



Anno Accademico 2010 – 2011



Corso di Progettazione dei Sistemi di Controllo
Prof. Luca Schenato
Prof. Gianluigi Pillonetto

Gottardo Giuseppe, matr. 621855
Lanzini Andrea, matr. 621860
Zanin Claudia, matr. 626618

Applicazione di tecniche di machine learning per problemi di real-time tracking in reti di videosorveglianza

Introduzione (1/4)

Scopo del progetto

- ⦿ Applicazione delle tecniche di Machine Learning ad un problema di visual-tracking
- ⦿ Sviluppo di un algoritmo in real-time con SVM
- ⦿ Riduzione dell'onere computazionale

Introduzione (2/4)

Problematiche

- ⦿ Onere computazionale
- ⦿ Metodi di selezione di features e patterns
- ⦿ Adattabilità del classificatore

Introduzione (3/4)

Stato dell'arte

- 2 fasi:

- Fase iniziale → Training

- Fasi successive → Test (classificazione)

- Filtro di Kalman per tracking del soggetto in esame

Introduzione (4/4)

Contributo apportato dal progetto

Creazione dell'iperpiano ottimo mediante approssimazione
quadratica di una funzione di costo

Concetti di base (1/6)

Machine Learning

Unsupervised
Learning

Supervised
Learning

Clustering

Classificazione

Binaria

Multi-classe

Support Vector Machine (SVM)

Supervised Learning

Apprendimento basato

su Training Set → Insieme di coppie (examples)



Costruzione classificatore



Test su nuove istanze

Istanza

Label

Features

fissate
a priori

Concetti di base (2/6)

Support Vector Machine

Caso linearmente separabile

Caso NON linearmente separabile



- Minimizza il rischio strutturale e non quello empirico
- Riassume l'informazione del Training Set nei SVs



Minore onere computazionale

Concetti di base (3/6)

SVM – caso linearmente separabile

- Training set T linearmente separabile

- Iperpiano di separazione

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

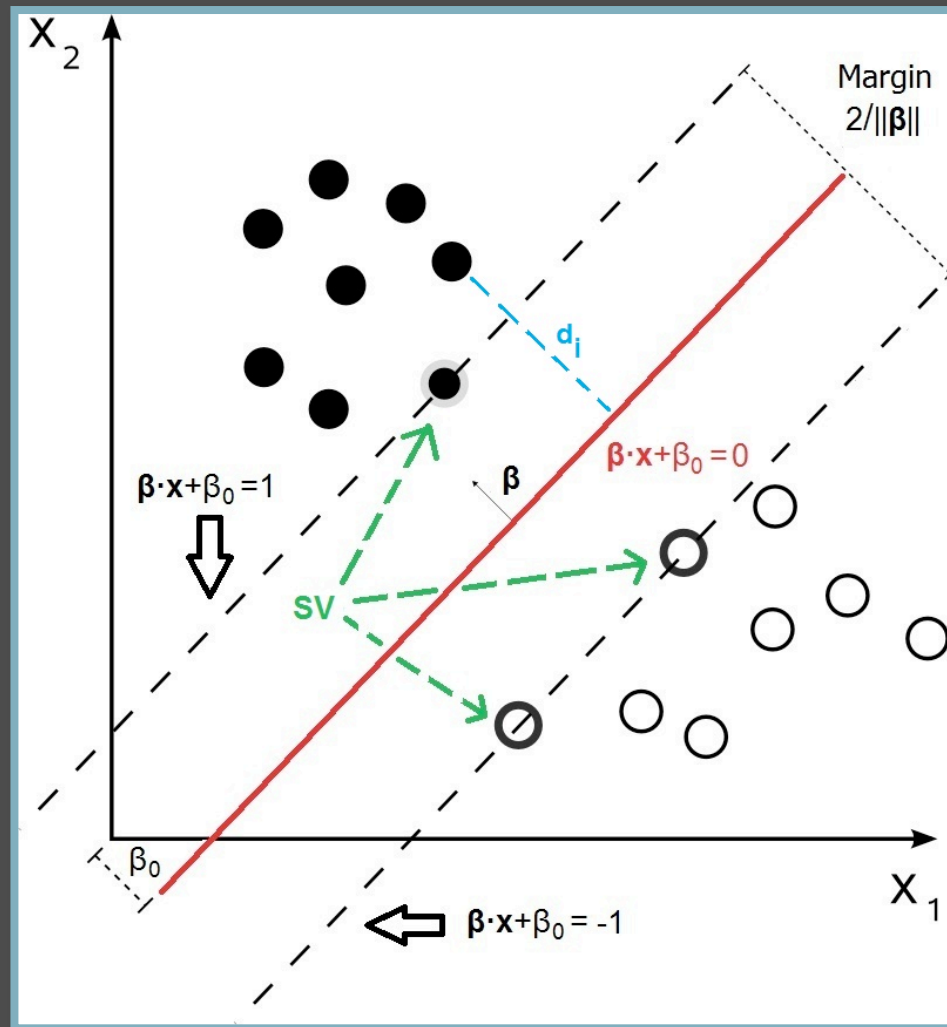
- Distanza dei punti dall'iperpiano

$$d_i = \frac{\boldsymbol{\beta} \cdot \mathbf{x}_i + \beta_0}{\beta}$$

- Obiettivo: individuare l'OSH lineare

- Massimizzazione margine
- Ricerca Support Vectors (SVs)

Caso dati linearmente separabili



Concetti di base (4/6)

SVM – caso non linearmente separabile

- Training set T non linearmente separabile

- Iperpiano di separazione

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

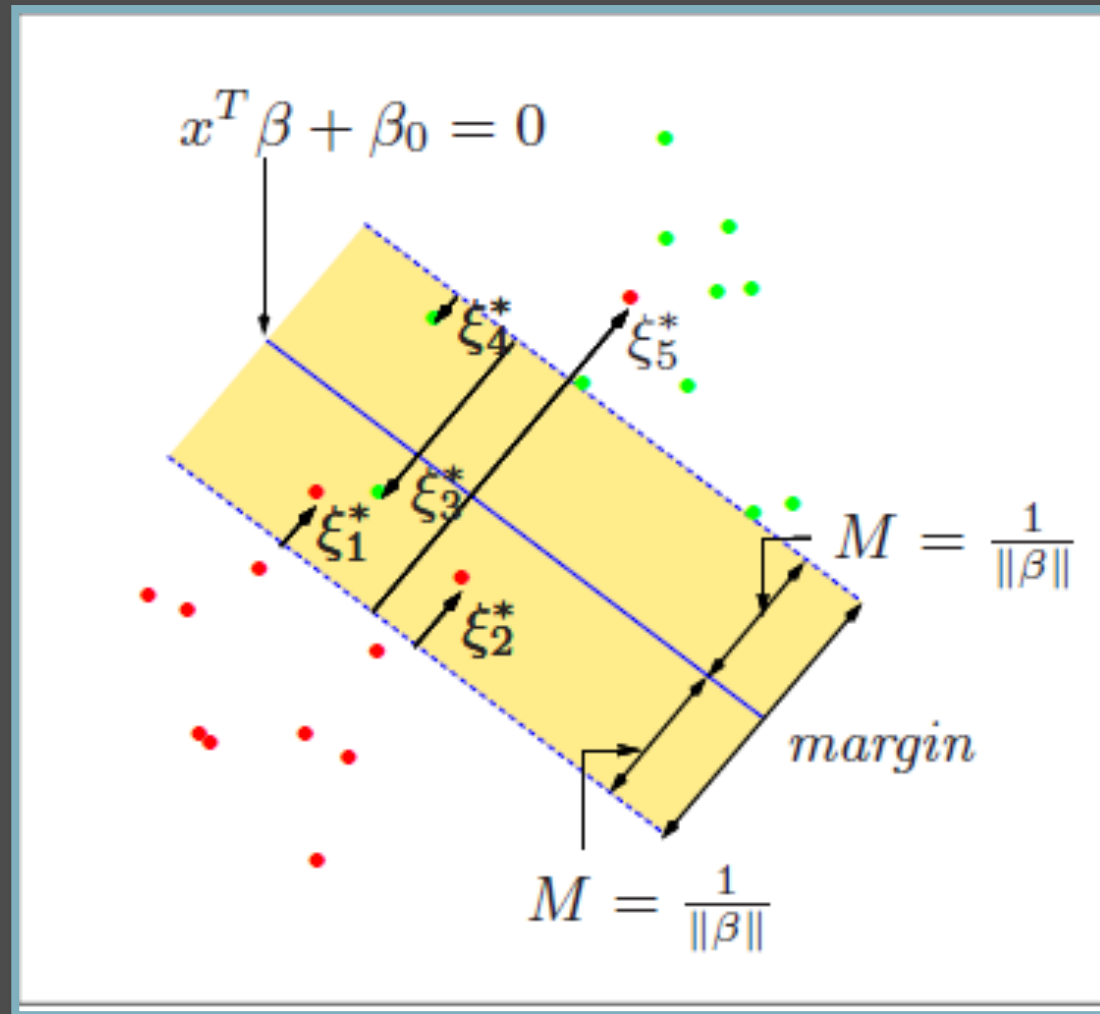
- Introduzione variabili di slack ξ per le miss-classifications

- Obiettivo: individuare l'OSH lineare

 - Massimizzazione margine

 - Ricerca Support Vectors (SVs)

Caso dati non linearmente separabili



Concetti di base (5/6)

Contributo apportato dal progetto

- Riformulazione del problema tramite **minimizzazione** di

Hinge Loss Function

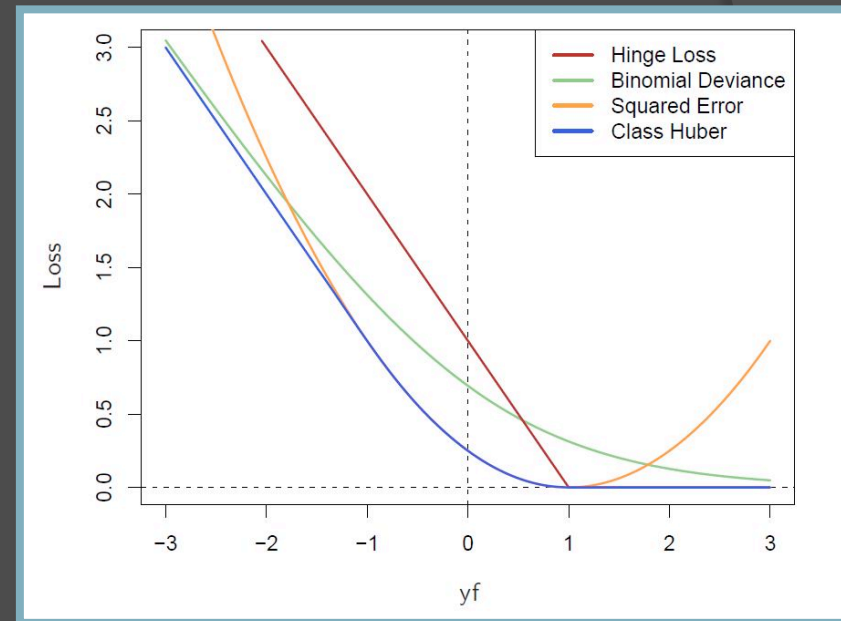
+

termine di regolarizzazione:

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

- Approssimazione differenziabile della funzione di costo:

$$\sum_{i=1}^{N_P} \log[1 + e^{-y_i f(x_i)}] + \lambda \|\beta\|^2$$



Concetti di base (6/6)

Contributo apportato dal progetto

Inserimento dinamica iperpiano \longrightarrow modello random-walk

$$\begin{bmatrix} \beta_{t+1} \\ \beta_{0t+1} \end{bmatrix} = \begin{bmatrix} \beta_t \\ \beta_{0t} \end{bmatrix} + \begin{bmatrix} w_t \\ w_{0t} \end{bmatrix}$$

$$Q = \begin{bmatrix} \text{Var}(w_t) & 0 \\ 0 & \text{var}(w_{0t}) \end{bmatrix}$$

↓

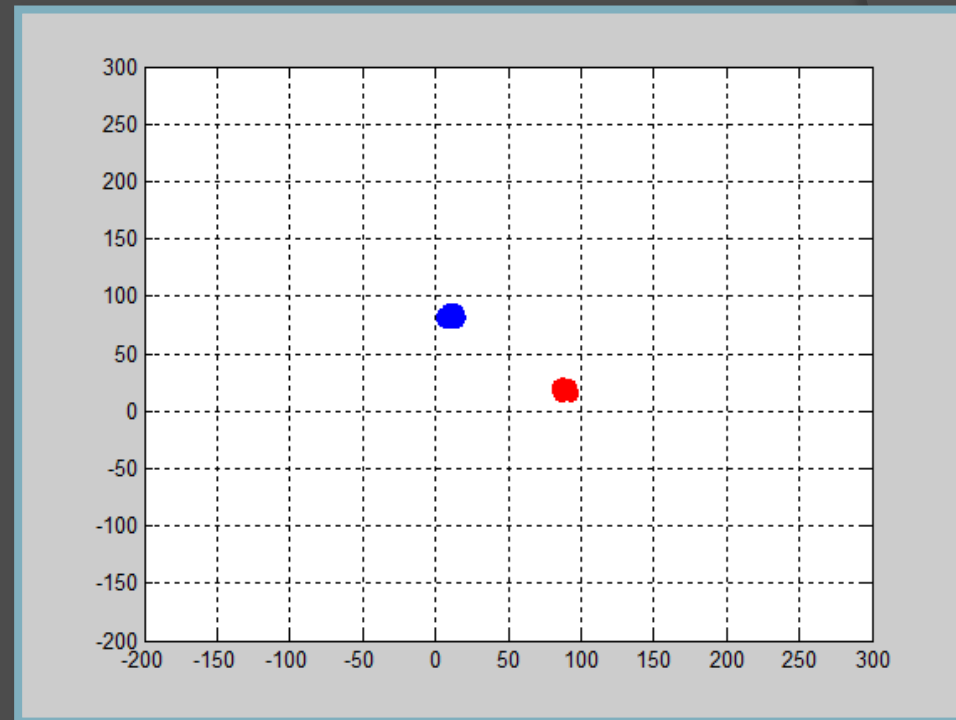
Algoritmo Sequential Quadratic Programming
(SQP) dinamico

- ↓
- ⊙ Approssimazione quadratica della funzione di costo
 - ⊙ Ricerca minimo ad ogni istante di tempo (linearizzazioni)

Implementazione (1/2)

Generazione del modello dinamico

- ⊙ Realizzazione in \mathcal{R}^2
- ⊙ 2 features: x, y
- ⊙ Iperpiano: retta
- ⊙ Moto uniforme
- ⊙ Velocità random



Implementazione (2/2)

Algoritmo SQP dinamico

⊙ Training iniziale con *libSVM* → iperpiano iniziale

⊙ Impostazioni iniziali (Q, N_lin)



⊙ Classificazione sul frame successivo

⊙ Ricerca nuovo iperpiano ottimo

⊙ Linearizzazioni iterate della funzione di costo

⊙ Ricerca del minimo



Nuovo frame

Simulazione SQP dinamico (1/3)

- ⊙ Impostazione parametri
 - ⊙ *Scelta della dinamica dei punti (fissa)*
 - ⊙ *Varianza Q del rumore di modello*
 - ⊙ *Numero di linearizzazioni*

Simulazione SQP dinamico (2/3)

- 30 frames
- 5, 10, 20 linearizzazioni
- Q

- $Q = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 100 \end{bmatrix}$

- $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- $Q = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- $Q = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$

- $Q = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$

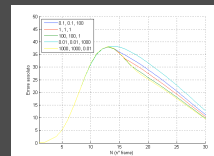
Simulazione SQP dinamico (3/3)

Studio prestazioni

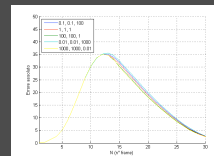
- Confronto distanze punto-retta

→ caso ideale/caso reale (errore assoluto)

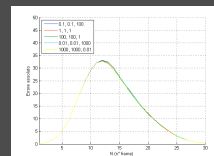
- 5 linearizzazioni



- 10 linearizzazioni

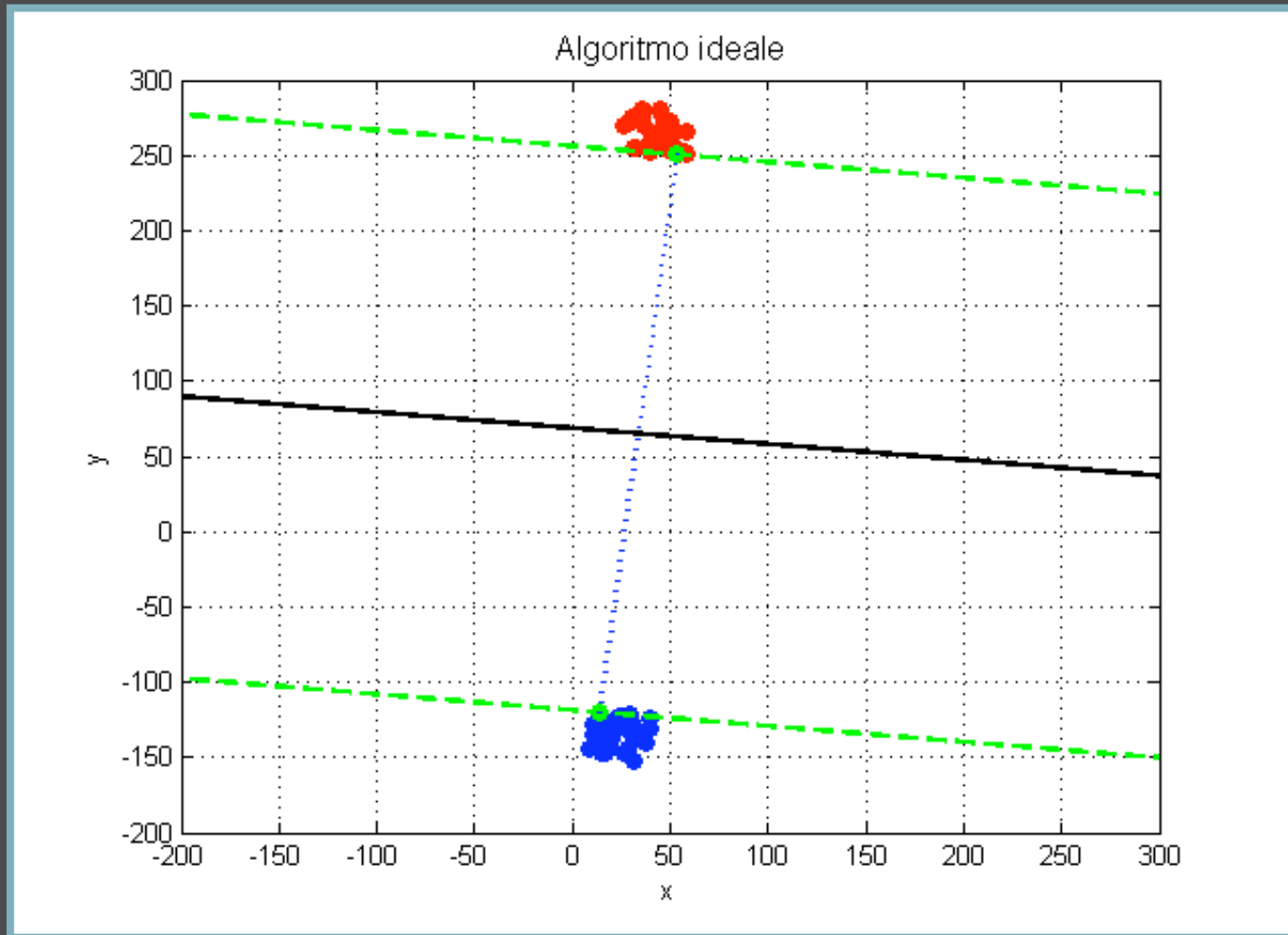


- 20 linearizzazioni

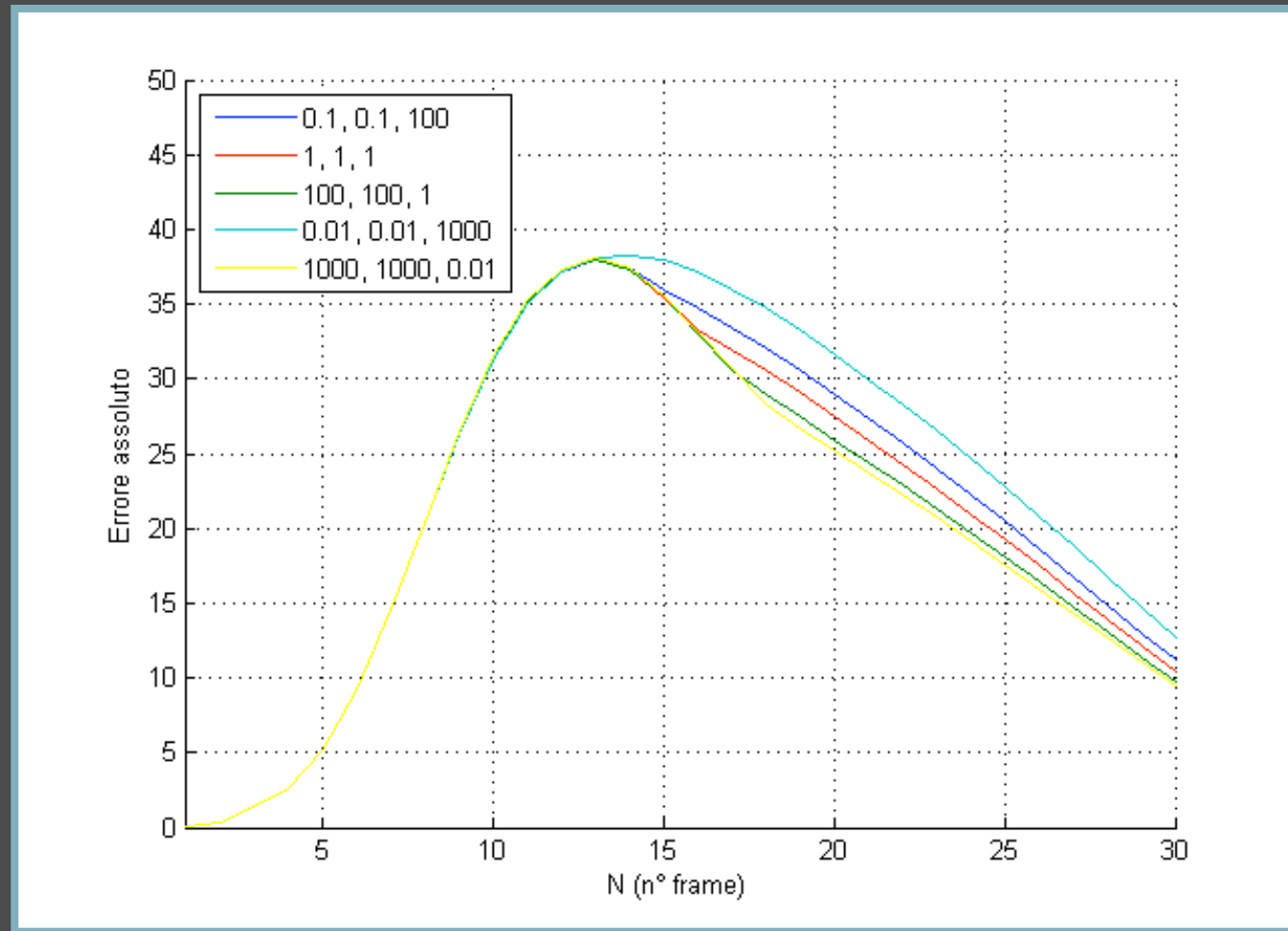


VIDEO

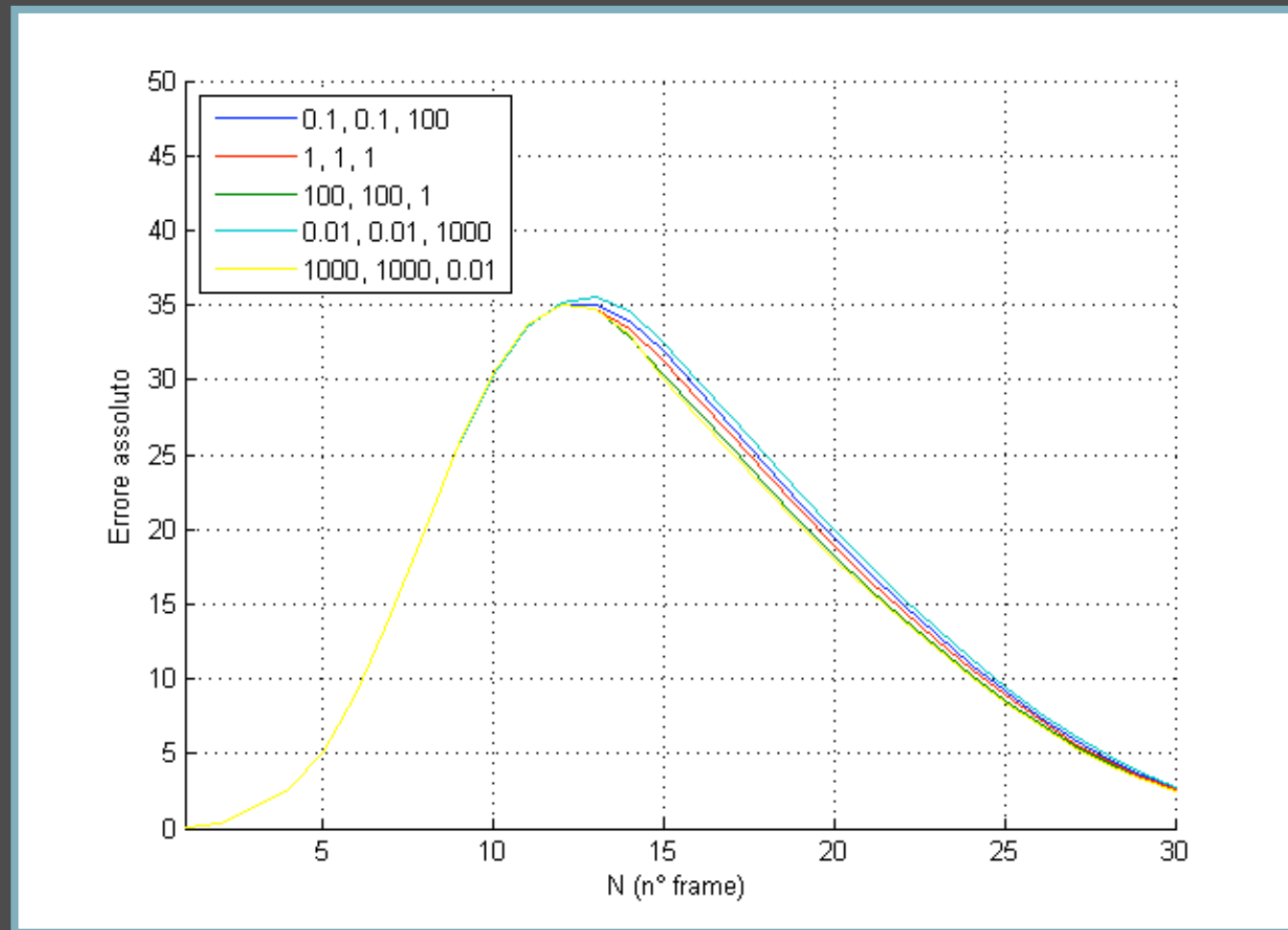
Distanza punto-retta



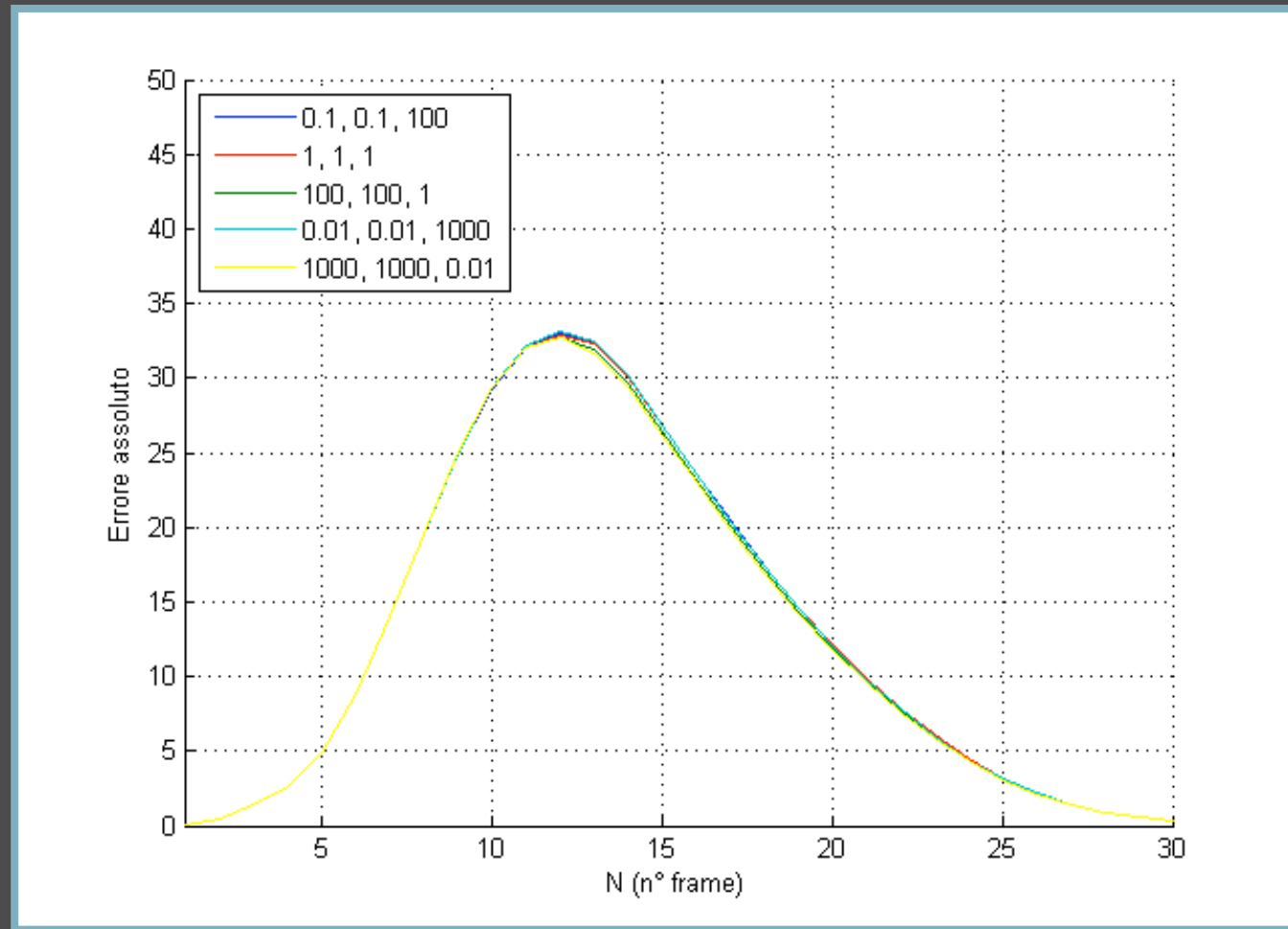
5 linearizzazioni



10 linearizzazioni

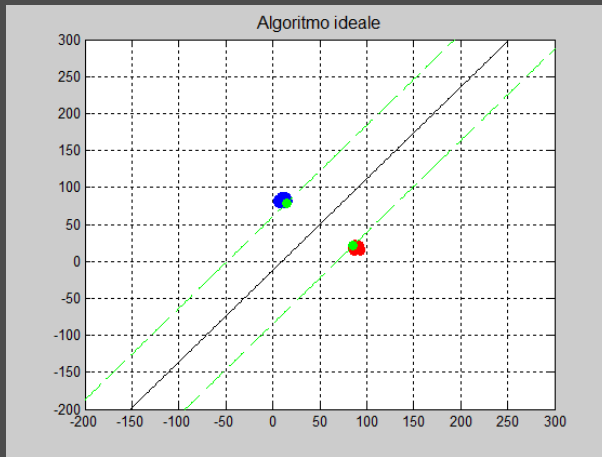


20 linearizzazioni

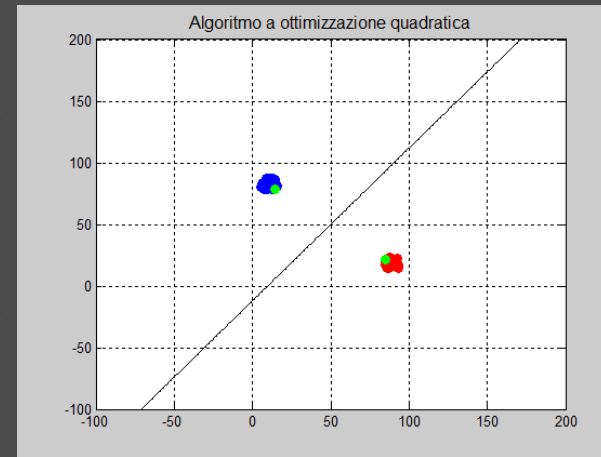


Filmati simulazioni SQP dinamico

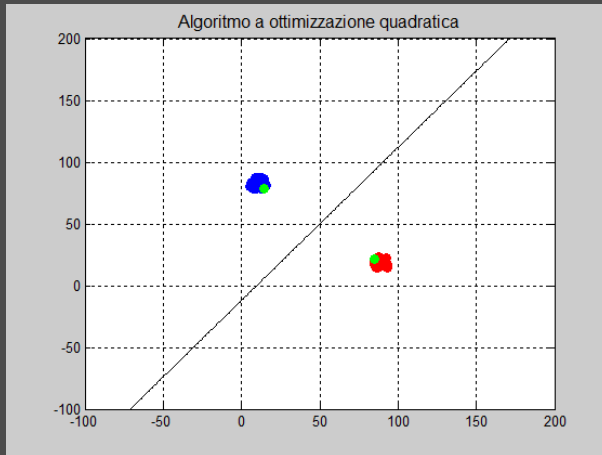
Caso ideale



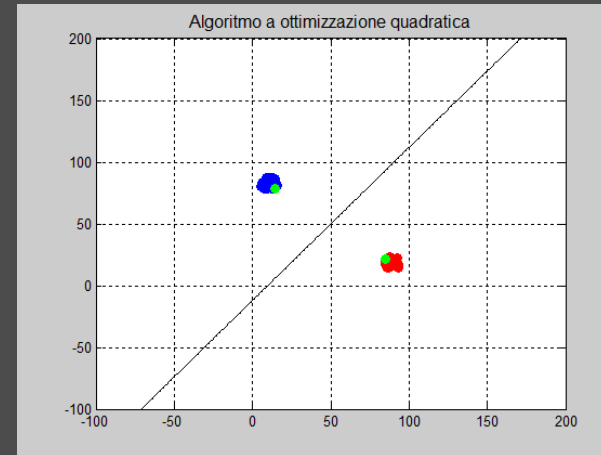
20 lin., $Q \rightarrow 0.01, 1000$



5 lin., $Q \rightarrow 1000, 0.01$



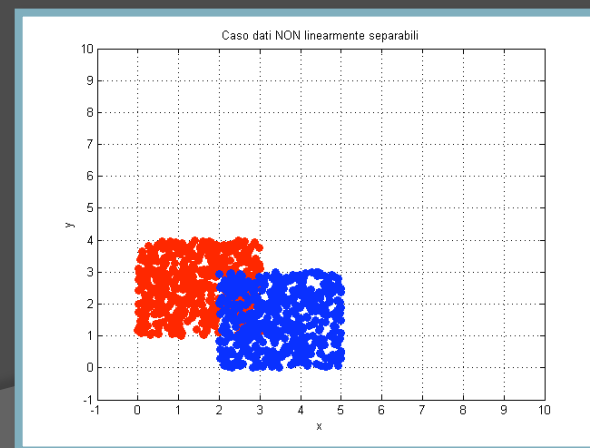
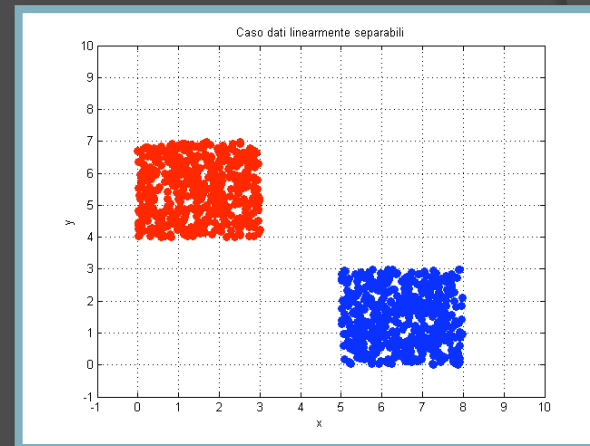
5 lin., $Q \rightarrow 0.01, 1000$



Implementazione (1/2)



Generazione del modello statico

- Realizzazione in \mathcal{R}^2
- Dataset di 1000 punti
- 2 casi:
 - Dati linearmente separabili
 - Dati non linearmente separabili



Implementazione (2/2)

Algoritmo SQP statico

- Dataset pre-creato
- Impostazioni iniziali (β_{iniz} , N_{lin})
- Training iniziale con *algoritmo SQP* ($Q = 0$) eseguito su un gruppo di punti a scelta \longrightarrow *primo iperpiano*

- Analisi nuovo sottogruppo del dataset 
- Ricerca nuovo iperpiano ottimo

Nuovo
sottogruppo

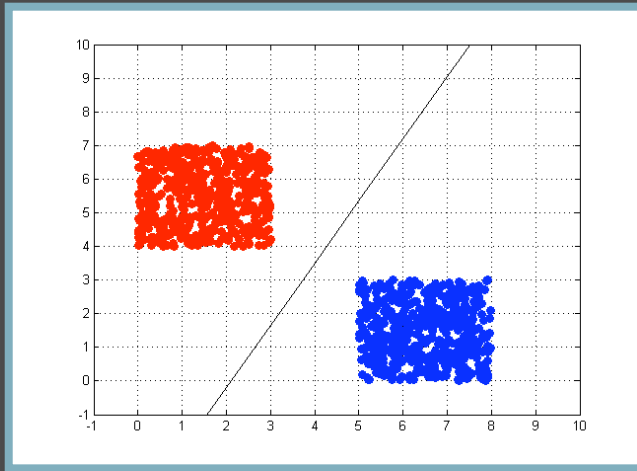
Simulazione SQP statico (1/4)

- ⊙ Studio di 3 differenti casi:
 - ⊙ *Training totale con svmtrain*
 - ⊙ *Training totale con SQP*
 - ⊙ *Training iniziale + successivi parziali con SQP*
- ⊙ Impostazione parametri (nel 3° caso)
 - ⊙ *Varianza $Q = 0$ del rumore di modello*
 - ⊙ *100 punti per il Training iniziale*
 - ⊙ *numero variabile di punti per i sottogruppi successivi al primo*
 - ⊙ *20 linearizzazioni*

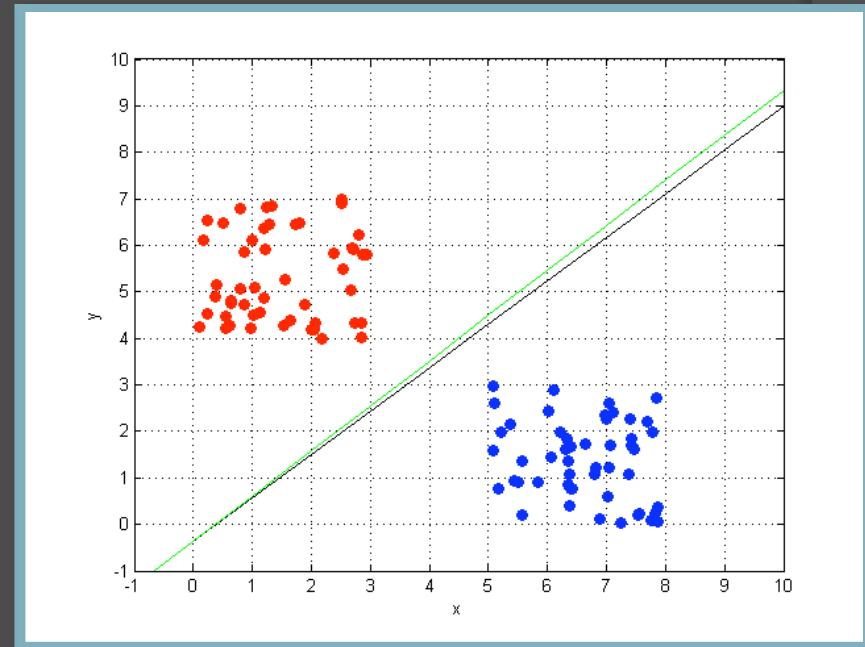
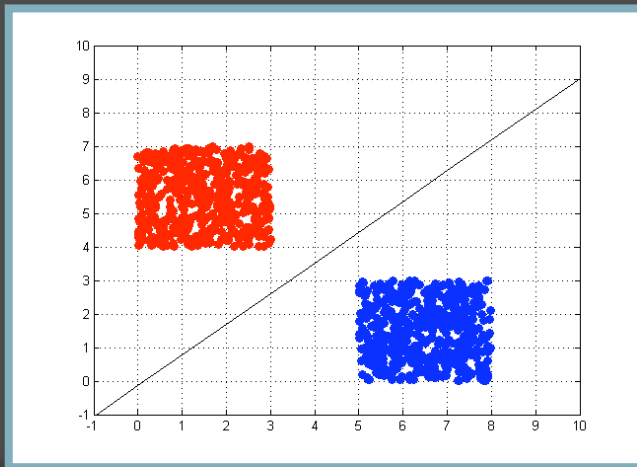
Simulazione SQP statico (2/4)

- Case data linearmente separabili (20 linearizzazioni)

svmtrain



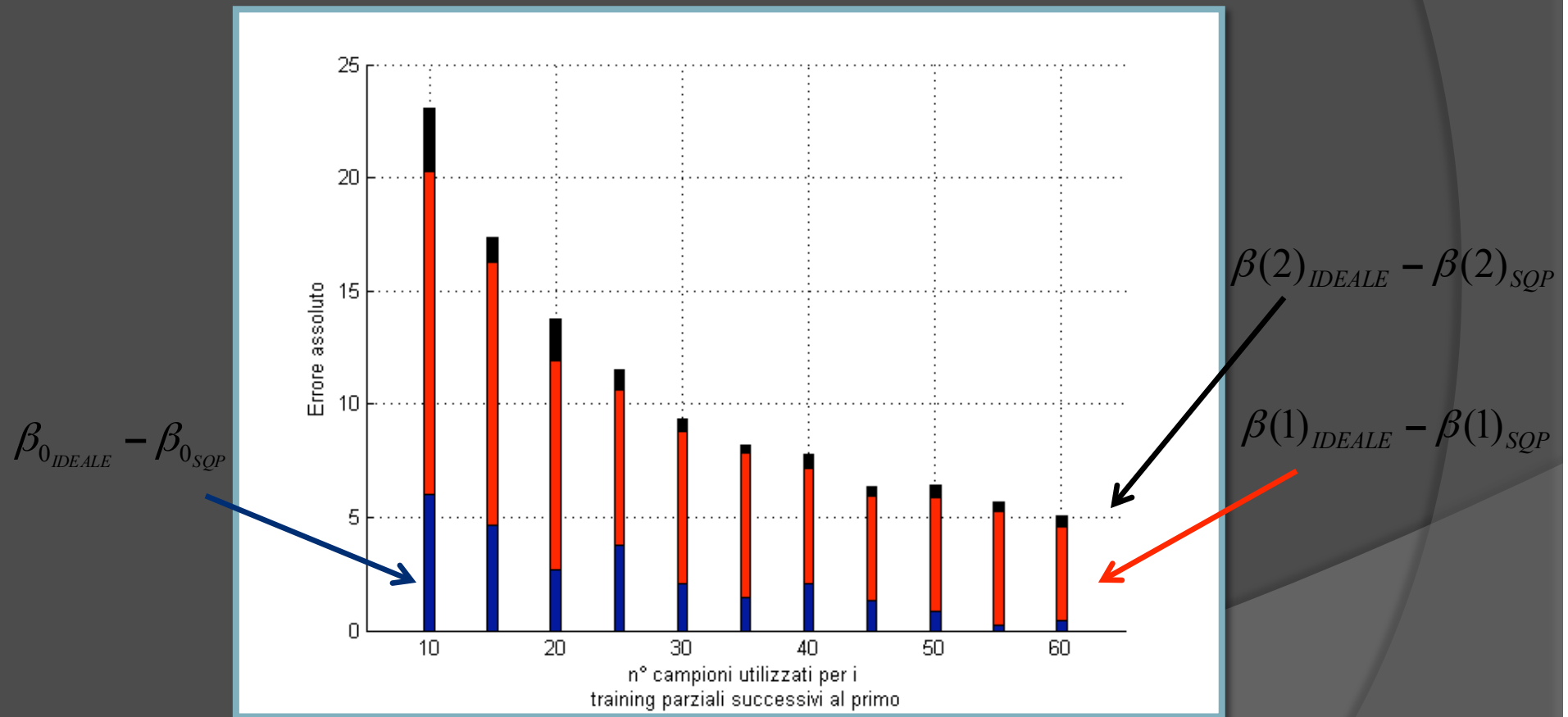
SQP totale – 20 lin.



SQP parziale – 20 lin., 100+50 punti

Simulazione SQP statico (3/4)

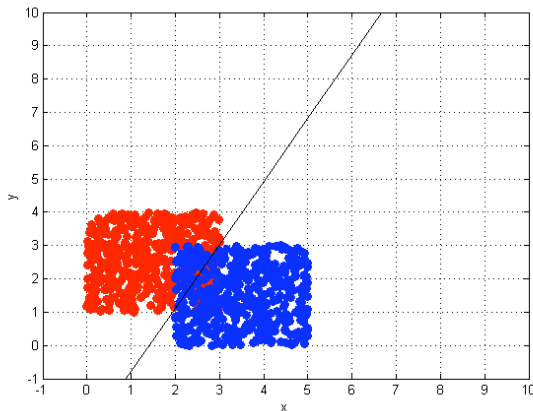
- Prestazioni - caso dati linearmente separabili



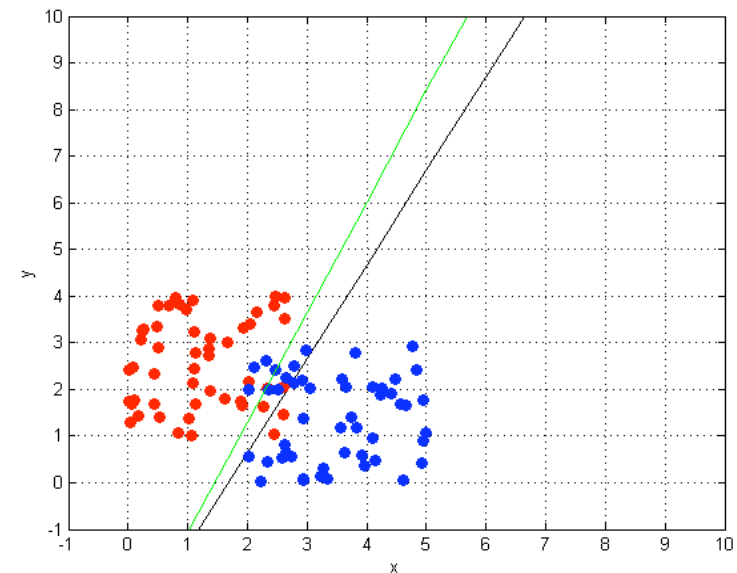
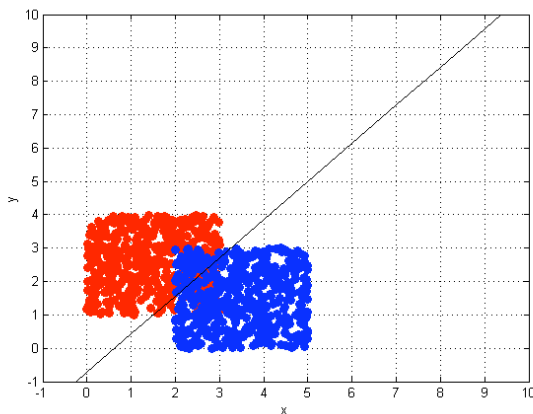
Simulazione SQP statico (4/4)

- Caso dati linearmente non separabili (20 linearizzazioni)

svmtrain



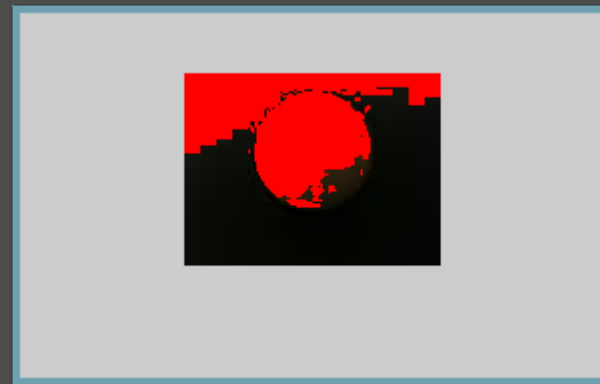
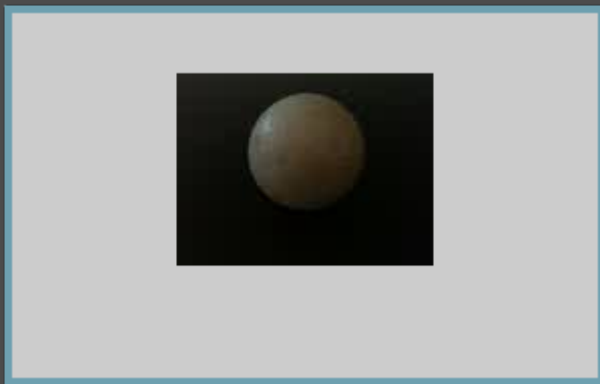
SQP totale – 20 lin.



SQP parziale – 20 lin., 100+50 punti

Simulazione 2 SQP dinamico

- ⊙ Applicazione algoritmo SQP su video
- ⊙ 3 features → R, G, B
- ⊙ Foreground → pallina bianca con variazioni esposizione
- ⊙ Background → ambiente molto scuro



Conclusioni

⊙ Caso SQP dinamico

- ⊙ Capacità di adattamento alla variazione di features
- ⊙ Generalità del modello utilizzato
- ⊙ Assenza di controllo

⊙ Caso SQP statico

- ⊙ Algoritmo distribuito → minore onere computazionale
- ⊙ Efficienza buona rispetto allo stato dell'arte

Possibili sviluppi futuri

- ⦿ Controllo adeguato correzione dinamica iperpiano
- ⦿ Miglioramento sensibilità della funzione di costo
- ⦿ Implementazione algoritmo con filtro di Kalman iterato
- ⦿ Utilizzo nell'algoritmo SQP dinamico del Training creato con SQP statico