



UNIVERSITÀ
DEGLI STUDI
DI PALERMO



Classificatori

CORSO DI BIG DATA
a.a. 2019/2020

Prof. Roberto Pirrone

Sommario

- Generalità
- Selezione delle feature
- Decision Tree
- Random Forests
- Classificatori probabilistici
 - Regressione logistica
 - Naive Bayes
- Support Vector Machine
- Valutazione della bontà della classificazione

Generalità

- Dato un insieme di dati di addestramento, ciascuno associato ad una «etichetta» che individua l'appartenenza del dato ad una classe, *la classificazione consiste nel predire il valore dell'etichetta per dati di test mai visti dall'algoritmo*
- È una classe di algoritmi di apprendimento supervisionati
 - Le etichette provengono da una associazione artificiale (dipendente dall'applicazione) ai dati e non dalla naturale tendenza di questi ultimi a formare cluster
 - È, forse, la tipologia di algoritmo di ML più comune

Generalità

- In generale, dato un insieme di n punti in \mathbb{R}^d appartenenti ad un dataset \mathcal{D} , questi vengono associati ad un insieme di etichette in $\{1, \dots, k\}$
 - Spesso la classificazione è binaria: $\{0, 1\}$ ovvero $\{-1, 1\}$
- Due modalità di funzionamento
 - Predizione esplicita dell'etichetta
 - Score numerico (probabilità) di appartenenza del punto ad una certa classe

Selezione delle feature

- Filtri: indicatori numerici della rilevanza delle feature
- Modelli «wrapped»: un algoritmo di classificazione viene usato per valutare la performance su un sotto-insieme di feature e quindi «avvolge» il vero e proprio algoritmo di classificazione in uno schema di ricerca delle feature rilevanti
- Modelli «embedded»: l'algoritmo stesso fornisce indicazioni sulle feature rilevanti e, dopo averle individuate, viene riaddestrato solo su di esse

Selezione delle feature

- Filtri

- Gini index

- Entropia

$$G(v_i) = 1 - \sum_{j=1}^k p_j^2$$

Valore i-esimo dell'attributo

Frazione dei punti che hanno il valore v_i nella classe j

$$G = \sum_{i=1}^r n_i G(v_i) / n$$

Frazione dei punti che hanno il valore v_i

$$E(v_i) = - \sum_{j=1}^k p_j \log_2(p_j), \quad E = \sum_{i=1}^r n_i E(v_i) / n$$

Selezione delle feature

- Filtri

- Fisher score

$$F = \frac{\sum_{j=1}^k p_j (\mu_j - \mu)^2}{\sum_{j=1}^k p_j \sigma_j^2}$$

- Fisher Linear Discriminant

- Generalizzazione del Fisher Score per combinazioni lineari di feature
 - Tende a trovare, in forma supervisionata, la direzione di massima variazione delle feature e, per contro, l'iperpiano perpendicolare che separa meglio le classi rispetto alle feature stesse

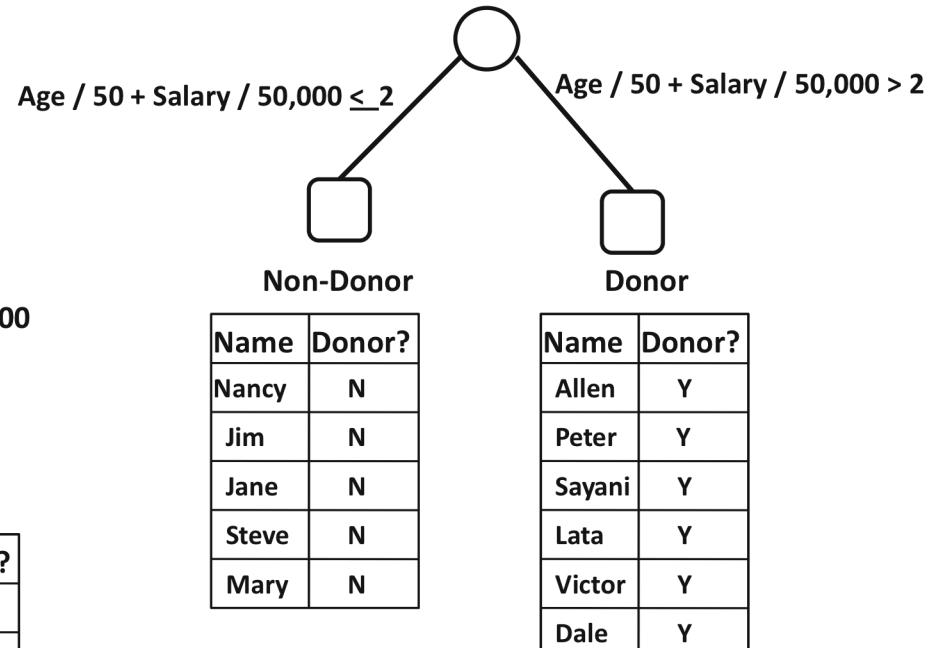
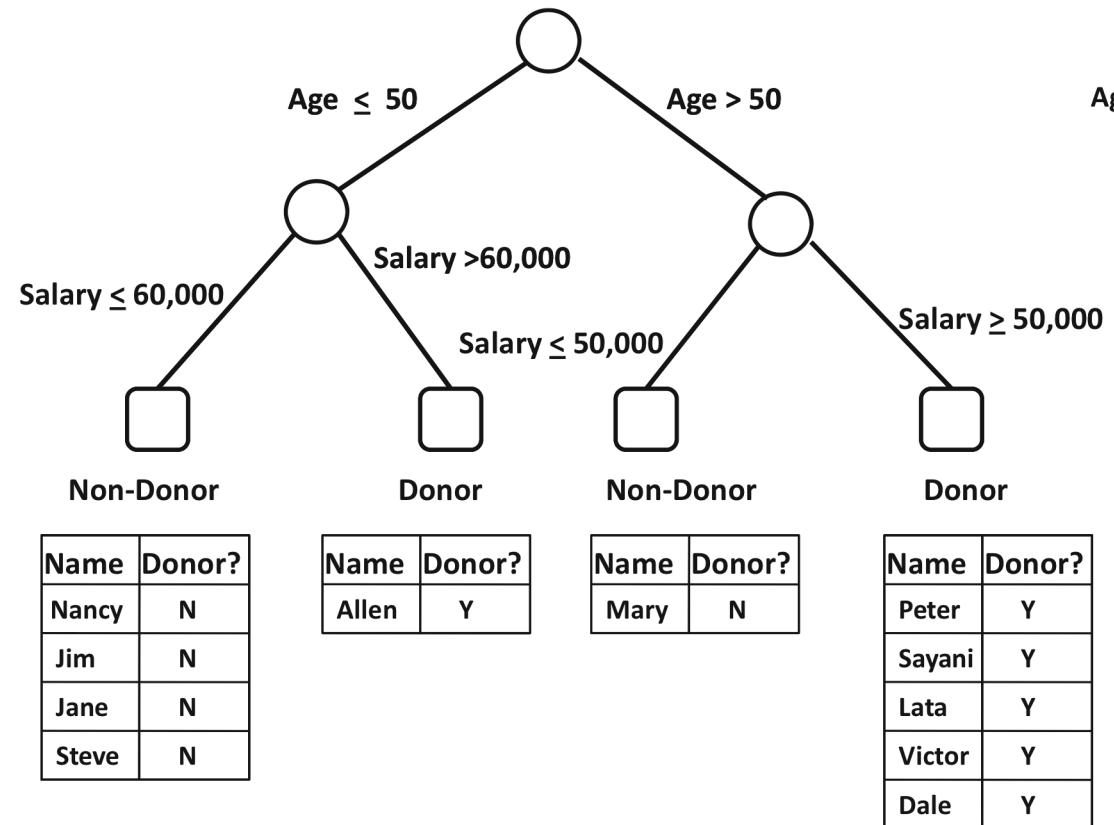
Selezione delle feature

- Modelli «wrapped»
 - Si parte da un insieme di feature $F = \{\}$
 - Si aggiungono feature a F e si testa l'accuratezza dell'algoritmo \mathcal{A} per accettare l'aggiunta delle nuova feature a F
 - L'incremento di F si può fare secondo diverse strategie
 - Random
 - Aggiunta della feature con maggior potere discriminativo rispetto ad un criterio di filtro

Decision Tree

- Il data set viene suddiviso ricorsivamente in parti più piccole sulla base del discriminare sui valori degli attributi
- I nodi foglia dell'albero vengono attribuiti alla classe dominante
 - Gerarchico similmente al clustering
 - Split univariato → decisione su un solo attributo
 - Split multivariato → decisione su un insieme di attributi

Decision tree



Decision tree

Algorithm *GenericDecisionTree*(Data Set: \mathcal{D})

begin

 Create root node containing \mathcal{D} ;

repeat

 Select an eligible node in the tree;

 Split the selected node into two or more nodes
 based on a pre-defined split criterion;

until no more eligible nodes for split;

 Prune overfitting nodes from tree;

 Label each leaf node with its dominant class;

end

$$G(S) = 1 - \sum_{j=1}^k p_j^2, \quad \text{Gini-Split } (S \Rightarrow S_1 \dots S_r) = \sum_{i=1}^r \frac{|S_i|}{|S|} G(S_i)$$

Suddivisione del data set in r sottoinsiemi
per uno split a r vie

- Split binario
- Split a r vie per attributi categorici
- Split binario per attributi categorici binarizzati
- Error rate: la frazione di elementi che non appartengono alla classe dominante
- Gini index
- Entropia

Decision Tree

- In genere i nodi vicini ai nodi foglia lavorano su pochi dati e sono proni al rumore nell'applicare lo split
 - Tendenza al overfitting
 - Si utilizzano tecniche di stop che prediligono alberi poco profondi
- Si utilizzano tecniche di pruning dei nodi foglia che vanno in overfit
 - Si valida l'eventuale incremento dell'accuracy su un validation set espunto dai dati di addestramento

Random Forests

- È un ensemble di classificatori decision tree
 - Nei metodi di ensemble, più classificatori vengono addestrati sul data set e i risultati sono combinati tra loro per ottenere una predizione più robusta
- Uso del «bagging»
 - Tecnica per creare un ensemble di classificatori i.i.d. in modo tale da ridurre la varianza della stima da σ^2 a σ^2/k
 - Si generano k data set con campionamento con rimpiazzo: i nuovi data set contengono duplicati
 - Si addestrano i classificatori e la predizione è data dalla maggioranza o dalla media dei voti espressi da ciascuno su ogni campione

Random Forests

- I nodi più elevati della gerarchia sono sostanzialmente invarianti
- È necessario diversificare gli alberi per aumentare la capacità di predizione
- L'insieme dei decision tree usa un criterio random per lo split nei diversi alberi dell'ensemble

Random Forests

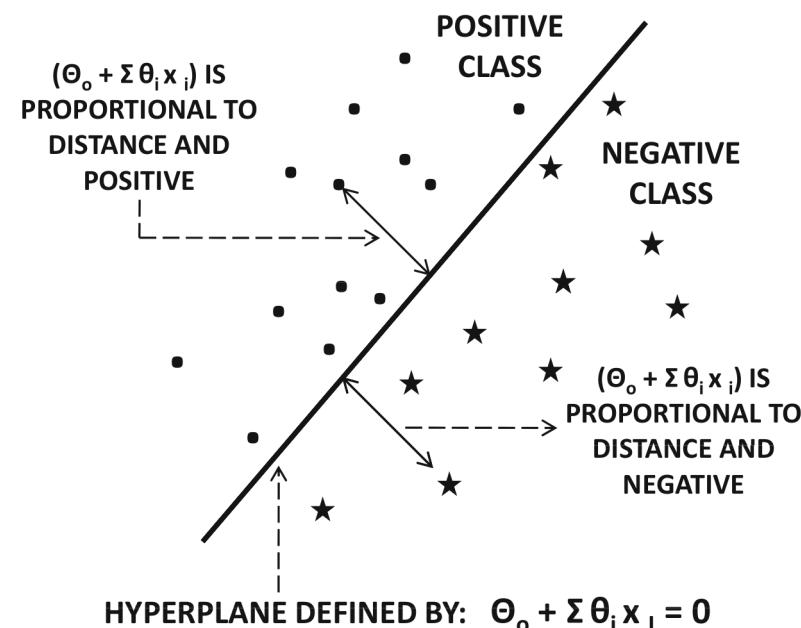
- Random split selection
 - Forest-RI: Un sottoinsieme casuale di q feature viene estratto per lo split
 - Forest-RC: un sottoinsieme L di feature viene estratto e si creano q combinazioni lineari che vengono usate come feature multivariate per lo split

Regressione logistica

$$P(C = +1 | \bar{X}) = \frac{1}{1 + e^{-(\theta_0 + \sum_{i=1}^d \theta_i x_i)}}$$

$$P(C = -1 | \bar{X}) = \frac{1}{1 + e^{(\theta_0 + \sum_{i=1}^d \theta_i x_i)}}$$

$$P(C = +1) \equiv P(C = -1) = 0.5 \rightarrow \theta_0 + \sum_{i=1}^d \theta_i x_i = 0$$



Regressione logistica

$$\mathcal{LL}(\bar{\Theta}) = \log(\mathcal{L}(\bar{\Theta})) = - \sum_{\bar{X}_k \in \mathcal{D}_+} \log \left(1 + e^{-\left(\theta_0 + \sum_{i=1}^d \theta_i x_k^i \right)} \right) - \sum_{\bar{X}_k \in \mathcal{D}_-} \log \left(1 + e^{\left(\theta_0 + \sum_{i=1}^d \theta_i x_k^i \right)} \right)$$

$$\begin{aligned} \frac{\partial \mathcal{LL}(\bar{\Theta})}{\partial \theta_i} &= \sum_{\bar{X}_k \in \mathcal{D}_+} \frac{x_k^i}{1 + e^{\left(\theta_0 + \sum_{i=1}^d \theta_i x_i \right)}} - \sum_{\bar{X}_k \in \mathcal{D}_-} \frac{x_k^i}{1 + e^{-\left(\theta_0 + \sum_{i=1}^d \theta_i x_i \right)}} \\ &= \sum_{\bar{X}_k \in \mathcal{D}_+} P(\overline{X_k} \in \mathcal{D}_-) x_k^i - \sum_{\bar{X}_k \in \mathcal{D}_-} P(\overline{X_k} \in \mathcal{D}_+) x_k^i \end{aligned}$$

$$\theta_i \leftarrow \theta_i + \alpha \left(\sum_{\overline{X}_k \in \mathcal{D}_+} P(\overline{X_k} \in \mathcal{D}_-) x_k^i - \sum_{\overline{X}_k \in \mathcal{D}_-} P(\overline{X_k} \in \mathcal{D}_+) x_k^i \right)$$

Addestramento con
gradient ascent e termine di
regolarizzazione

$$-\lambda \sum_{i=1}^d \theta_i^2 / 2$$

Naive Bayes

$$P(C = c|x_1 = a_1, \dots, x_d = a_d) = \frac{P(C = c)P(x_1 = a_1, \dots, x_d = a_d|C = c)}{P(x_1 = a_1, \dots, x_d = a_d)}$$
$$\propto P(C = c)P(x_1 = a_1, \dots, x_d = a_d|C = c)$$

$$P(x_1 = a_1, \dots, x_d = a_d|C = c) = \prod_{j=1}^d P(x_j = a_j|C = c)$$

$$P(C = c|x_1 = a_1, \dots, x_d = a_d) \propto P(C = c) \prod_{j=1}^d P(x_j = a_j|C = c)$$

$$P(x_j = a_j|C = c) = \frac{q(a_j, c) + \alpha}{r(c) + \alpha \cdot m_j}$$

Si assume una distribuzione di Bernoulli sui dati
Addestramento con EM
In fase di test fornisce probabilità

Laplacian smoothing: m_j è il
numero di valori distinti di a_j

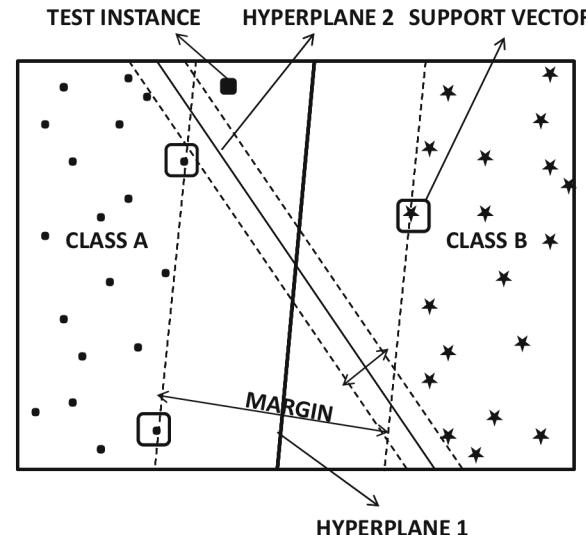
Support Vector Machines

$$\overline{W} \cdot \overline{X_i} + b \geq 0 \quad \forall i : y_i = +1$$

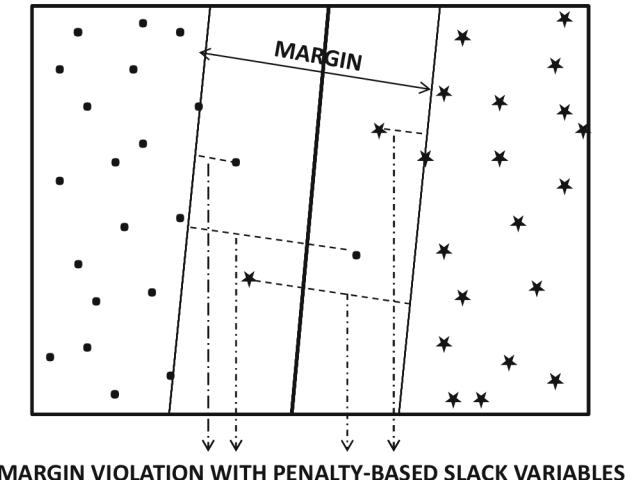
$$\overline{W} \cdot \overline{X_i} + b \leq 0 \quad \forall i : y_i = -1$$

Con un apposito scaling di \overline{W} e b :

$$\begin{cases} \overline{W} \cdot \overline{X_i} + b \geq +1 \forall i : y_i = +1 \\ \overline{W} \cdot \overline{X_i} + b \leq -1 \forall i : y_i = -1 \end{cases} \Rightarrow y_i(\overline{W} \cdot \overline{X_i} + b) \geq +1, \forall i$$



(a) Hard separation



(b) Soft separation

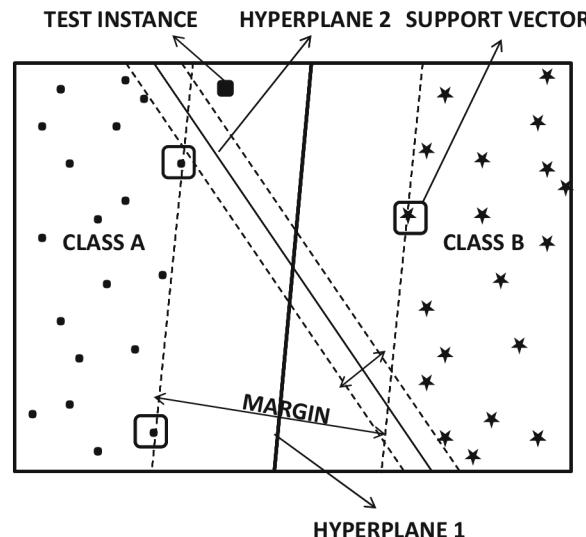
Support Vector Machines

distanza iperpiani : $2/\|\bar{W}\|$

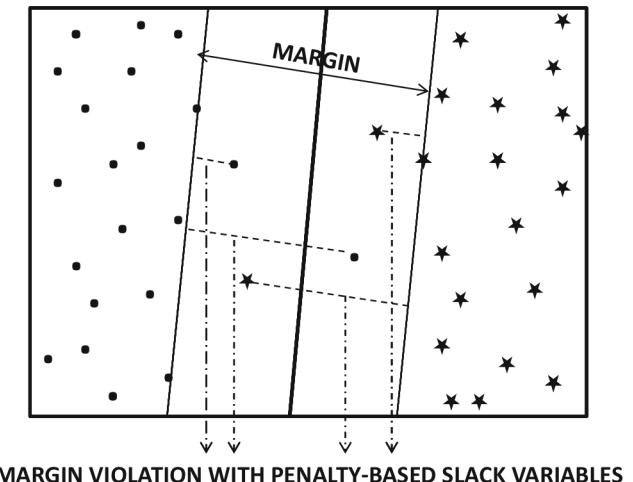
funzione obiettivo : $\|\bar{W}\|^2/2$

$$L_P = \frac{\|\bar{W}\|^2}{2} - \sum_{i=1}^n \lambda_i [y_i(\bar{W} \cdot \bar{X}_i + b) - 1], \quad \lambda_i \geq 0$$

support vectors : $\lambda_i [y_i(\bar{W} \cdot \bar{X}_i + b) - 1] = 0$



(a) Hard separation



(b) Soft separation

Support Vector Machines

$$\nabla L_P = \nabla \frac{\|\overline{W}\|^2}{2} - \nabla \sum_{i=1}^n \lambda_i [y_i(\overline{W} \cdot \overline{X}_i + b) - 1] = 0$$

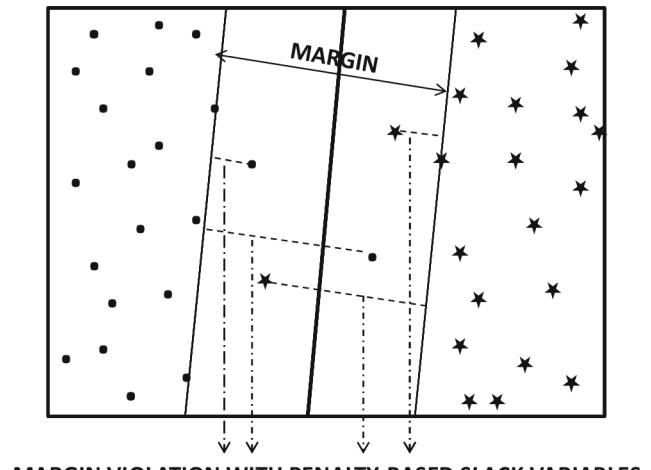
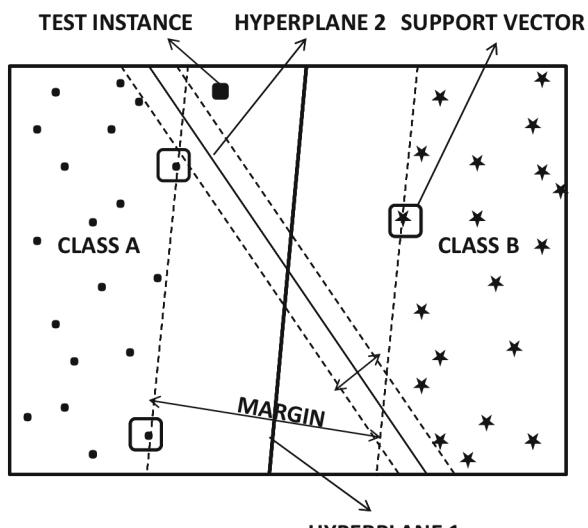
$$\overline{W} = \sum_{i=1}^n \lambda_i y_i \overline{X}_i, \quad \sum_{i=1}^n \lambda_i y_i = 0$$

formulazione del problema duale di massimizzazione :

$$L_D = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \overline{X}_i \cdot \overline{X}_j$$

addestramento lungo il gradiente :

$$\frac{\partial L_D}{\partial \lambda_i} = 1 - y_i \sum_{j=1}^n y_j \lambda_j \overline{X}_i \cdot \overline{X}_j, \quad (\lambda_1 \dots \lambda_n) \leftarrow (\lambda_1 \dots \lambda_n) + \alpha \left(\frac{\partial L_D}{\partial \lambda_1} \dots \frac{\partial L_D}{\partial \lambda_n} \right)$$



Support Vector Machines

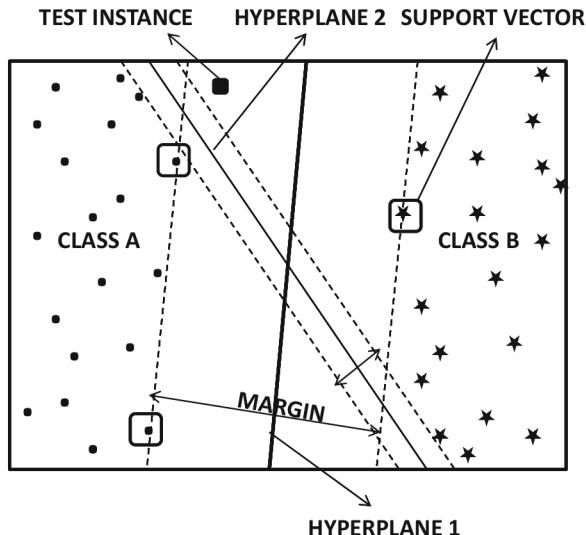
$$\begin{cases} \overline{W} \cdot \overline{X_i} + b \geq +1 - \xi_i & \forall i : y_i = +1 \\ \overline{W} \cdot \overline{X_i} + b \leq -1 + \xi_i & \forall i : y_i = -1 \end{cases}, \quad \xi_i \geq 0 \quad \forall i$$

funzione obiettivo :

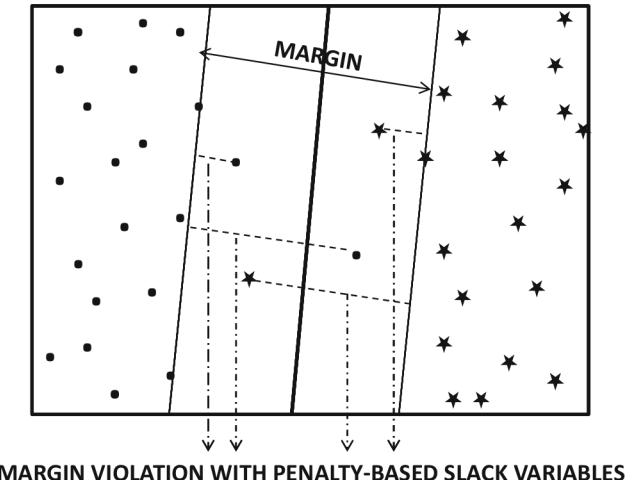
$$O = \frac{\|\overline{W}\|^2}{2} + C \sum_{i=1}^n \xi_i$$

Lagrange penalty :

$$L_P = \frac{\|\overline{W}\|^2}{2} + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \lambda_i [y_i(\overline{W} \cdot \overline{X_i} + b) - 1 + \xi_i] - \sum_{i=1}^n \beta_i \xi_i$$



(a) Hard separation



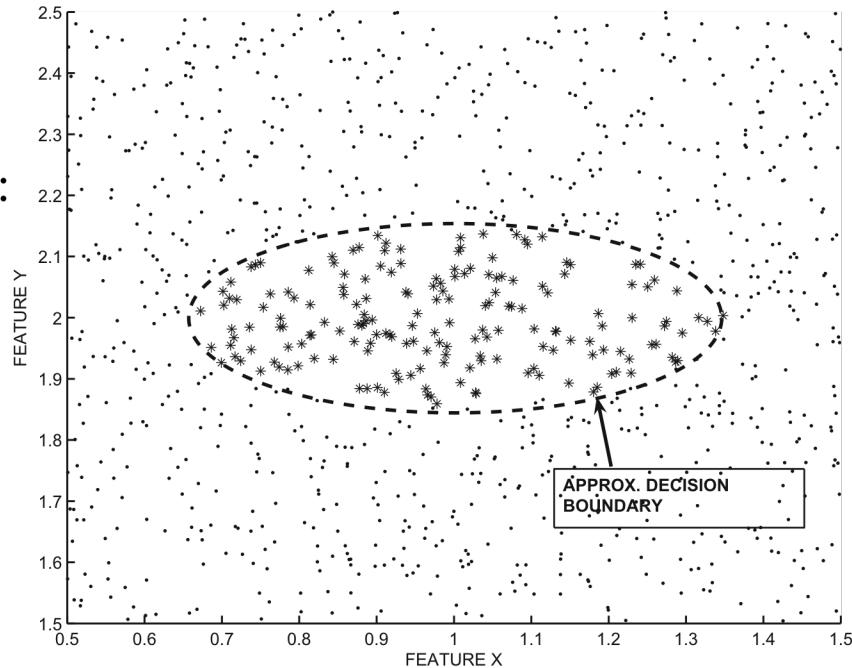
(b) Soft separation

Support Vector Machines

kernel trick per superfici di separazione non lineari :

$$L_D = \sum_{i=1}^n \lambda_i - \frac{1}{2} \cdot \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j K(\overline{X}_i, \overline{X}_j)$$

Function	Form
Gaussian radial basis kernel	$K(\overline{X}_i, \overline{X}_j) = e^{- \overline{X}_i - \overline{X}_j ^2 / 2\sigma^2}$
Polynomial kernel	$K(\overline{X}_i, \overline{X}_j) = (\overline{X}_i \cdot \overline{X}_j + c)^h$
Sigmoid kernel	$K(\overline{X}_i, \overline{X}_j) = \tanh(\kappa \overline{X}_i \cdot \overline{X}_j - \delta)$



Valutazione della bontà della classificazione

$$ACC = \frac{1}{N} \sum_{k=1}^C \sum_{x \in C_k} I(C(x) \equiv C_k), \quad I = \begin{cases} 0, & C(x) \neq C_k \\ 1, & C(x) = C_k \end{cases}$$

$$bACC = \frac{1}{C} \sum_{k=1}^C \frac{1}{|C_k|} \sum_{x \in C_k} I(C(x) \equiv C_k)$$

Valutazione della bontà della classificazione

Predizione → Reference ↓	C_i	Altre classi
C_i	TP_i	FN_i
Altre classi	FP_i	TN_i

$$P_{\text{micro}} = \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C TP_i + FP_i}$$

$$R_{\text{micro}} = \frac{\sum_{i=1}^C TP_i}{\sum_{i=1}^C TP_i + FN_i}$$

$$F_{1\text{micro}} = 2 \frac{P_{\text{micro}} \cdot R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}}$$

$$P_i = \frac{TP_i}{TP_i + FP_i}$$

$$R_i = \frac{TP_i}{TP_i + FN_i}$$

$$F_1 = 2 \frac{P_i \cdot R_i}{P_i + R_i}$$

$$P_{\text{macro}} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FP_i} = \frac{\sum_{i=1}^C P_i}{C}$$

$$R_{\text{macro}} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i}{TP_i + FN_i} = \frac{\sum_{i=1}^C R_i}{C}$$

$$F_{1\text{macro}} = 2 \frac{P_{\text{macro}} \cdot R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}}$$

Valutazione della bontà della classificazione

$ROC(t) \propto FPR(t)$ vs. $TPR(t)$

$$TPR(t) \equiv Recall(t), FPR(t) = \frac{FP(t)}{FP(t) + TN(t)}$$

AUC, Area Under the Curve, tende a 1 per buona classificazione

