Machines** si intendono metodi di classificazione per i quali non si ricorre a un approccio statistico (per modellare la probabilità di appartenenza a una classe) ma si utilizza un approccio puramente geometrico
- Si cerca una linea (o un iperpiano) nello spazio delle variabili X che separi le classi
- Nel seguito si tratterà essenzialmente il caso della classificazione binaria
- Nel libro di JWHT si distingue fra diversi affinamenti di tale idea: il classificatore con massimo margine (se le classi sono perfettamente separabili con un iperpiano), il classificatore con vettore di supporto (nel caso in cui non tutti i dati sono separabili con un iperpiano) e le Support Vector Machines (per le quali il separatore fra le classi può essere non lineare)
- Si noti che per tale algoritmo si codifica la variabile risposta dicotomica Y così che assuma i due valori -1 e 1.

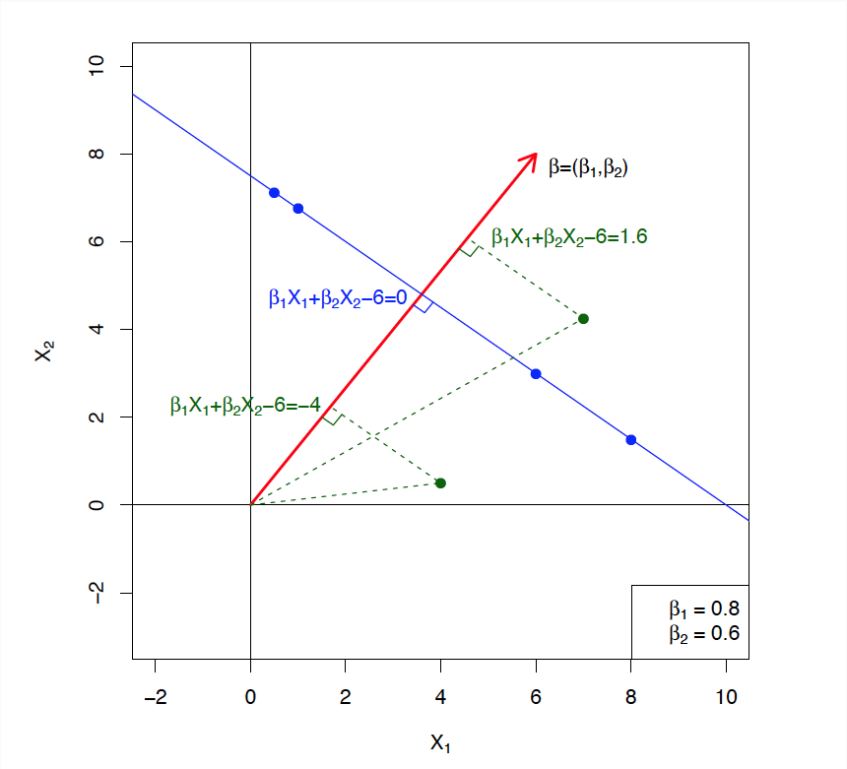
Classificatore con Massimo margine

Iperpiano di separazione e classificatore con massimo margine

- In generale l'equazione di un iperpiano nello spazio a p dimensioni ha la forma

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0$$

- Se $p = 2$ l'iperpiano si riduce a una retta



Separazione in due classi

- Un iperpiano di (perfetta) separazione per un insieme di n osservazioni è tale che

$$f(X) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} > 0 \quad \text{se} \quad Y_i = 1$$

e

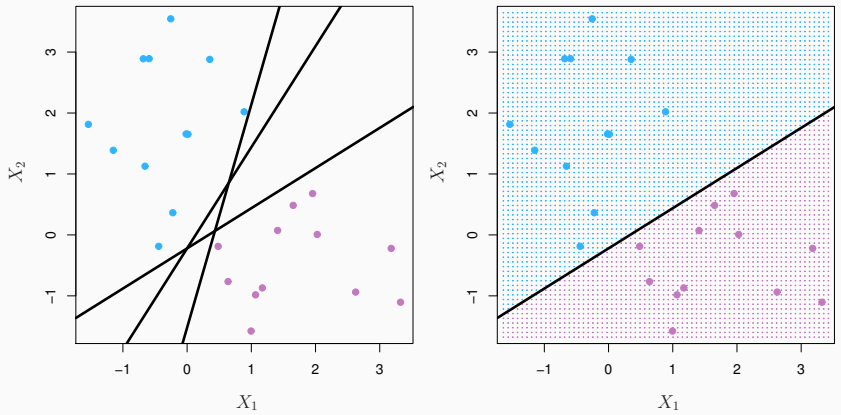
$$f(X) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} < 0 \quad \text{se} \quad Y_i = -1$$

- Si può anche scrivere che tale iperpiano ha la seguente proprietà

$$Y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip}) > 0$$

- Se tale iperpiano esiste abbiamo un classificatore naturale

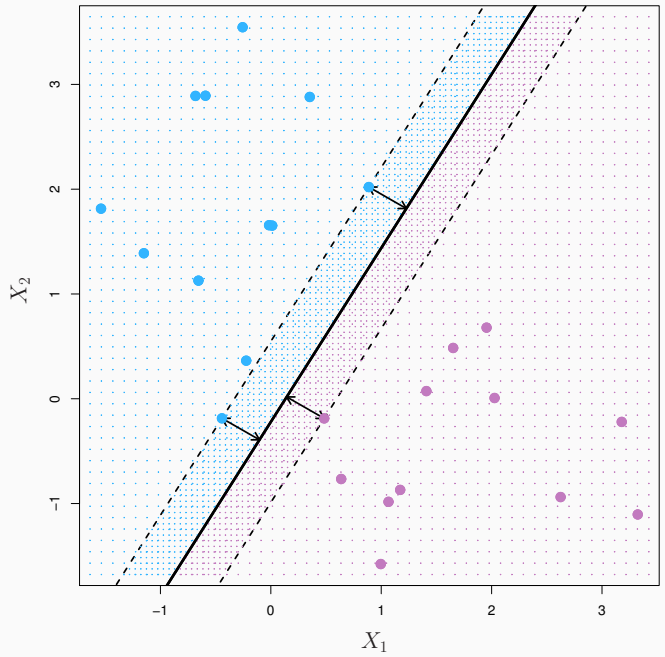
Classificatore con massimo margine



- i punti blu del grafico sono quelli per cui $Y_i = +1$ invece per quelli rosa $Y_i = -1$, allora $Y_i f(X_i) > 0 \ \forall i$ e $f(X) = 0$ è l'iperpiano di separazione.
- E' evidentemente che vi sono più iperpiani separatori che andrebbero bene
- Si noti che il valore di $f(x_i)$ è la distanza del punto dall'iperpiano di separazione e può essere usato come una misura della nostra fiducia nella sua classificazione in una delle due porzioni di spazio

Classificatore con Massimo Margine

- Tra tutti gli iperpiani di separazione, si cerca quello per cui è massimo il divario, o il *margine* tra le due classi. Il margine è definito come la minima distanza delle osservazioni dall'iperpiano.
- Nel grafico è rappresentato il margine ovvero la distanza dei punti più vicini alla retta di separazione e la retta di separazione è quella per cui esso è massimo. Tre punti nel grafico sono sul margine.

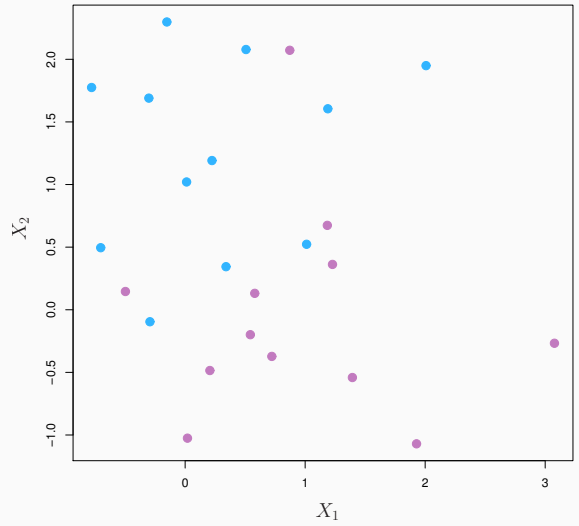


Ricerca dell'iperpiano di separazione con massimo margine

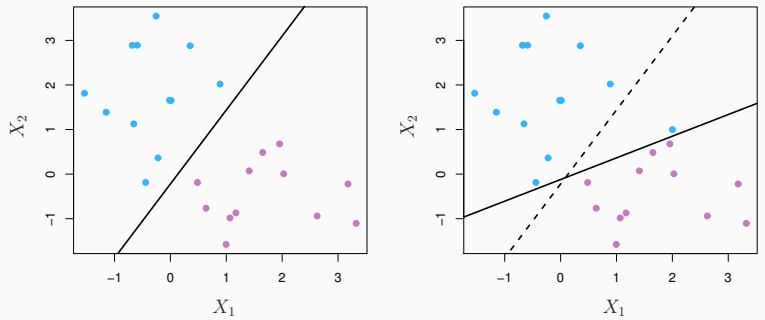
- La ricerca dell'iperpiano di separazione con massimo margine è un problema di ottimizzazione vincolata
  - Si cercano i valori $\beta_0, \beta_1, \ldots, \beta_p$ per cui risulti massimo il margine $M > 0$
  - con il vincolo (di normalizzazione) $\sum_{j=1}^{p} \beta_j^2 = 1$ per cui risulti
  - $y_i(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip}) > M$
- Si cerca cioè quell'iperpiano per cui risulti massima la distanza fra i punti più vicini all'iperpiano.
- Nella figura vi sono 3 punti equidistanti dal mrgine: questi tre punti sono i vettori di supporto (*support vectors*). Si noti che l'iperpiano dipende essenzialmente da questi punti e non da tutte le altre osservazioni.
- Si tratta in realtà di un problema di ottimizzazione quadratica convessa che può essere risolto in modo efficiente (per i dettagli si veda il volume di HFT)
- Una volta definito l'iperpiano $\hat{f}(X)$ sulla base di un training set si può calcolare $\hat{f}(x)$ per un x^* del test set e definire la regola di classificazione per cui se $\hat{f}(x*) > 0$ allora $y = 1$

Caso di non perfetta separazione (o di separazione instabile)

- Si noti tuttavia che nella figura sotto non esiste un iperpiano (una retta) di perfetta separazione



- e si noti che, anche nel caso di prefetta separazione, un solo punto aggiunto cambia la retta in modo drastico: l'iperpiano esiste ma è estremamente instabile



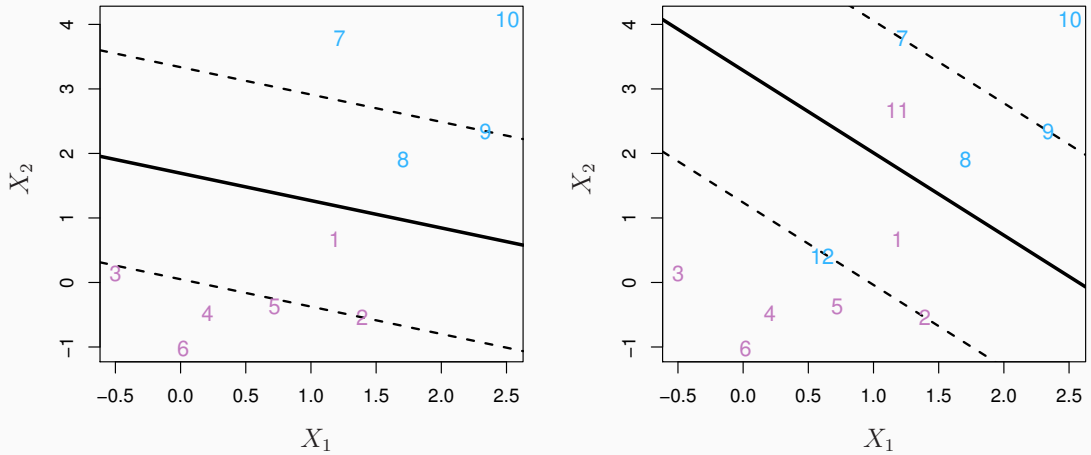
Classificatore con vettori di supporto

Uso del margine morbido (soft margin)

- In casi come quelli illustrati (di maggiore interesse pratico) si potrebbe considerare un classificatore basato su un iperpiano che non separi perfettamente le due classi
- Questo consente
  - Minore sensibilità a singole osservazioni
  - Migliore classificazione della maggior parte dei dati
- Cioè, potrebbe essere inevitabile classificare erroneamente alcuni dati del training set per poi ottenere una migliore performance nella classificazione di nuove osservazioni.

Uso del margine morbido

Si consentirà quindi ad alcune osservazioni di stare dalla parte errata del margine o addirittura di stare dalla parte errata dell'iperpiano, come nella figura sotto. Ove ci sono sia osservazioni sul lato errato del margine (le osservazioni 1 e 8 nella prima figura) o anche sul lato errato dell'iperpiano (la 11 nella seconda figura)

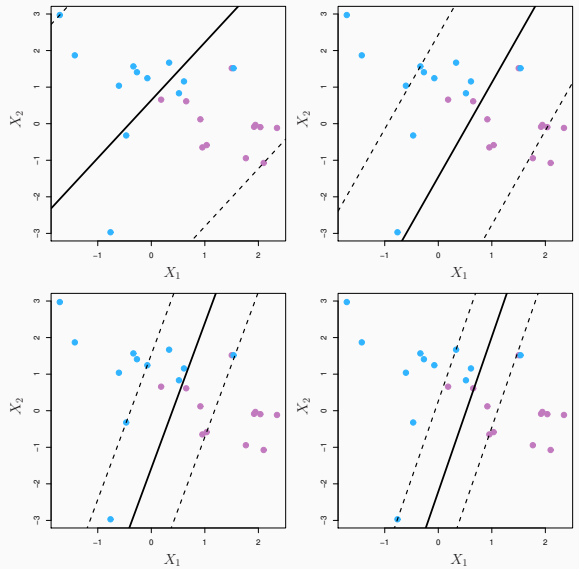


Ottimizzazione con margine morbido

- Si cercano i valori $\beta_0, \beta_1, \ldots, \beta_p$
  - per cui risulti massimo il margine $M > 0$
  - con il vincolo (di normalizzazione) $\sum_{j=1}^{p} \beta_j^2 = 1$
  - e tale che $y_i(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p) > M(1 - \epsilon_i)$
    - ove $\epsilon_i \geq 0, \quad \sum_{1=1}^{N} \epsilon_i \leq C$
    - $C > 0$ è una costante che ha il ruolo di parametro di tuning
- M è il margine e vogliamo sia più ampio possibile
- le variabili ϵ_i consentono a ciascuna osservazione di stare sul lato sbagliato del margine o dell'iperpiano
- ϵ_i=0 se l'i-esima osservazione è sul lato giusto rispetto al margine
- $\epsilon_i > 0$ se è dentro il margine o $\epsilon_i > 1$ se è sul lato sbagliato

Il ruolo del parametro C nel controllo del trade-off distorsione-varianza

- C limita il numero di osservazioni che violano il margine e quindi dice quanto si è severi nel sopportare tali violazioni
- se $C > 0$ non più di C dati possono essere mal classificati
- Se C è elevato si è più tolleranti a violazioni del margine che è quindi più ampio. Nei grafici sotto si rappresenta il margine per valori di C decrescenti



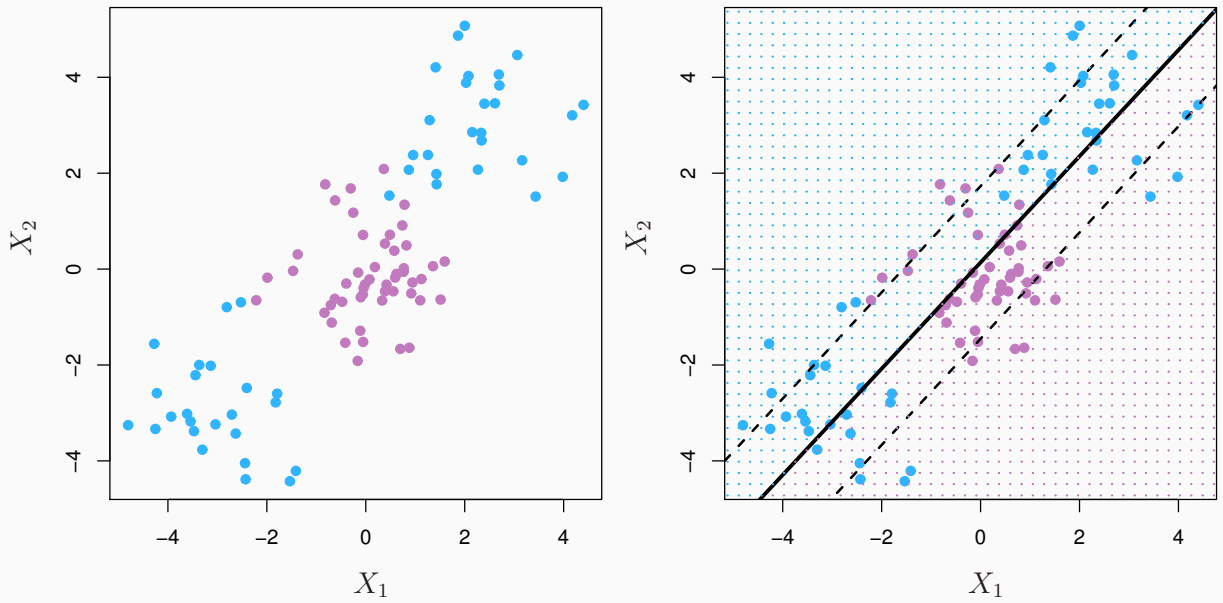
Vettori di supporto (support vectors classifier)

- si noti che solo le osservazioni che sono sul margine o che lo violano (stando dalla parte giusta dell'iperpiano) hanno rilievo nel determinare la funzione
- Tali osserrvazioni sono dette **vettori di supporto**
- Le osservazioni che sono distanti dal margine e sul lato giusto dell'iperpiano hanno scarsa influenza (il che rende il metodo robusto rispetto osservazioni anomale)
- Se C è elevato trovo molti vettori di supporto, se C è vicino a 0 saranno poche le osservazioni che influiranno sulla determinazione dell'iperpiano
- quindi se C è piccolo il classificatore avrà poco bias ma molta varianza

Confini di separazione non lineari

Confini di separazione non lineari

La figura mostra un caso nel quale un confine lineare per separare le classi non funziona qualunque sia il valore di C



Un'idea per superare il problema è quello di allargare lo spazio delle covariate

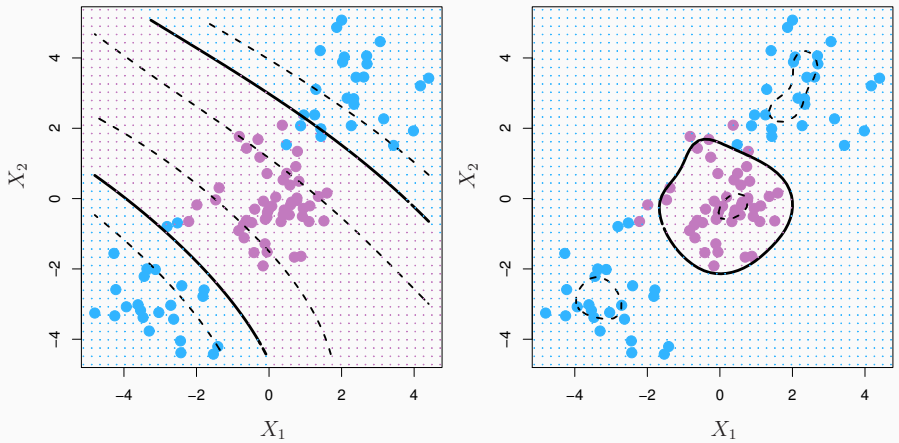
Allargamento dello spazio delle covariate

- Una semplice idea potrebbe essere quella di considerare per ogni variabile il suo quadrato, cioè $X_1, X_1^2, X_2.X_2^2, \ldots$
- Lo spazio entro cui si collocano i valori di X passa da p-dimensioni a $2p$
  - Si adatta un classificatore con margine lineare soffice nello spazio di dimensioni $2p$
  - Questo risulta in un margine non lineare nello spazio a p dimensioni
- Esempio: Supponiamo di usare $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$ invece che solo (X_1, X_2).
  - La funzione che delimita la classificazione ha la forma nello spazio originale $\beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_1^2 + \beta_4X_2^2 + \beta_5X_1X_2 = 0$ (che sono le sezioni di una conica).

Uso di funzioni polinomiali

L'esempio in figura è ottenuto, nel pannello di sinistra, usando polinomiali cubiche. Da 2 variabili si passa a 9 e la soluzione nello spazio allargato risolve il problema nello spazio di dimensione originale.

Tuttavia come si vede nel pannello di destra si potrebbe cercare una soluzione ancora più flessibile e più efficace.



Support Vector Machines

Uso del prodotto scalare fra le osservazioni

- L'uso di polinomiali torna utile finchè p è basso. Ma è di difficile gestione al crescere di p
- Un modo per introdurre non linearità nei classificatori con vettori di supporto è usare i nuclei (**kernels**) che introducono un allargamento dello spazio delle covariate
- Per definire i kernels, si osservi che la classificazione con vettori di supporto può essere anche formulata facendo ricorso al prodotto scalare fra i vettori relativi alle osservazioni j e k

$$\langle x_j, x_k \rangle = \sum_{i=1}^{p} x_{ij} x_{ik}$$

(invece che le osservazioni stesse)

- il classificatore lineare con vettori di supporto può essere espresso anche come

$$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i \langle x, x_i \rangle$$

(i parametri sono tanti quante sono le osservazioni)

Uso del prodotto scalare fra le osservazioni

- $$f(x) = \beta_0 + \sum_{i=1}^{N} \alpha_i \langle x, x_k \rangle$$

  (i parametri sono tanti quante sono le osservazioni)
- I parametri α_i e β_0 sono stimati calcolando tutti i prodotti scalari $\langle x_j, x_k \rangle$ fra tutte le coppie osservazioni del training set (sono in tutto $\binom{N}{2}$)
- Tuttavia gli α_i risultano diversi da 0 solo per i vettori dell'insieme di supporto (per cui $\epsilon_i \neq 0$)
- Quindi definendo con S l'insieme dei vettori di supporto si l'espressione precedente si semplifica (in quanto coinvolge meno termini)

  $$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i \langle x_j, x_k \rangle$$
- In definitiva, per rappresentare il classificatore lineare $f(x)$ ci servono solo i prodotti scalari

Support Vector Machines e nuclei (kernels)

- Per quanto visto prima possiamo costruire un classificatore con vettori di supporto se calcoliamo un prodotto scalare fra le osservazioni.
- Questo può esser fatto anche, generalizzando il prodotto scalare e sostituendolo con alcune funzioni speciali dette *kernels* (nuclei). Ad esempio

$$K(x_j, x_k) = \left(1 + \sum_{i=1}^{p} x_{ij} x_{ik}\right)^d$$

  definisce un prodotto scalare per polinomi di grado d - che implica $\binom{p+d}{d}$ funzioni di base
- la soluzione ha la forma

$$f(x) = \beta_0 + \sum_{i \in S} \hat{\alpha}_i K(x, x_i)$$

- Quando il classificatore con vettori di supporto è abbinato a un kernel non lineare si parla di **Support Vector Machines**

Nuclei radiali

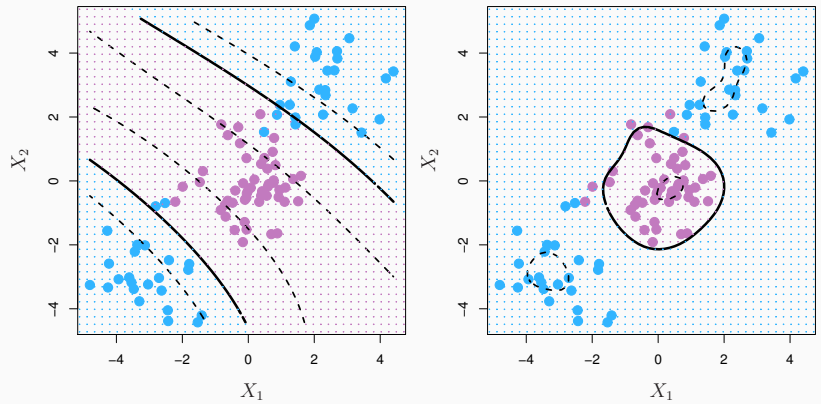
- Si può inoltre definire il nucleo radiale come

$$K(x_j, x_k) = exp(-\gamma \sum_{i=1}^{p} (x_{ij} - x_{ik})^2)$$

- con soluzione

$$f(x) = \beta_0 + \sum_{i \in S}^{N} \hat{\alpha}_i K(x, x_k)$$

La soluzione con kernel radiale permette di ottenere la curva di separazione nella figura di destra



Aspetti finali

Più di 2 classi

- Così com'è definita la support vector machine funziona con 2 classi. Cosa fare se abbiamo più classi?
- Strategia OVA (One versus All).
  - si costruiscono K differenti SVM classificatori a 2-classi $\hat{f}_k(x), k = 1, \ldots, K$
  - Si classifica un punto del test set x^* alla classe per cui which $\hat{f}_k(x^*)$ è più grande
- Strategia OVO (One versus One).
  - Si costruiscono $\binom{K}{2}$ classificatori per ciascuna coppia di classi $\hat{f}_{kl}(x)$
  - Si classifica x^* alla classe che vince in più coppie
- Se K non è troppo grande è preferibile OVO.

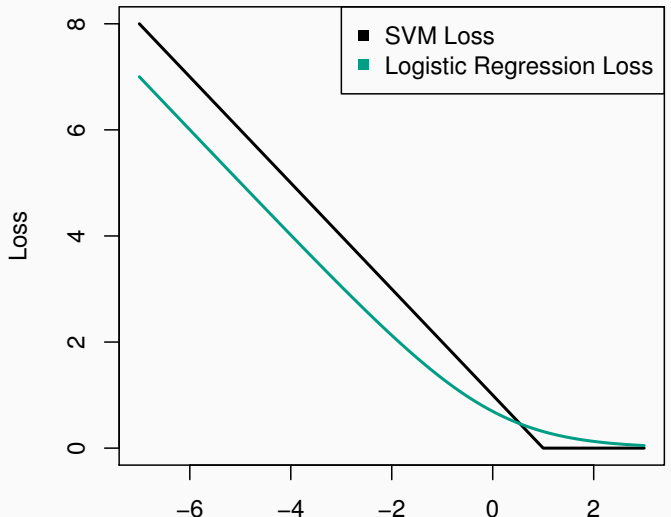
SVM e regressione logistica

Se $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$ un classificatore con vettori di supporto può essere riscritto come segue:

- cercare $\beta_0, \beta_1, \ldots, \beta_p$ per cui sia minima

$$\sum^{N} max[0, 1 - y_i f(x_i)] + \lambda \sum_{j=1}^{p} \beta_j^2$$

- questa espressione ha la forma *perdita + penalità*
- e questa funzione di perdita è nota come *perdita cerniera* (hinge loss)
- essa è simile alla perdita definita da -logverosimiglianza in un modello di regressione logistica



Usare SVM o Regressione Logistica?

- Se le classi sono (quasi perfettamente) separabili, SVM funziona meglio della regressione logistica (si ricordi che la regressione logistica non funziona, senza correttivi, in caso di perfetta separazione)
- Se non è questo il caso, allora la regressione logistica con penalizzazione tipo ridge fornisce risultati molto simili.
- Si noti che con SVM non si stimano le probabilità $P(Y = 1|X)$.
- L'uso di confini non lineari SVM con kernel (radiale o polinomiale) è molto popolare.
- L'idea dei kernels potrebbe essere estesa a regressione logistica e LDA, ma i calcoli sono molto più complessi (si usano tutte le osservazioni e non solo quelle dei vettori di supporto)
- Per quanto riguarda `R` esistono diverse funzioni per la classificazione con Support Vector machines: la più popolare è la funzione `svm()` nel pacchetto `e1071`