

# Secondo report

Giacomo Longo (4336477) e Roberta Tassara (4336488)

28 Novembre 2019

# 1 Confronto tra gli approcci lineari e a kernel per la regressione

## 1.1 Approccio lineare

Per l'approccio lineare, seguiamo lo stesso schema del report precedente: cerchiamo di formulare una funzione del tipo  $f(\underline{x}) = \underline{w}^T \underline{x}$  per approssimare l'andamento di una funzione reale a partire da dei suoi valori campionati.

Tale funzione assumerà una forma  $w_n x_n + \dots + w_0 x_0$  o altrimenti una forma  $w_{n+1} x_n + \dots + w_1 x_0 + w_0$  qualora decidessimo di aggiungere a  $X$  un vettore di uni a rappresentare il termine di *bias*, per chiarezza, la funzione così definita ha la forma  $f(\underline{x}) = \underline{w}^T \underline{x} + b$ .

$$\underline{w} = \underset{\underline{w}}{\operatorname{argmin}} \|X\underline{w} - \underline{y}\|^2 + \lambda \|\underline{w}\|^2$$

Con  $\lambda$  definito come *iperparametro di regolarizzazione*.

Il codominio di  $f(\underline{x})$  è  $\mathbf{R}$ .

I due approcci sottostanti sono frutto del teorema della rappresentazione.

### 1.1.1 Approccio lineare primale

$$\underline{w} = (X^T X + \lambda I)^{-1} X^T \underline{y}$$

La complessità di questo approccio è  $O(d^2)$  ove  $d$  è il numero di dimensioni di  $\underline{x}$ .

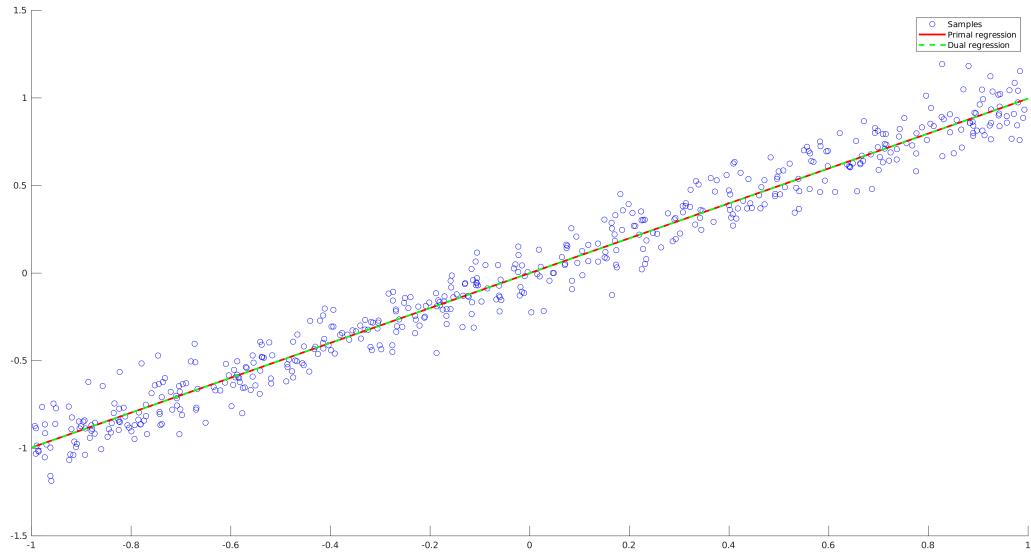
Di conseguenza è raccomandato per quando i dati sono tanti ma hanno un basso numero di feature.

### 1.1.2 Approccio lineare duale

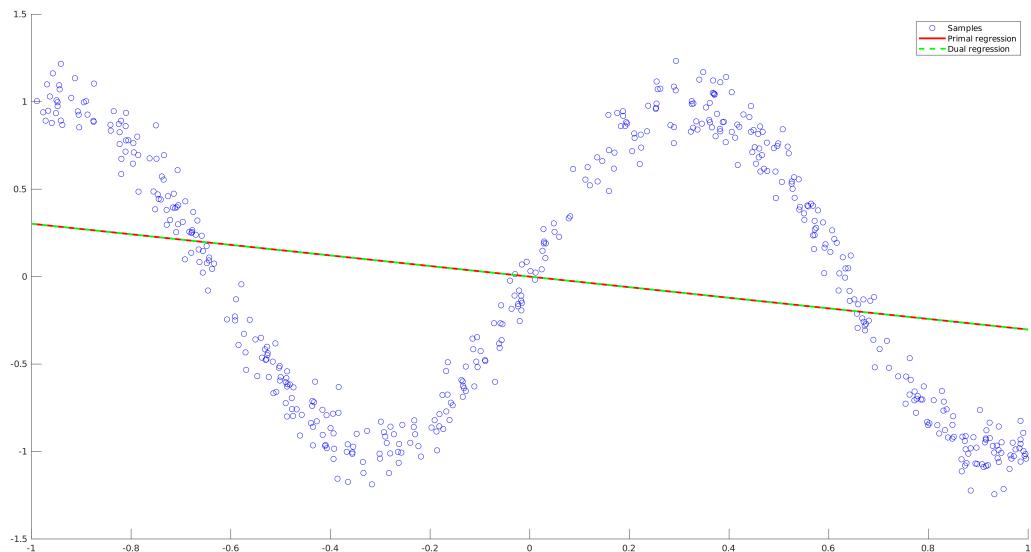
$$\begin{aligned}\underline{w} &= X^T \underline{\alpha} \\ \underline{\alpha} &= (Q + \lambda I)^{-1} \underline{y} \\ Q_{ij} &= \underline{x}_i^T \underline{x}_j\end{aligned}$$

La complessità di questo approccio è  $O(n^2)$  ove  $n$  è il numero di sample. Di conseguenza è raccomandato per quando i dati hanno molta cardinalità ma poche colonne.

### 1.1.3 Analisi dell'approccio



In questo esempio il regressore lineare puó generare una buona approssimazione della funzione originale.



Introducendo invece una non linearitá, il regressore non riesce a generare una funzione che approssimi correttamente.

## 1.2 Approccio a kernel

L'approccio a kernel consiste nello sfruttamento di particolari funzioni, dette *kernel* capaci di generare uno spazio vettoriale di dimensione sufficiente a rappresentare anche funzioni non lineari.

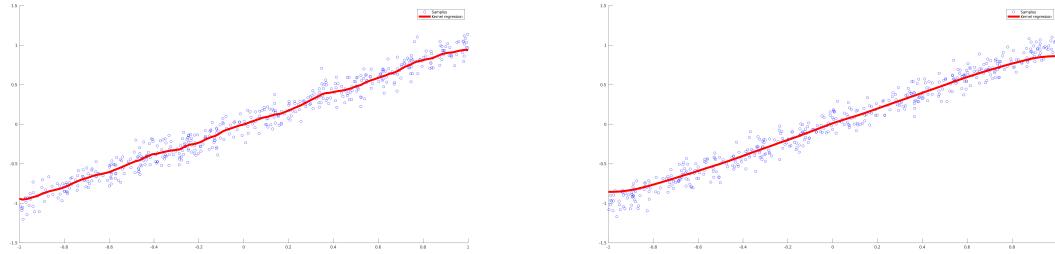
La funzione frutto del processo ha la forma

$$f(\underline{x}) = \underline{w}^T \underline{\phi}(\underline{x}) = \sum_{i=1}^n \alpha_i K_{x_i}(x)$$

$$K_{x_i}(x) = K(x_i, x) = \exp -\gamma \|x_i - x\|^2$$

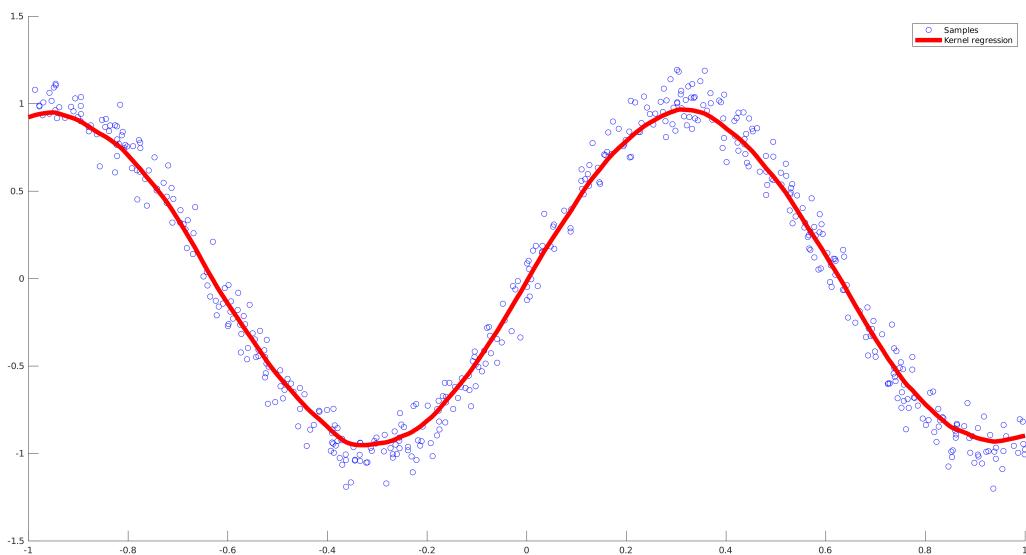
### 1.2.1 Analisi dell'approccio

Considerando una funzione lineare, analizziamo il risultato della variazione del parametro  $\gamma$



A sinistra viene utilizzato  $\gamma = 1$ , a destra viene utilizzato  $\gamma = 0.1$ . Possiamo notare che per  $\gamma$  maggiore, la funzione ha al suo interno più componenti non lineari (ad alta frequenza) che si mostrano come vibrazioni rispetto alla linea.

Diminuendo il parametro si ottiene una funzione più "pulita", seppur si evidenzino le non linearità presenti agli estremi dell'intervallo considerato.



Analizzando nuovamente la funzione seno, l'approccio a kernel ci permette di ottenere un'approssimazione valida anche nel caso non lineare.

## 2 Classificatore a due classi

Desideriamo dividere i nostri sample in due classi  $\{+1, -1\}$ .

A differenza del caso precedente, non ci interessa piú sapere la distanza della nostra funzione dai sample reali ma solo se un oggetto é stato classificato correttamente  $fy > 0$  o erroneamente  $fy < 0$ .

La nostra intuizione ci suggerirebbe di utilizzare una loss function a gradino: a sinistra dello zero, 1, a destra dello zero, 0.

In questo modo, potremmo generare il nostro classificatore tramite la minimizzazione.

Tale funzione però, non ci consente di trovare il suo minimo in maniera efficiente.

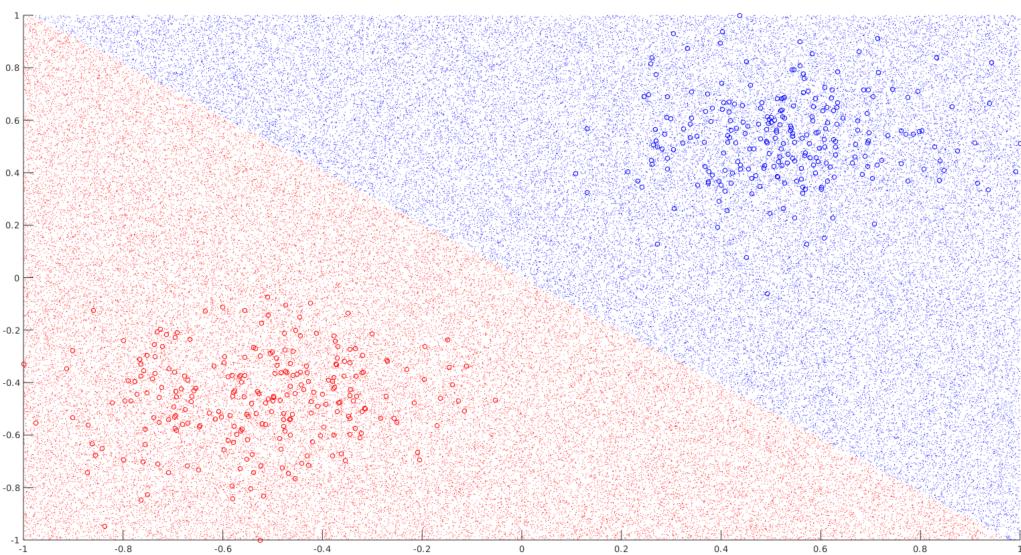
Provando quindi a applicare la square loss, scopriamo che questa funzione si presta al compito:

1. Il minimo della square loss si ha nella zona compresa tra 0 e 2.
2. Alla zona a sinistra dello zero (ovvero la parte in cui il classificatore sbaglia), sono associati valori  $>$  rispetto alla zona relativa al punto 1
3. La zona per  $x > 2$  é maggiore della zona del punto 1

La minimizzazione della square loss quindi ha due effetti:

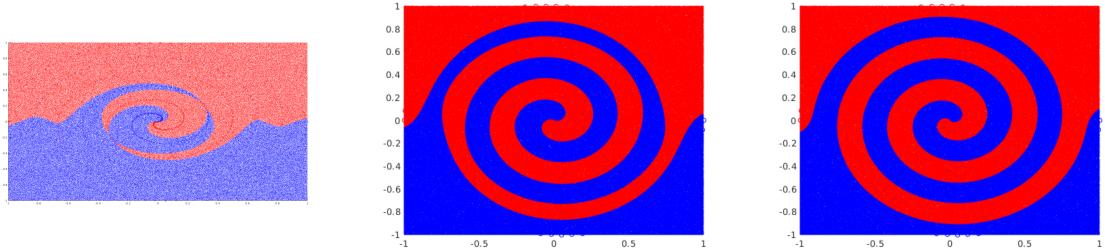
- Minimizzazione degli errori di classificazione (per il punto 2)  $\rightarrow$  nostro obiettivo
- Minimizzazione della distanza della loss function rispetto al punto  $x = 1$  (per il punto 3)  
 $\rightarrow$  effetto secondario

Di conseguenza la square loss rappresenta una buona approssimazione anche per risolvere questo problema.



Utilizzando una funzione lineare, possiamo tracciare una retta che divide due classi di punti bilanciati rispetto a tale asse.

Come nel caso della regressione lineare, nel caso i punti fossero non bilanciati rispetto all'origine, si sarebbe dovuto introdurre un termine di bias.



Utilizzando un classificatore a kernel, possiamo classificare anche punti che non sono bilanciati rispetto a una ideale retta che li divida. Qui si hanno tre classificatori al variare di  $\gamma$ , da sinistra verso destra:  $\gamma = 0.1, \gamma = 1, \gamma = 10$ . Come si può notare, il classificatore per gamma molto basso non riesce a classificare correttamente tutti i sample, avendo bisogno di una funzione più complessa. Aumentando  $\gamma$  si ottiene invece uno classificatore più preciso, da notare che il classificatore "intermedio", nonostante riesca a classificare correttamente tutti i punti, presenta delle imperfezioni nella sua forma che possono causare una misclassificazione di eventuali punti presenti nelle zone deformate.

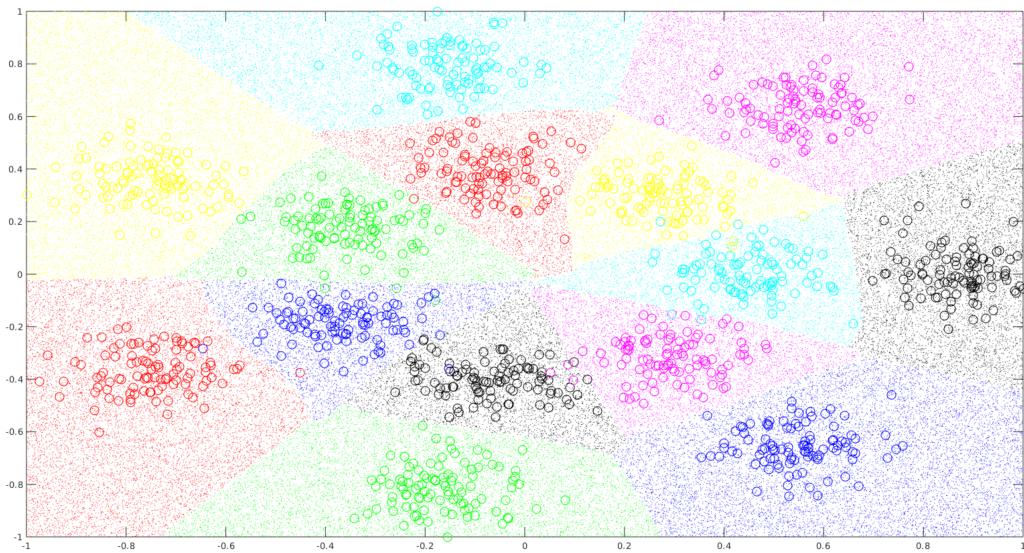
### 3 Classificatore multiclass A.V.A.

Per classificare in più classi, utilizziamo la tecnica ALL VS ALL, esiste anche un'altra tecnica, la ONE VS ALL che non tratteremo in questa relazione.

La tecnica AVA consiste nel trasformare il problema in uno o più problemi di binary classification in modo da riutilizzare la conoscenza acquisita precedentemente.

$$n_p = \binom{c}{2} \quad n_p = \text{numero di problemi} \quad c = \text{numero di classi}$$

La classificazione di un punto consiste nell'utilizzare tutti i classificatori in sequenza e successivamente decidere l'appartenenza del punto sulla base di una votazione a maggioranza di tutti i sottoclassificatori.



Qui, utilizzando classificatori a kernel, si visualizzano le zone relative a 7 classi rappresentate dai 7 diversi colori.

Si evidenzia la forma non lineare delle zone risultanti.

Alcuni sample particolarmente distanti dal cluster più vicino, sono stati classificati erroneamente. Ad esempio, nel cluster di punti rossi in basso a sinistra è presente un sample blu.