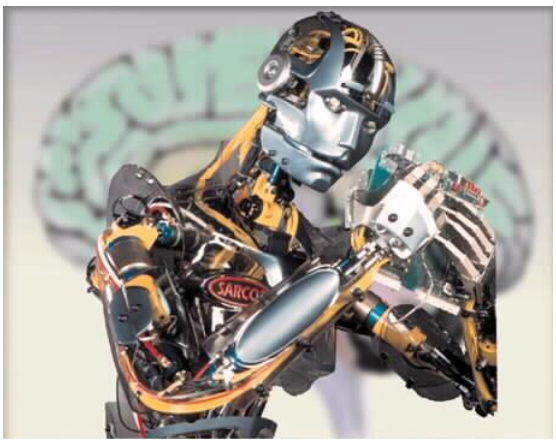
# Teoria di base

## Machine Learning



Il Machine Learning (noto anche come Apprendimento Automatico), sotto-area fondamentale dell'Intelligenza Artificiale, è una disciplina scientifica che si occupa dell'implementazione di algoritmi che permettono ai calcolatori (learners) di sviluppare delle decisioni intelligenti (behaviours) e di riconoscere in maniera automatica modelli complessi (Pattern Recognition) in base a dati empirici.

Più in dettaglio, i dati disponibili hanno il compito di illustrare le relazioni tra le variabili d'interesse, mentre ciò che viene richiesto al learner è sfruttare l'informazione fornita da dati stessi, al fine di rilevare le caratteristiche più importanti della loro distribuzione di probabilità.

La difficoltà dell'approccio risiede nella generalizzazione delle decisioni da intraprendere a partire dall'esperienza; l'obiettivo principale è infatti quello di sviluppare delle decisioni intelligenti anche in situazioni mai osservate prima (test data) a partire da una quantità piuttosto ridotta di casi noti a priori (training data).

Esistono in letteratura diversi tipi di algoritmi di Machine Learning, i più popolari sono:

- Apprendimento Supervisionato (Supervised Learning);
- Apprendimento non-Supervisionato (Unsupervised Learning);
- Apprendimento con rinforzo (Reinforcement Learning)

Nel seguito saranno trattati brevemente solo i primi due.

## Classificazione

Nell'ambito del Machine Learning e del Patterns Recognition, con il termine Classificazione ci si riferisce ad una procedura algoritmica che assegna ogni nuovo dato in ingresso (istanza) ad una delle categorie possibili (classi).

Se le classi di distinzione sono solo due, si parla di **classificazione binaria**, altrimenti si parla di **classificazione multi-classe**.

Più in dettaglio, ad ogni categoria corrisponde una etichetta diversa (label); l'algoritmo affida ad ogni istanza una label che indica semplicemente a quale classe appartiene il dato.

Una procedura in grado di esplicare tale funzione è denominata comunemente **classificatore**.

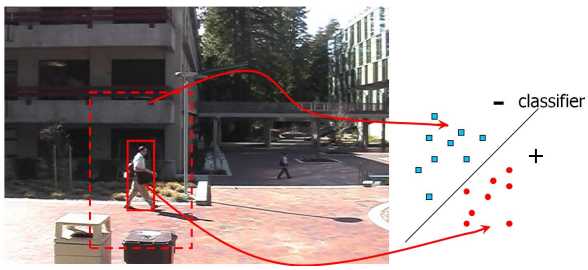



**Figura 1: esempio tipico di classificazione binaria, riconoscimento del foreground dal background**

L'istanza è formalmente descritta attraverso un vettore di caratteristiche (features) che insieme costituiscono una descrizione globale, ma al contempo riassuntiva, del dato da classificare.

Esistono numerosi tipi di features utilizzabili, nella pratica quelle più comunemente adottate sono: valori interi, valori reali, features nominali o categoriche, ordinali.

### Supervised - Unsupervised Learning

La classificazione normalmente si riferisce al riconoscimento supervisionato (Supervised Learning), ovvero ad una procedura che impara a classificare nuove istanze basandosi sul cosiddetto **Training Set**.

Quest'ultimo non è altro che un insieme di esempi (examples) scelti a priori; ogni example è una coppia formata da un'istanza e dalla corrispondente label, correttamente affibiata dall'operatore.

La fase di istruzione dell'algoritmo per l'ottenimento del classificatore desiderato, in base al training set, è denominata fase di Training.

I passi fondamentali da seguire per risolvere un problema di riconoscimento supervisionato sono:

- decisione del tipo di training examples;
- costruzione del training set che deve rappresentare al meglio il contesto reale e applicativo su cui si compirà la classificazione;
- rappresentazione della funzione di decisione nello spazio delle features;
- scelta della struttura del classificatore e del corrispondente algoritmo implementativo (SVM, Decision Trees, AdaBoost ecc);
- prova dell'algoritmo ottenuto sul training set e taratura di eventuali parametri di controllo tramite validazione;
- valutazione dell'accuratezza e delle prestazioni del classificatore mediante applicazione ad un insieme di nuove istanze (**Test set**).

L'Unsupervised Learning viene invece utilizzato nei casi in cui non si hanno a disposizione le labels per la fase di training.

In questo caso non si parla più di Classificazione ma di Clustering, ovvero suddivisione degli examples in sottoinsiemi disgiunti tali che gli esempi in uno stesso gruppo risultino molto simili e gli esempi in gruppi diversi abbastanza differenti, secondo i criteri posti dall'operatore.

Questo tipo di apprendimento presenta risultati peggiori rispetto al caso supervised, d'altra parte è l'unico a poter essere utilizzato in real time e a non richiedere necessariamente l'intervento dell'operatore umano, sia nella fase di training sia in occasione di classificazioni errate da parte dell'algoritmo, che andrebbe quindi corretto tramite un nuovo allenamento.

Il tipo di apprendimento utilizzato nella parte implementativa di questo progetto si presenta come un ibrido tra i due appena presentati, dato che riunisce i vantaggi del lavoro in real time, caratterizzante l'Unsupervised Learning, al periodo di training tipico del Supervised Learning, per il conseguimento di performance più elevate.

## Support Vector Machine

Tra le motivazioni che hanno portato alla scelta della tecnica di classificazione SVM, le principali sono:

- solida teoria di base;
- a differenza dei metodi di pattern recognition focalizzati sulla minimizzazione del rischio empirico, ovvero basati sulla minimizzazione degli errori di classificazione nel solo training set, l'SVM si propone di minimizzare il rischio strutturale e dunque la probabilità di classificare in maniera sbagliata anche i nuovi dati in ingresso, oltre a quelli di allenamento;
- l'SVM condensa tutta l'informazione contenuta nel training set in pochi punti essenziali per il learning, alleggerendo così il carico computazionale dell'algoritmo implementativo.

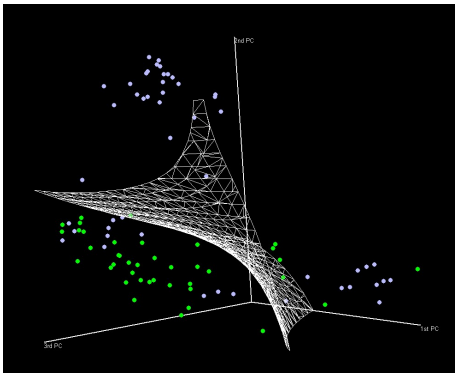


**Figura 2: esempio di classificazione binaria SVM in $\mathbb{R}^3$**

Senza perdita di generalità si considererà d'ora in avanti solo il problema di classificazione binaria, in un contesto di tipo supervisionato.

Inoltre saranno trattati separatamente il caso di dati linearmente separabili e la generalizzazione al caso non linearmente separabile, entrambi risolti mediante tecnica SVM lineare.

Per quanto riguarda l'utilizzo di funzioni di decisione non-lineari (SVM non-lineare) si rimanda all'appendice A.

### Linear SVM: Caso Linearmente Separabile

Sia dato il Training Set T, composto dai punti $\mathbf{x_i} \in \mathbb{R}^N$, spazio delle features, $\forall i = 1, 2, \ldots, N$.

Ogni punto $\mathbf{x_i}$ è etichettato mediante una label $y_i \in \{-1, 1\}$, a seconda dell'appartenenza ad una delle due classi.

L'obiettivo della classificazione è ricavare l'equazione dell'iperpiano (funzione decisionale lineare) che suddivida l'insieme T, lasciando tutti i punti afferenti alla stessa classe dalla stessa parte dello spazio delle features e contemporaneamente che massimizzi la distanza tra le classi e l'iperpiano stesso (iperpiano ottimo di separazione).

Il training set T si dice Linearmente Separabile se

$$\exists\, \boldsymbol{\beta} \in \mathbb{R}^N \quad e \quad \beta_0 \in \mathbb{R} \quad \text{t.c.}$$

$$y_i(\boldsymbol{\beta} \cdot \mathbf{x_i} + \beta_0) \geq 1, \quad \forall i = 1, \ldots, N. \tag{1}$$

La coppia $(\boldsymbol{\beta},\ \beta_0)$ definisce un iperpiano di equazione

$$f(\mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{x} + \beta_0 = 0 \tag{2}$$

denominato iperpiano di separazione.

Ponendo $\beta = \| \boldsymbol{\beta} \|$, la distanza con segno $d_i$ del punto $\mathbf{x_i}$ dall'iperpiano (2) è data da

$$d_i = \frac{\boldsymbol{\beta} \cdot \mathbf{x_i} + \beta_0}{\beta} \tag{3}$$

Dalle relazioni (1) e (3) si ottiene

$$y_i d_i \geq \frac{1}{\beta}, \quad \forall\ i = 1, \ldots, N \tag{4}$$

Si nota dunque che $1/\beta$ è il limite inferiore per la distanza tra i punti $\mathbf{x_i}$ e l'iperpiano di separazione.

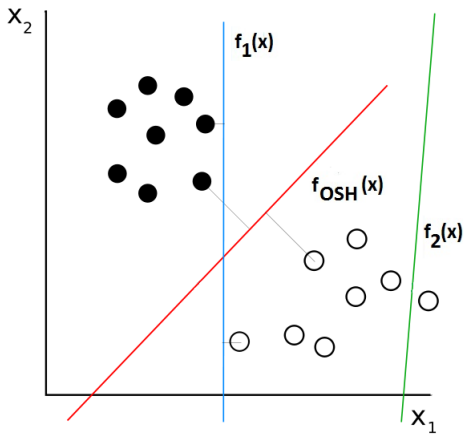



**Figura 3: spazio delle features ($\mathbb{R}^2$) con vari iperpiani di separazione, la decisione ottima è in rosso.**

È importante a questo punto introdurre la nozione di rappresentazione parametrica dell'iperpiano di separazione.

Dato un iperpiano di separazione identificato dalla coppia $(\boldsymbol{\beta},\ \beta_0)$ per il training set T linearmente separabile, la rappresentazione canonica dell'iperpiano si ottiene riscalando la coppia $(\boldsymbol{\beta},\ \beta_0)$ nella coppia $(\boldsymbol{\beta}',\ \beta_0')$, in modo che la distanza dei punti $\mathbf{x_j}$ più vicini all'iperpiano sia esattamente $1/\beta'$, ovvero in modo che

$$\min_{\mathbf{x_j} \in T} \{y_i(\boldsymbol{\beta}' \cdot \mathbf{x_j} + \beta_0')\} = 1. \tag{5}$$

Per semplicità, d'ora in poi assumeremo sempre l'iperpiano di separazione già in forma canonica, ovvero $\boldsymbol{\beta} = \boldsymbol{\beta}'$ e $\beta_0 = \beta_0'$.

Dato un T linearmente separabile, l'obiettivo di questa trattazione teorica è ricavare l'equazione dell'OSH (Optimal Separating Hyperplane), ovvero l'iperpiano di separazione che massimizza la distanza dai punti di T a lui più vicini (si vedano la figure (4) e (5) ).

In altre parole l'iperpiano ottimo, identificato dalla coppia $(\bar{\boldsymbol{\beta}},\ \bar{\beta}_0)$, è la soluzione del problema vincolato seguente:

$$\begin{aligned} &\text{Minimizzare} && \tfrac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 \\ &\text{soggetto a} && y_i(\boldsymbol{\beta} \cdot \mathbf{x_i} + \beta_0) \geq 1, \quad \forall i \end{aligned}$$

detto Problema Primale.

La quantità $2/\beta$ è detta margine, essendo il limite inferiore della minima distanza tra i punti di classi diverse (si veda la figura.

Questa quantità rappresenta approssimativamente il grado di difficoltà del problema; più il margine è stretto e più il problema è difficile da risolvere.

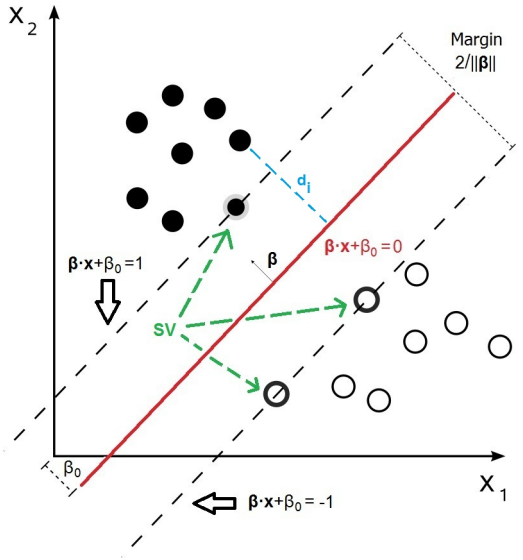



**Figura 4: Iperpiano ottimo e margine relativo in $\mathbb{R}^2$, caso linearmente separabile**

Il problema primale si risolve solitamente con il Metodo dei Moltiplicatori di Lagrange.

Denotando con $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ il vettore degli N moltiplicatori di Lagrange associati ai vincoli dati da (1), risolvere il problema primale equivale a trovare il punto di sella della funzione lagrangiana

$$L(\boldsymbol{\beta},\ \beta_0,\ \boldsymbol{\alpha}) = \frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 - \sum_{i=1}^{N} \alpha_i \{y_i(\boldsymbol{\beta} \cdot \mathbf{x_i} + \beta_0) - 1\} \tag{6}$$

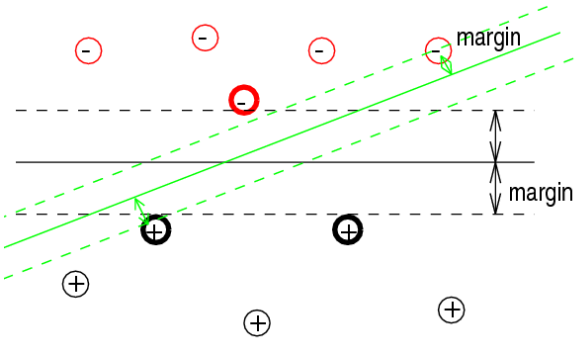



**Figura 5: l'iperpiano ottimo di decisione è associato al margine più ampio**

Nel punto di sella L presenta un minimo in $\boldsymbol{\beta} = \bar{\boldsymbol{\beta}}$ e $\beta_0 = \bar{\beta}_0$ ed un massimo in $\boldsymbol{\alpha} = \bar{\boldsymbol{\alpha}}$; dunque:

$$\frac{\partial L(\boldsymbol{\beta},\ \beta_0,\ \boldsymbol{\alpha})}{\partial \beta_0} = \sum_{i=1}^{N} y_i \alpha_i = 0 \qquad (7)$$

$$\frac{\partial L(\boldsymbol{\beta},\ \beta_0,\ \boldsymbol{\alpha})}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{i=1}^{N} y_i \alpha_i \boldsymbol{x_i} = 0 \qquad (8)$$

Una volta sostituite le equazioni (7) e (8) nella (6), si nota più chiaramente che risolvere il problema primale equivale a risolvere il seguente problema vincolato, detto Problema Duale:

$$\begin{aligned} &\text{Minimizzare} && -\tfrac{1}{2}\boldsymbol{\alpha}^T D \boldsymbol{\alpha} + \textstyle\sum_{i=1}^{N} \alpha_i \\ &\text{soggetto a} && \textstyle\sum_{i=1}^{N} y_i \alpha_i = 0 \\ & && \alpha_i \geq 0 \quad \forall i \end{aligned}$$

dove $D \in \mathbb{R}^{NxN}$ e t.c. $[D]_{ij} = y_i y_j \mathbf{x_i} \cdot \mathbf{x_j}$.

Dalla (8) si vede immediatamente che

$$\bar{\boldsymbol{\beta}} = \sum_{i=1}^{N} \bar{\alpha}_i y_i \mathbf{x_i}, \qquad (9)$$

mentre $\bar{\boldsymbol{\beta}}_0$ si può determinare da $\bar{\boldsymbol{\alpha}}$, soluzione del problema duale, e dalle condizioni di Karush-Kühn-Tucker (KKT)

$$\bar{\alpha}_i \{ y_i (\bar{\boldsymbol{\beta}} \cdot \mathbf{x_i} + \bar{\beta}_0) - 1 \} = 0, \qquad i = 1, \ldots, N. \qquad (10)$$

È molto importante osservare che gli unici $\bar{\alpha}_i$ non-nulli nella (8) sono quelli per cui i vincoli (1) sono soddisfatti con l'eguaglianza. Conseguenza di ciò è che il vettore ottimo $\bar{\boldsymbol{\beta}}$ è una combinazione lineare di un numero relativamente basso dei punti $\mathbf{x_i}$.

Questi punti sono chiamati Support Vector (SV), essendo i punti più vicini all'iperpiano ottimo di separazione ed anche gli unici punti del training set T indispensabili a determinare l'OSH stesso (si rimanda alla figura (4)).

Infine dati due qualunque SV, $\mathbf{x_r}$ e $\mathbf{x_s}$, tali che $y_r = 1$, $y_s = -1$ e $\bar{\alpha}_s$, $\bar{\alpha}_r \geq 0$, il parametro $\bar{\beta}_0$ si può ottenere nel modo seguente:

$$\bar{\beta}_0 = -\frac{1}{2} \bar{\boldsymbol{\beta}} \cdot (\mathbf{x_r} + \mathbf{x_s}) \qquad (11)$$

In conclusione il problema di classificazione di un nuovo dato, $\mathbf{x}$, si riduce all'osservazione del segno di

$$f_{OSH}(\mathbf{x}) = \bar{\boldsymbol{\beta}} \cdot \mathbf{x} + \bar{\beta}_0$$

Dunque i SV condensano tutta l'informazione, contenuta in T, necessaria a classificare i nuovi dati.

### Linear SVM: Caso Non-linearmente Separabile

Il caso di classificazione con training set non linearmente separabile si può affrontare generalizzando quanto visto fin'ora.

Si introducono, infatti, N varibili non-negative $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)$ in modo che

$$y_i(\boldsymbol{\beta}\cdot\mathbf{x_i}+\beta_0) \geq 1-\xi_i, \qquad i = 1, 2, \ldots, N \quad (12)$$

L'inserimento di queste variabili di slack permette di considerare anche possibili classificazioni errate.

L'iperpiano ottimo di separazione generalizzato è allora la soluzione del Problema Primale generalizzato:

$$\begin{array}{ll} \text{Minimizzare} & -\frac{1}{2} \parallel \boldsymbol{\beta} \parallel^2 + \gamma \sum_{i=1}^{N} \xi_i \\ \text{soggetto a} & y_i(\boldsymbol{\beta} \cdot \mathbf{x_i} + \beta_0) \geq 1 - \xi_i, \ \forall i \\ & \xi_N \geq 0, \quad \forall i \end{array}$$

Il termine aggiuntivo $\gamma \sum_{i=1}^{N} \xi_i$ rende l'OSH meno sensibile all'eventuale presenza di outliers nel training set, mentre il parametro $\gamma$ si può interpretare come un parametro di regolarizzazione.

Infatti l'iperpiano ottimo tende a massimizzare il margine per bassi valori di $\gamma$ e a minimizzare il numero di classificazioni errate per valori di $\gamma$ elevati.

Anche in questo caso si può trasformare il problema primale nel corrispondente Problema Duale generalizzato:

$$\begin{array}{ll} \text{Minimizzare} & -\frac{1}{2}\boldsymbol{\alpha}^T D \boldsymbol{\alpha} + \sum_{i=1}^{N} \alpha_i \\ \text{soggetto a} & \sum_{i=1}^{N} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq \gamma, \quad \forall i \end{array}$$

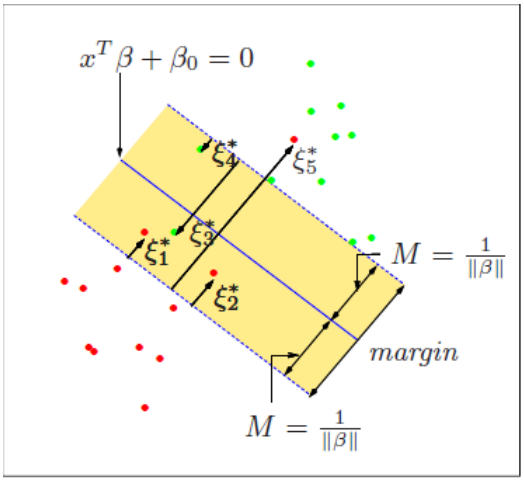



**Figura 6: iperpiano ottimo di decisione per training set non linearmente separabile**

con D stessa matrice del caso linearmente separabile.

Dalla soluzione ottima $\bar{\boldsymbol{\alpha}}$ del problema duale generalizzato, si ricava facilmente

$$\bar{\boldsymbol{\beta}} = \sum_{i=1}^{N} \bar{\alpha}_i y_i \mathbf{x_i}, \qquad (13)$$

mentre $\bar{\beta}_0$ si può determinare dalle condizioni di KKT:

$$\begin{aligned} \bar{\alpha}_i\{y_i(\bar{\boldsymbol{\beta}} \cdot \mathbf{x_i} + \beta_0) - 1 + \bar{\xi}_i\} &= 0 \ \forall i \ (14) \\ (\gamma - \bar{\alpha}_i)\bar{x}i_i &= 0 \ \forall i \ (15) \end{aligned}$$

dove $\bar{\xi}$ è il valore di $\xi$ calcolato nel punto di sella.

Come in precedenza, i punti $\mathbf{x_i}$ a cui corrispondono $\bar{\alpha}_i \geq 0$ sono chiamati Support Vector.

In questo contesto bisogna fare una distinzione ulteriore tra i SV corrispondenti ad $\bar{\alpha}_i \leq \gamma$ e quelli per cui $\bar{\alpha}_i = \gamma$.

Nel primo caso, dalla condizione (15) segue che $\bar{\xi}_i = 0$, e dunque, dalla (14), che i support vectors giacciono esattamente sul margine (Margin Vectors).

Invece i SV connessi ad $\bar{\alpha}_i = \gamma$ (chiamati generalmente errori) sono punti:

- classificati non correttamente, se $\xi > 1$;
- classificati correttamente ma all'interno del margine dell'OSH
  se $0 < \xi < 1$;
- Margin Vectors, se $\xi = 0$.

Infine tutti i punti che non sono SV sono classificati correttamente e giacciono al di fuori del margine.

## Ulteriore sviluppo teorico

Il problema di classificazione fin'ora affrontato si può equivalentemente riformulare utilizzando una funzione di penalità (detta Loss Function):

$$\sum_{i=1}^{N} [1 - y_i f(x_i)]_+ + \lambda \parallel \boldsymbol{\beta} \parallel^2$$

dove $L(y_i, f(x_i)) := [1 - y_i f(x_i)]_+$ è la Hinge Loss Function, che pesa l'errore compiuto sulla classificazione di ogni punto $x_i$, mentre il termine quadratico finale è un termine di regolarizzazione (penalità quadratica) e $\lambda = \frac{1}{\gamma}$.

Il contributo apportato da questo progetto allo stato dell'arte consiste nell'implementazione di un algoritmo SVM basato sul Sequential Quadratic Programming (SQP).

L'algoritmo SQP qui utilizzato può essere considerato come una versione semplificata del filtro di Kalman iterato (IKF), dato che non necessita di un modello di misura probabilistico $p(y|x)$, di difficile realizzazione in questo contesto.

Per quanto appena riportato, è necessario svolgere una linearizzazione del primo addendo della funzione () fino al secondo ordine, dato che quest'ultimo non è una quadratica in $\boldsymbol{\beta}$ e $\beta_0$. Occorre però, per prima cosa, rendere questo addendo differenziabile; per questo si è deciso di sostituire la Hinge Loss function con la Binomial Deviance:

$$L(y_i, f(x_i)) = log[1 + e^{-y_i f(x_i)}] \tag{16}$$

e infine linearizzare il nuovo funzionale di costo:

$$\sum_{i=1}^{N} log[1 + e^{-y_i f(x_i)}] + \lambda \parallel \boldsymbol{\beta} \parallel^2 \tag{17}$$

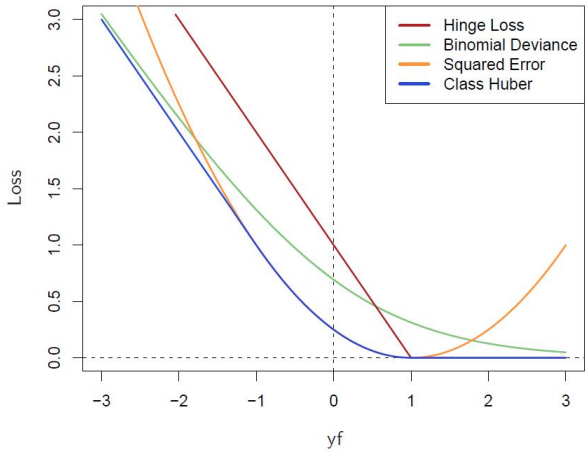



**Figura 7: funzione di penalità Hinge Loss e sua approssimazione differenziabile Binomial Deviance**

Ciò che rimane da inserire a questo punto è la dinamica dell'iperpiano ottimo di separazione, in modo da trasformare il problema di ottimizzazione da statico in dinamico, ovvero da risolvere ad ogni istante di tempo t:

$$\min_{\boldsymbol{\beta}_t,\ \beta_{0_t}} \sum_{i=1}^{N} log[1 + e^{-y_i f(x_i)}] + \lambda \parallel \boldsymbol{\beta_t} \parallel^2 \tag{18}$$

Il modello non-lineare utilizzato per rappresentare la dinamica su $\boldsymbol{\beta}$ è il modello random walk seguente:

$$\begin{bmatrix} \boldsymbol{\beta}_{t+1} \\ \beta_{0_{t+1}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\beta}_t \\ \beta_{0_t} \end{bmatrix} + \begin{bmatrix} w_t \\ w_{0_t} \end{bmatrix} \tag{19}$$

dove $w_t \in \mathbb{R}^N$ e $w_{0_t} \in \mathbb{R}$ sono rumori gaussiani bianchi tra loro scorrelati e

$$Q = \begin{bmatrix} Var[w_t] & \mathbf{0} \\ \mathbf{0} & var[w_{0_t}] \end{bmatrix}$$