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

Lasso Regression Algorithms Comparison from scratch

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

1 Introduzione

Il progetto svolto consiste nell'andare a risolvere il problema Lasso tramite diversi algoritmi. È stato risolto in tre modi diversi tramite l'implementazione in Matlab dell'algoritmo ISTA, ADMM ed infine la versione simulata di ADMM distribuito su più agenti. Nella sezione finale sono presenti delle comparazioni dei 3 algoritmi nei tempi di calcolo, numero di iterazioni impiegate e i grafici che illustrano le condizioni di convergenza durante le varie iterazioni.

2 Dataset

Il dataset utilizzato è il California Housing, un insieme di dati pubblici che contiene 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;

Nel 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 successivamente descritti sono stati implementati nella classe `LassoReg`. In base ai parametri passati all'oggetto instanziato 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*:

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

$$\text{s.t.} \quad \mathbf{x} - \mathbf{z} = 0$$

Risolviamo il problema mediante ADMM scalato:

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

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

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

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:

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

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

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

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

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

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

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

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

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

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 andiamo 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 x rispetto alla propria porzione di dati.

$$\min_{(x_1,\dots,x_N),\mathbf{z}} \sum_{i=1}^{N} \|\mathbf{y}_i - A_i\mathbf{x}_i\|_2^2 + \lambda\|\mathbf{z}\|_1$$
$$\text{s.t.} \quad \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \quad i = 1,\dots,N$$

Risolviamo il problema mediante la versione scalata dell'ADMM:

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

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

Nel passo 1 si hanno N problemi distribuibili su vari agenti con una soluzione in forma chiusa, come nel caso centralizzato.

Nel passo 2 abbiamo bisogno di tutte le variabili primali e duali per calcolare il valore globale z, quindi verrà eseguito in un fusion center.

L'aggiornamento del passo 3 invece viene 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 = 9 (ADMM distribuito);

Il confronto delle prestazioni ottenute è mostrato nella Tabella 1.

Tabella 1: Comparazione algoritmi

	R2	time (s)	iterazioni
ISTA	0.7689	2.09	32417
ADMM	0.7688	3.94e-4	3
ADMM-Dist	0.7692	0.018	186

Da notare che i valori presenti nella Tabella 1 sono stati calcolati considerando uno split randomico del dataset in train set e test set. Esecuzioni diverse possono portare a risultati diversi rispetto a quelli mostrati. Successivamente sono mostrate le variazioni dei criteri di convergenza durante le iterazioni degli algoritmi ed i relativi plot delle predizioni effettuate rispetto ai valori reali.

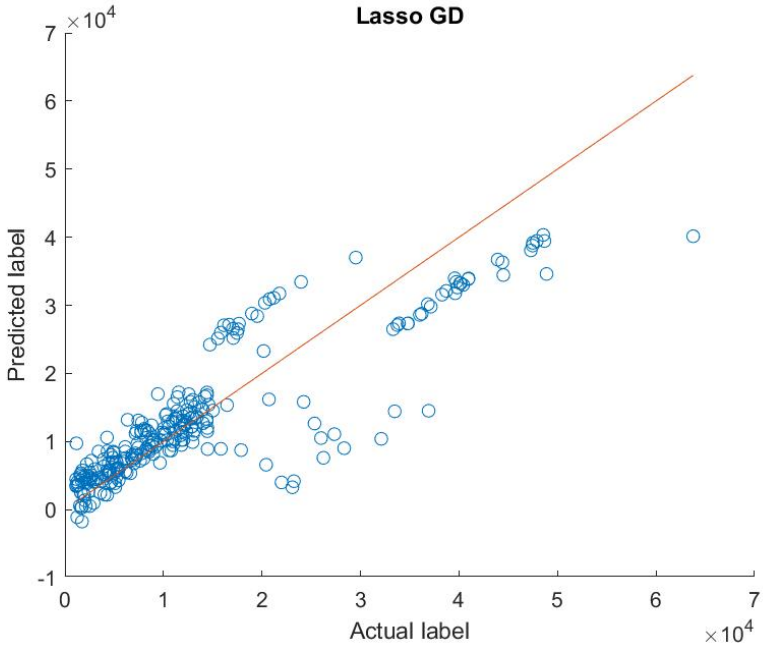
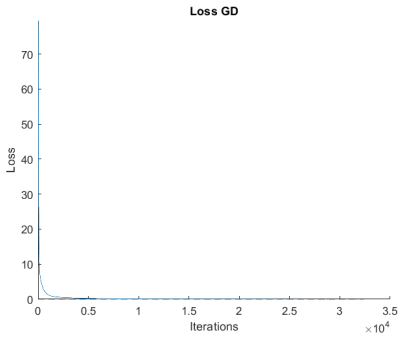
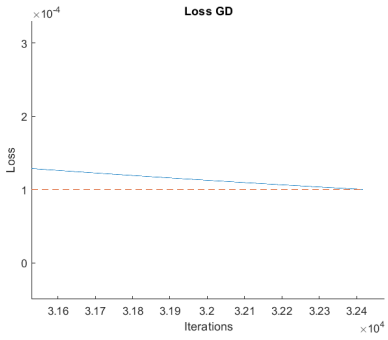


Figura 1: Predizioni ottenute con Soft-Thresholding.



(a) Convergenza del Soft-Thresholding.



(b) Convergenza del Soft-Thresholding con zoom nel cambio della condizione.

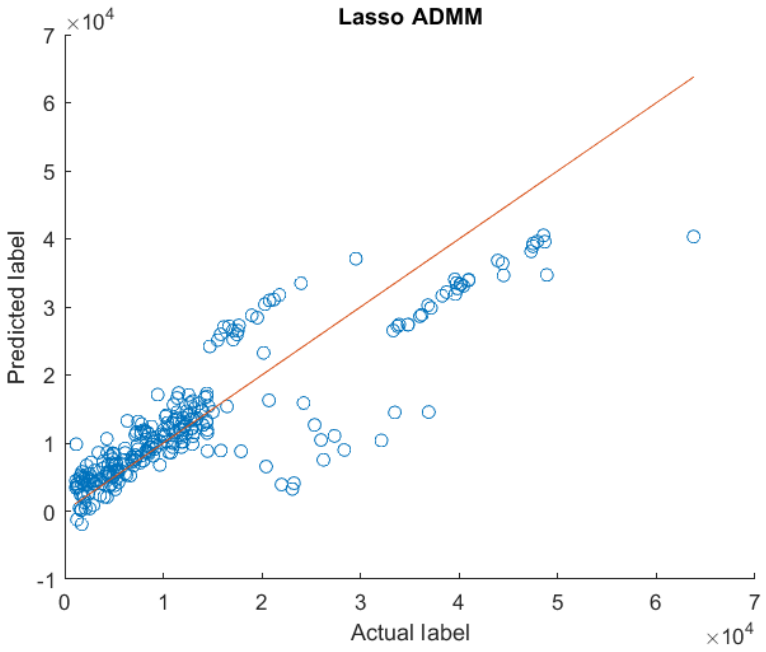
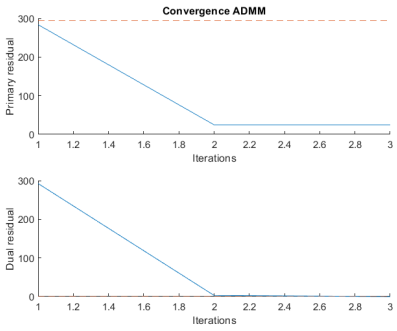
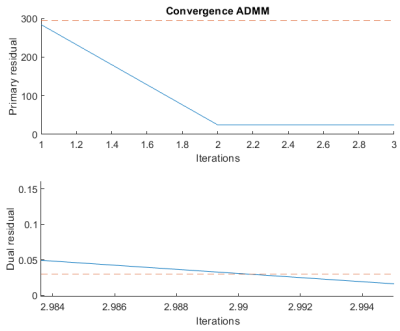


Figura 2: Predizioni ottenute con ADMM.



(a) Convergenza di ADMM.



(b) Convergenza di ADMM con zoom nel cambio della condizione.

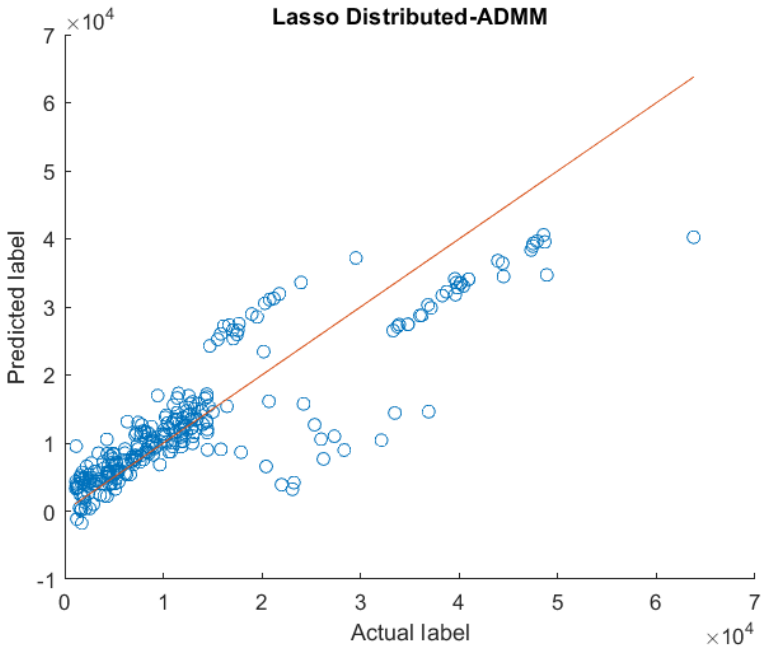
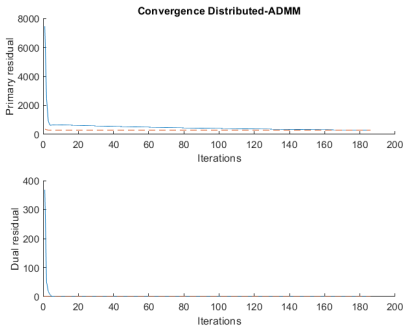
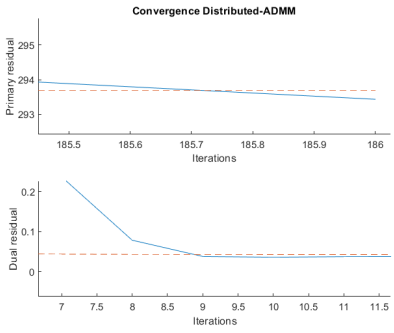


Figura 3: Predizioni ottenute con ADMM distribuito.



(a) Convergenza di ADMM distribuito.



(b) Convergenza di ADMM distribuito con zoom nel cambio della condizione.