



A.D. 1308  
**unipg**

DIPARTIMENTO  
DI INGEGNERIA

Progetto di  
**Signal Processing and Optimization for Big Data**

Corso di Laurea in Ingegneria Informatica e Robotica

Curriculum Data Science – A.A. 2024-2025

DIPARTIMENTO DI INGEGNERIA

docente

Prof. Paolo BANELLI

# **Confronto tra Algoritmi di Regressione Lasso Sviluppati da Zero**

363433 **Gian Marco Ferri** gianmarco.ferri@studenti.unipg.it

# 1 Introduzione

Questo progetto affronta la risoluzione del problema di regressione Lasso tramite l'implementazione e il confronto di diversi algoritmi. In particolare, sono stati sviluppati in Matlab tre approcci distinti: ISTA, ADMM e una versione simulata di ADMM distribuito su più agenti.

Nella parte finale della documentazione vengono presentati dei confronti tra i tre algoritmi in termini di tempi di calcolo, numero di iterazioni e grafici che illustrano l'andamento della convergenza durante le iterazioni.

# 2 Dataset

Il dataset utilizzato è il California Housing, un insieme di dati pubblici che raccoglie informazioni relative alle abitazioni in California. Ogni riga rappresenta un blocco residenziale e le colonne includono le seguenti variabili:

- **longitude**: longitudine del blocco;
- **latitude**: latitudine del blocco;
- **housing\_median\_age**: età mediana delle abitazioni nel blocco;
- **total\_rooms**: numero totale di stanze nel blocco;
- **total\_bedrooms**: numero totale di camere da letto nel blocco;
- **population**: popolazione nel blocco;
- **households**: numero di nuclei familiari nel blocco;
- **median\_income**: reddito mediano delle famiglie nel blocco;
- **median\_house\_value**: valore mediano delle abitazioni nel blocco;

Durante la fase di preprocessing, i valori mancanti nella variabile *total\_bedrooms* sono stati sostituiti con la mediana. Tutte le feature numeriche sono state normalizzate nell'intervallo  $[0, 1]$ . La variabile categorica *ocean\_proximity* è stata esclusa dall'analisi.

### 3 Architettura ed algoritmi

Gli algoritmi descritti di seguito sono stati implementati all'interno della classe LassoReg. In base ai parametri passati all'oggetto istanziato dalla classe, è possibile scegliere l'algoritmo con cui verrà eseguita la fase di training, lo step-size, la tolleranza per la convergenza, il numero massimo di iterazioni e la penalità da applicare.

#### 3.1 Soft-Thresholding

Il problema Lasso originale è il seguente:

$$\min_{\mathbf{x}} \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{x}\|_1$$

In questo algoritmo il problema viene risolto attraverso iterazioni di gradient descent fino a convergenza. Poiché la norma  $L_1$  non è differenziabile, viene fatto ricorso al concetto di subdifferenziale attraverso l'uso dell'operatore di *soft-thresholding*.

$$\nabla \mathbf{x} = -\frac{2}{m} A(\mathbf{y} - A\mathbf{x}) + S(\mathbf{x}, \alpha)$$

dove

$$S(x, \alpha) = \begin{cases} x - \alpha & \text{se } x > \alpha \\ x + \alpha & \text{se } x < -\alpha \\ 0 & \text{altrimenti} \end{cases}$$

L'aggiornamento è dato da:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - t \nabla \mathbf{x}$$

Il criterio di convergenza è il seguente:

$$\text{se } |\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}| < \epsilon \rightarrow \text{break}$$

#### 3.2 ADMM

Iniziamo riformulando il problema originale introducendo la *slack variable*:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} \quad & \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{z}\|_1 \\ \text{s.t.} \quad & \mathbf{x} - \mathbf{z} = 0 \end{aligned}$$

Risolviamo il problema mediante ADMM scalato:

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}\|_2^2 \right\}$$

$$\mathbf{z}^{(k+1)} = \underset{\mathbf{z}}{\operatorname{argmin}} \left\{ \alpha \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\mathbf{x}^{(k+1)} - \mathbf{z} + \mathbf{u}^{(k)}\|_2^2 \right\}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(0)} + \sum_{i=1}^{k+1} \|\mathbf{x}^{(i)} - \mathbf{z}^{(i)}\|_2^2$$

Il primo step si può risolvere in forma chiusa eguagliando la derivata a 0.

$$\nabla_{\mathbf{x}} \left( \frac{1}{2} \|A\mathbf{x} - \mathbf{y}\|_2^2 + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}\|_2^2 \right) = 0$$

$$A^T (A\mathbf{x} - \mathbf{y}) + \rho(\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}) = 0$$

$$(A^T A + \rho I)\mathbf{x} = A^T \mathbf{y} + \rho(\mathbf{z}^{(k)} - \mathbf{u}^{(k)})$$

$$\rightarrow \mathbf{x}^{(k+1)} = (A^T A + \rho I)^{-1} (A^T \mathbf{y} + \rho(\mathbf{z}^{(k)} - \mathbf{u}^{(k)}))$$

Per il secondo step, invece, non esiste il gradiente, quindi, come nell'algoritmo precedente, si utilizza il sub-gradiente che in questo caso corrisponde al soft thresholding operator.

Concludendo, Lasso con ADMM scalato assume questa formulazione:

$$\mathbf{x}^{(k+1)} = (A^T A + \rho I)^{-1} (A^T \mathbf{y} + \rho(\mathbf{z}^{(k)} - \mathbf{u}^{(k)}))$$

$$\mathbf{z}^{(k+1)} = S_{\frac{\rho}{\alpha}}(\mathbf{x}^{(k+1)} + \mathbf{u}^{(k)})$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}$$

Il criterio di convergenza si basa sul calcolo dei residui primali e duali. Considerando  $n = \#features$  abbiamo che:

$$r = \|\mathbf{x} - \mathbf{z}\| \quad \text{residuo primale}$$

$$s = \|\rho(\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)})\| \quad \text{residuo duale}$$

$$\epsilon_r = \sqrt{n}\epsilon + \epsilon_{\text{rel}} \max(\|\mathbf{x}\|, \|\mathbf{z}\|) \quad \text{tolleranza primale}$$

$$\epsilon_s = \sqrt{n}\epsilon + \epsilon_{\text{rel}} \|\rho \mathbf{u}\| \quad \text{tolleranza duale}$$

$$\text{if } (r < \epsilon_r) \& (s < \epsilon_s) \rightarrow \text{break} \quad \text{criterio di convergenza}$$

### 3.3 ADMM distribuito

Il problema Lasso originale appartiene alla classe di problemi distribuibili in quanto il vettore delle osservazioni  $\mathbf{y}$  è separabile in  $N$  osservazioni e, di conseguenza, anche il prodotto scalare  $A\mathbf{x}$ .

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_N \end{bmatrix}, \quad A\mathbf{x} = \begin{bmatrix} A_1 \\ \vdots \\ A_N \end{bmatrix} \mathbf{x} = \begin{bmatrix} A_1\mathbf{x} \\ \vdots \\ A_N\mathbf{x} \end{bmatrix}$$

In relazione a ADMM, per la formulazione distribuita si va ad eseguire uno split del dataset in base al numero di agenti a disposizione, in modo tale che ognuno possa calcolare la propria variabile di ottimizzazione  $\mathbf{x}$  rispetto alla propria porzione di dati.

$$\begin{aligned} \min_{(\mathbf{x}_1, \dots, \mathbf{x}_N), \mathbf{z}} \quad & \sum_{i=1}^N \|\mathbf{y}_i - A_i\mathbf{x}_i\|_2^2 + \lambda \|\mathbf{z}\|_1 \\ \text{s.t.} \quad & \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \quad i = 1, \dots, N \end{aligned}$$

Risolviamo il problema mediante la versione scalata dell'ADMM:

$$\begin{aligned} \mathbf{x}_i^{(k+1)} &= \underset{\mathbf{x}_i}{\operatorname{argmin}} \left\{ \|\mathbf{y}_i - A_i\mathbf{x}_i\|_2^2 + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}^{(k)} + \mathbf{u}_i^{(k)}\|_2^2 \right\} \\ \mathbf{z}^{(k+1)} &= \underset{\mathbf{z}}{\operatorname{argmin}} \left\{ g(\mathbf{z}) + \frac{\rho}{2} \sum_{i=1}^N \|\mathbf{x}_i^{(k+1)} - \mathbf{z} + \mathbf{u}_i^{(k)}\|_2^2 \right\} \quad i = 1, \dots, N \\ &= S_{\frac{\lambda}{N\rho}}(\hat{\mathbf{x}}^{(k+1)} + \hat{\mathbf{u}}^{(k)}) \\ \mathbf{u}_i^{(k+1)} &= \mathbf{u}_i^{(k)} + (\mathbf{x}_i^{(k+1)} - \mathbf{z}^{(k+1)}) \quad i = 1, \dots, N \end{aligned}$$

Nel primo step si hanno  $N$  problemi distribuibili su vari agenti con una soluzione in forma chiusa, come nel caso centralizzato.

Nel secondo step servono tutte le variabili primali e duali per calcolare il valore globale  $\mathbf{z}$ , quindi verrà eseguito in un fusion center.

L'aggiornamento del terzo step viene invece eseguito localmente in ogni agente.

## 4 Comparazioni

I parametri con cui sono stati eseguiti i tre algoritmi sono i seguenti:

- iterazioni massime = 50000;
- step-size = 0.01;
- l1-penalty = 1;
- tolerance =  $1e-4$ ;
- agenti = 8 (ADMM distribuito);

Il confronto delle prestazioni ottenute è mostrato nella Tabella 1.

Tabella 1: Comparazione algoritmi

	R2	time (s)	iterazioni
ISTA	0.5588	12.876	50000
ADMM	0.5839	0.0010	4
ADMM-Dist	0.5726	0.0376	170

È importante sottolineare che i valori riportati sono stati ottenuti effettuando una suddivisione casuale del dataset in training set e test set: esecuzioni diverse potrebbero generare risultati leggermente differenti.

A seguire vengono illustrate le variazioni dei criteri di convergenza durante le iterazioni degli algoritmi, insieme ai grafici delle predizioni effettuate rispetto ai valori reali.

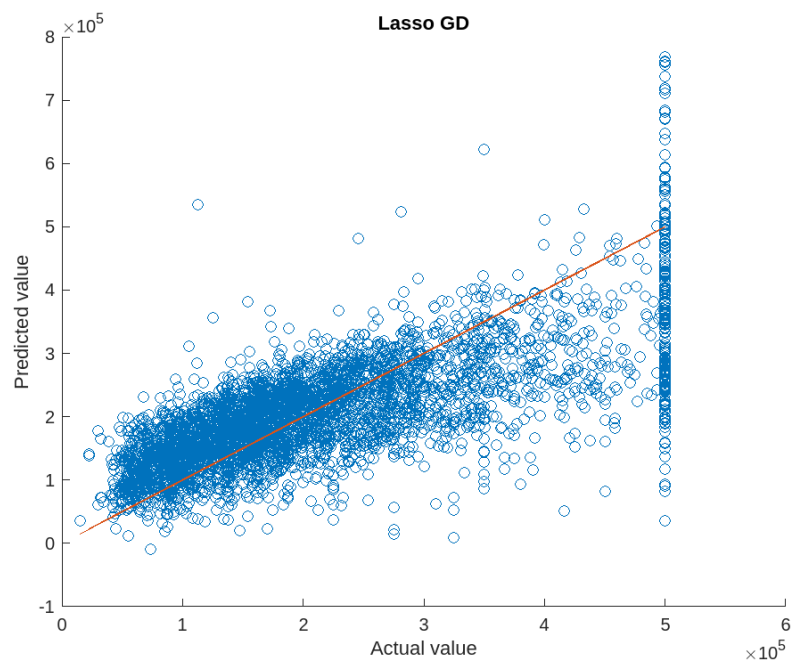


Figura 1: Predizioni ottenute con Soft-Thresholding.

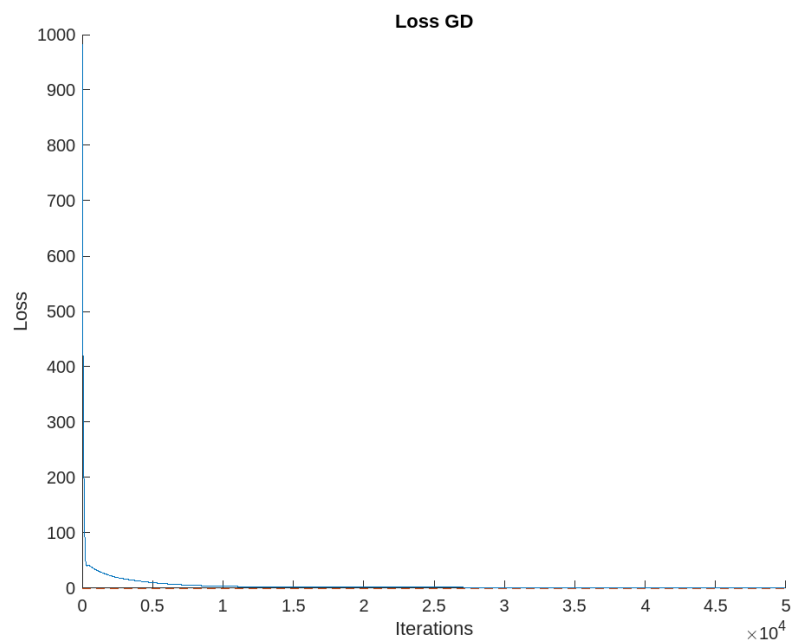


Figura 2: Convergenza del Soft-Thresholding.

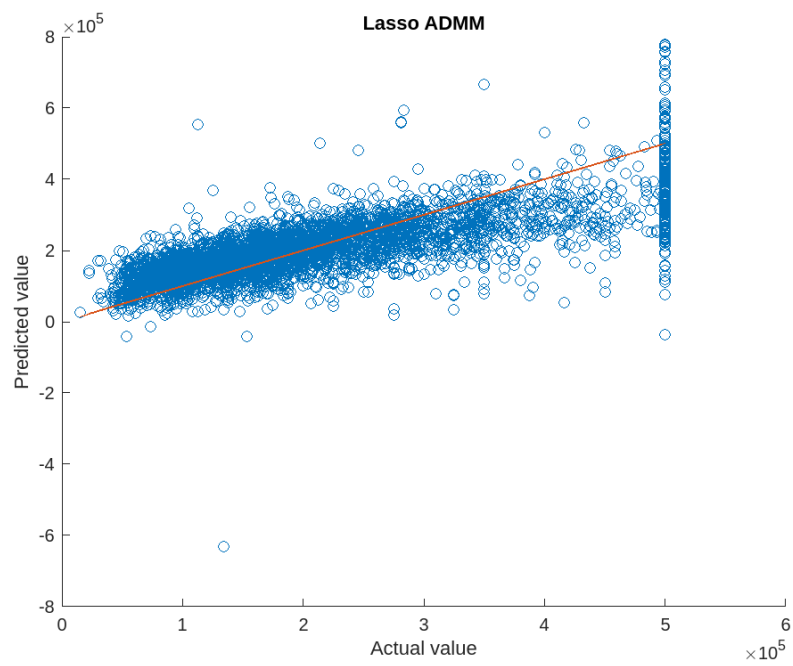


Figura 3: Predizioni ottenute con ADMM.

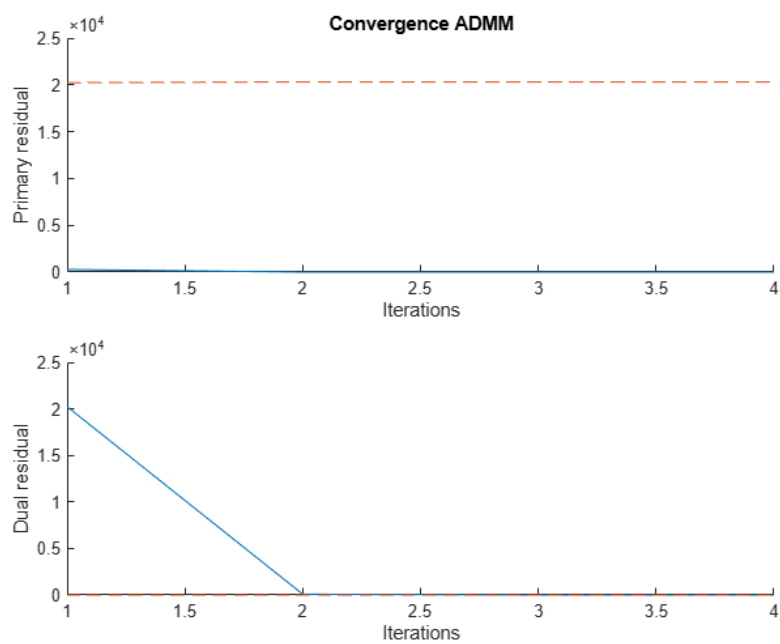


Figura 4: Convergenza di ADMM.



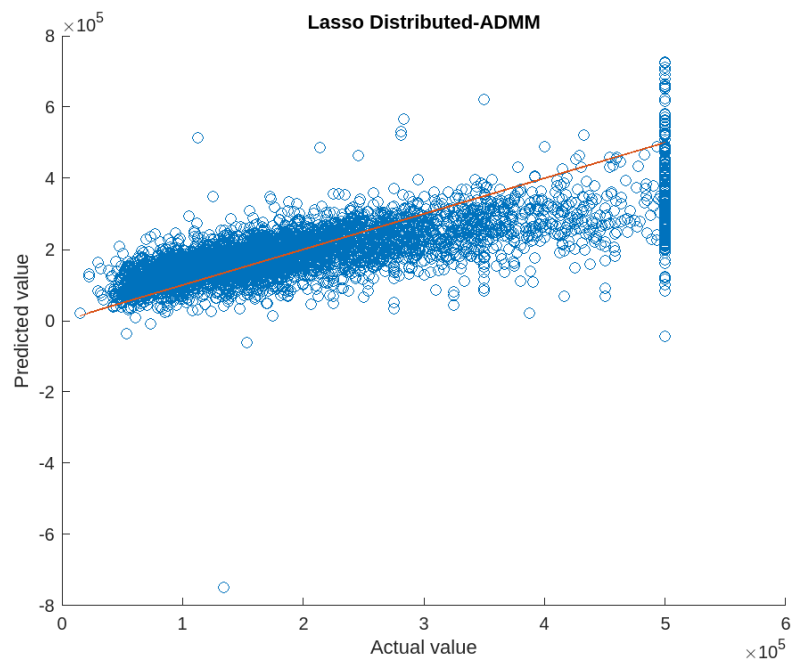


Figura 5: Predizioni ottenute con ADMM distribuito.

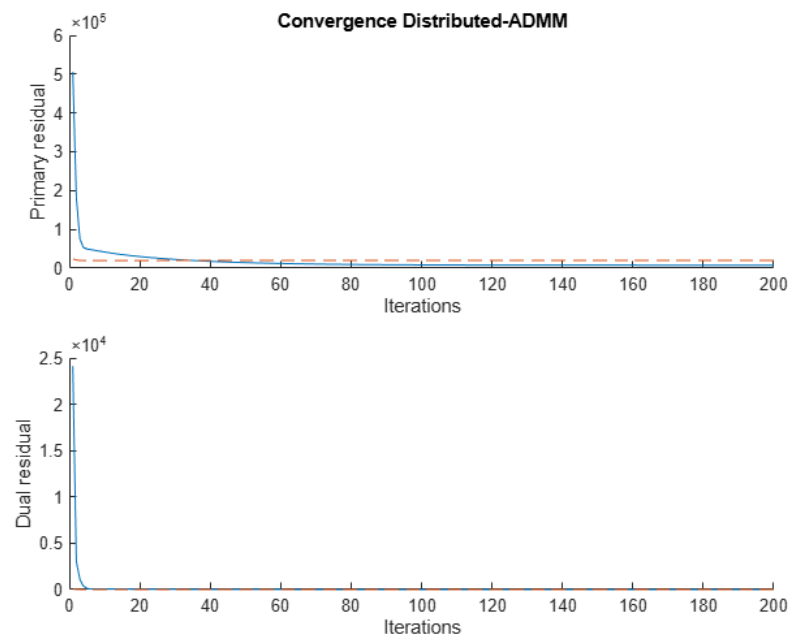


Figura 6: Convergenza di ADMM distribuito.