

Tesina Signal Processing & Optimization for Big Data

Michele Baglioni 330085

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 stato preso sulla piattaforma Kaggle ed è reperibile [qui](#). Brevemente, dai dati vogliamo stimare il costo di una fatturazione medica da parte di un'agenzia assicurativa rispetto ad un individuo. I parametri a disposizione sono i seguenti:

- sesso;
- indice di massa corporea;
- numero di figli coperti dalla stessa assicurazione;
- fumatore;
- regione di residenza;

Partendo dal dataset originale sono state effettuate delle manipolazioni dei dati come normalizzazione nell'intervallo $[0,1]$, pulizia dei valori nulli e trasformazioni delle variabili categoriche in dummy.

3 Architettura ed algoritmi

Gli algoritmi successivamente descritti sono stati implementati nella 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:

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

In questo algoritmo il problema viene risolto attraverso iterazioni di gradient descent fino ad arrivare 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.

$$\nabla \underline{x} = -\frac{2}{m} * A(\underline{y} - A\underline{x}) + \mathcal{S}(\underline{x}, \alpha) \quad \text{dove } \mathcal{S}(x, \alpha) = \begin{cases} x - \alpha & \text{if } x > \alpha \\ x + \alpha & \text{if } x < -\alpha \\ 0 & \text{else} \end{cases}$$

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

Il criterio di convergenza è il seguente:

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

3.2 ADMM

Iniziamo riformulando il problema originale introducendo la slack variable:

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

Risolviemo il problema mediante ADMM scalato:

$$\begin{aligned} \underline{x}^{(k+1)} &= \underset{\underline{x}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|A\underline{x} - y\|_2^2 + \frac{\rho}{2} \|\underline{x} - \underline{z}^{(k)} + \underline{u}^{(k)}\|_2^2 \right\} \\ \underline{z}^{(k+1)} &= \underset{\underline{z}}{\operatorname{argmin}} \left\{ \alpha \|\underline{z}\|_1 + \frac{\rho}{2} \|\underline{x}^{(k+1)} - \underline{z} + \underline{u}^{(k)}\|_2^2 \right\} \\ \underline{u}^{(k+1)} &= \underline{u}^{(0)} + \sum_{i=1}^{k+1} \|\underline{x}^i - \underline{z}^i\|_2^2 \end{aligned}$$

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

$$\begin{aligned} \nabla_x \left(\frac{1}{2} \|A\underline{x} - y\|_2^2 + \frac{\rho}{2} \|\underline{x} - \underline{z}^{(k)} + \underline{u}^{(k)}\|_2^2 \right) &= \underline{0} \\ A^T (A\underline{x} - y) + \rho(\underline{x} - \underline{z}^{(k)} + \underline{u}^{(k)}) &= \underline{0} \\ (A^T A + \rho I) \underline{x} &= A^T y + \rho(\underline{z}^{(k)} - \underline{u}^{(k)}) \\ \rightarrow \underline{x}^{(k+1)} &= (A^T A + \rho I)^{-1} (A^T y + \rho(\underline{z}^{(k)} - \underline{u}^{(k)})) \end{aligned}$$

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:

$$\begin{aligned} \underline{x}^{(k+1)} &= (A^T A + \rho I)^{-1} (A^T y + \rho(\underline{z}^{(k)} - \underline{u}^{(k)})) \\ \underline{z}^{(k+1)} &= \mathcal{S}_{\frac{\alpha}{\rho}}(\underline{x}^{(k+1)} + \underline{u}^{(k)}) \\ \underline{u}^{(k+1)} &= \underline{u}^{(k)} + \underline{x}^{(k+1)} - \underline{z}^{(k+1)} \end{aligned}$$

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

$r = \ \underline{x} - \underline{z}\ $	residuo primale
$s = \ \rho(\underline{z}^{(k+1)} - \underline{z}^{(k)})\ $	residuo duale
$\epsilon_r = \sqrt{n}\epsilon + \epsilon_{rel} \max(\ \underline{x}\ , \ \underline{z}\)$	tolleranza primale
$\epsilon_s = \sqrt{n}\epsilon + \epsilon_{rel} \ \rho u\ $	tolleranza duale
$\text{if } (r < \epsilon_r) \& (s < \epsilon_s) \rightarrow \text{break}$	criterio di convergenza

3.3 Distributed ADMM

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

$$\underline{y} = \begin{bmatrix} y_1 \\ \dots \\ y_N \end{bmatrix} \quad A\underline{x} = \begin{bmatrix} A_1 \\ \dots \\ A_N \end{bmatrix} \underline{x} = \begin{bmatrix} A_1 \underline{x} \\ \dots \\ A_N \underline{x} \end{bmatrix}$$

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.

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

Risolviamo il problema mediante la versione scalate dell'ADMM:

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

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.

	R2	time (s)	iterazioni
ISTA	0.7689	2,09	32417
ADMM	0.7688	3,94e-4	3
ADMM-Dist	0.7692	0.018	186

Tabella 1: Comparazione algoritmi.

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 diversi risultati 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.

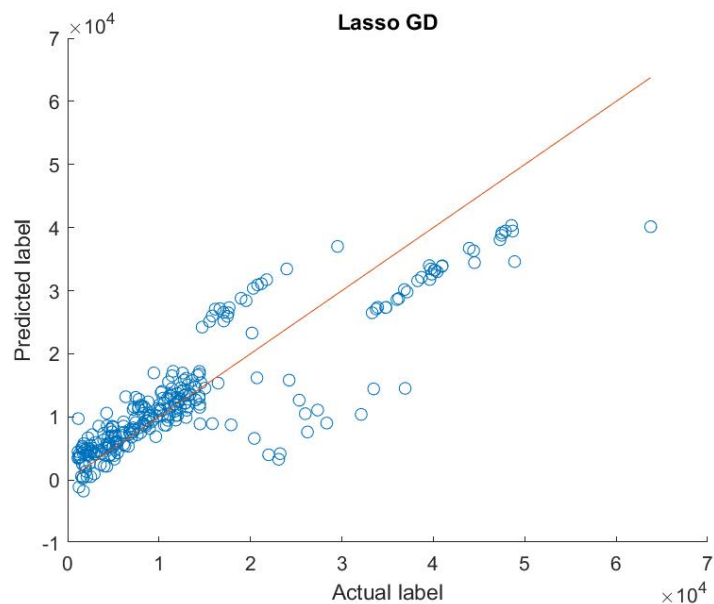
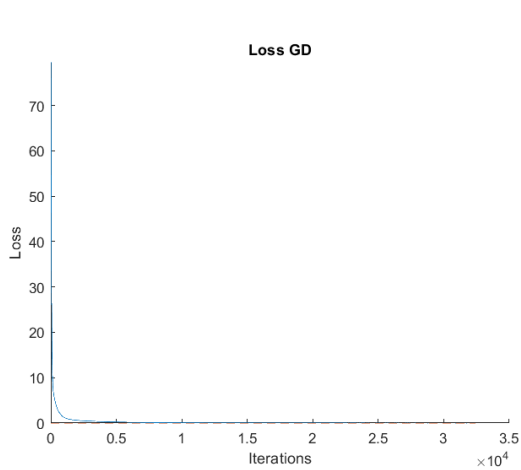
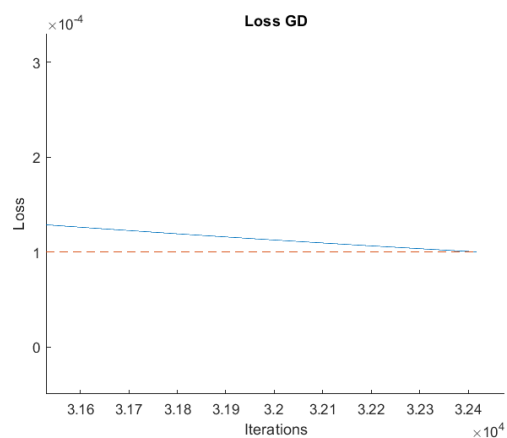


Figura 1: Predizioni ottenute con Soft-Thresholding.



(a) Convergenza del Soft-Thresholding.



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

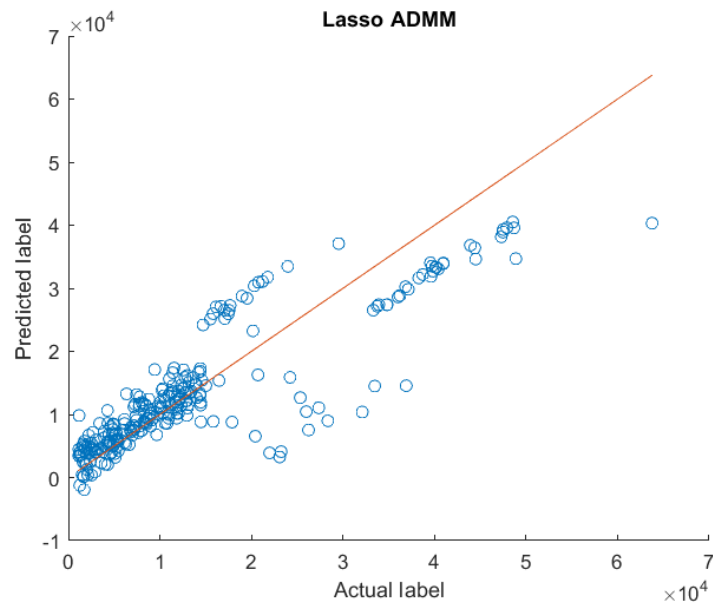
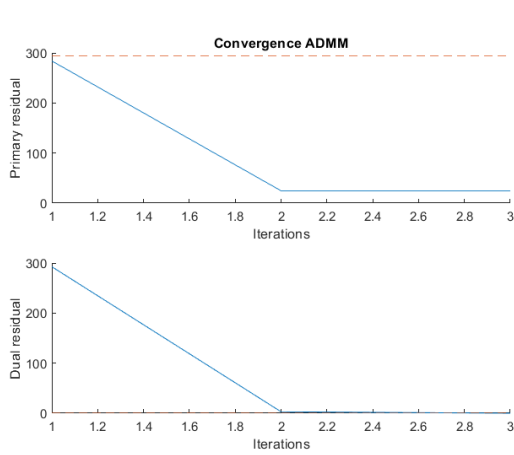
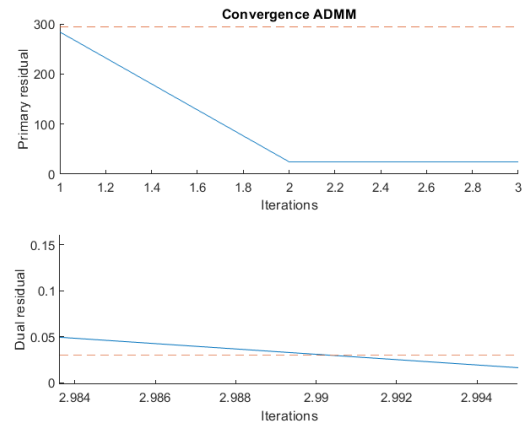


Figura 3: Predizioni ottenute con ADMM.



(a) Convergenza ADMM.



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

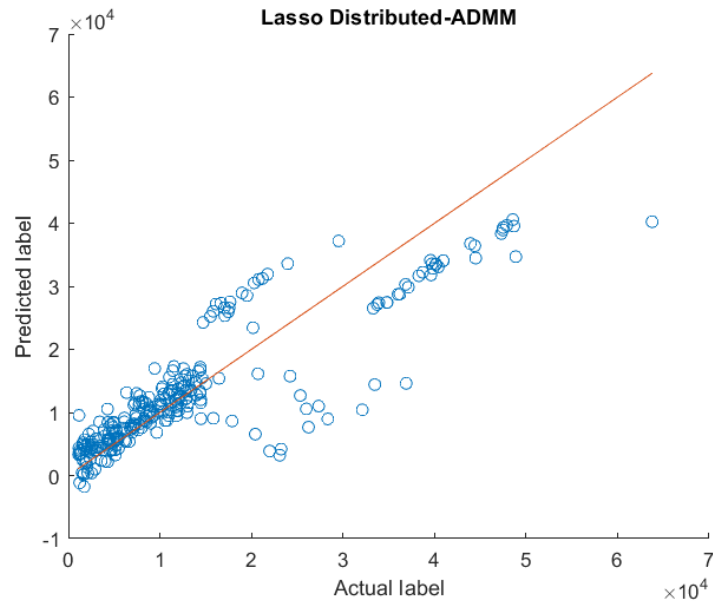
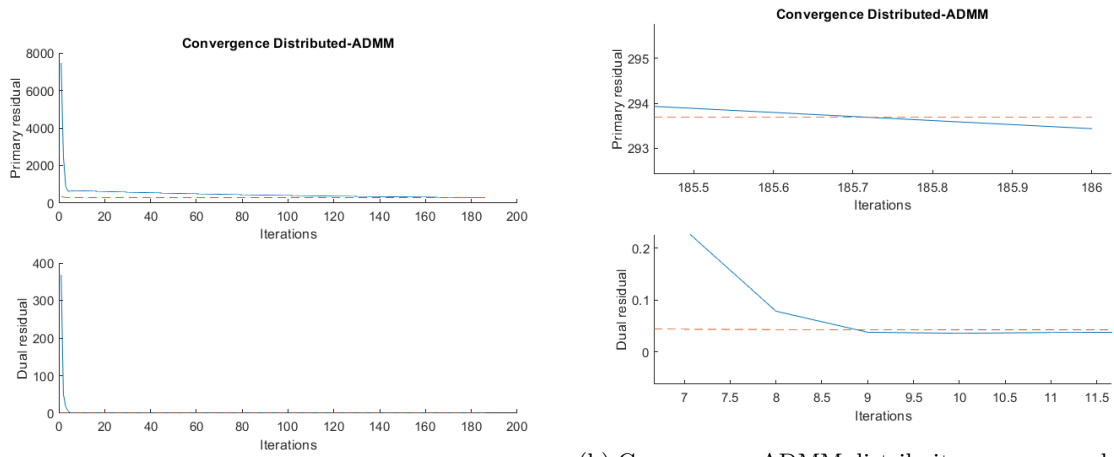


Figura 5: Predizioni ottenute con ADMM distribuito.



(a) Convergenza ADMM distribuito.

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