

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Matematica

Metodi matching pursuit per la classificazione di immagini

Tesi di Laurea in Matematica Computazionale

Relatrice:
Chiar.ma Prof.ssa
Valeria Simoncini

Presentata da:
Jacopo Betti

Anno Accademico 2022-2023

Introduzione

Il campo della rappresentazione sparsa è un campo di studio nato di recente che ha attratto molto interesse in questi anni. Alla base di tale campo c'è un problema da risolvere: data $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m < n$ di rango massimo, si vuole cercare la soluzione più sparsa possibile tra le infinite soluzioni del sistema sottodeterminato $\mathbf{Ax} = \mathbf{b}$.

I primi risultati importanti nel campo della rappresentazione sparsa si possono trovare in un articolo di Stephane Mallat e Zhifeng Zhang del 1993 che introduce i concetti di dizionario, mutua coerenza e tecniche pursuit greedy [1].

Un altro contributo importante è stato dato da Scott Shaobing Chen, David L. Donoho, e Michael Saunders che nel 1998 hanno dimostrato in un articolo che la ricerca della soluzione più sparsa del sistema può essere visto come un problema di ottimizzazione convessa che può essere risolto dal metodo Basic Pursuit (BP) [2].

Infine, nel 2003 David L. Donoho e Michael Elad hanno definito il concetto di spark e dimostrato il collegamento tra spark e unicità della soluzione più sparsa [3].

In questa tesi si studieranno alcuni di questi metodi e teoremi. In particolare, questa tesi è costituita da 4 capitoli:

- nel primo capitolo verranno enunciati teoremi e definizioni necessarie per questa tesi;
- nel secondo capitolo si parlerà dell'ortogonalizzazione di Gram-Schmidt e delle sue applicazioni nella risoluzione di problemi ai minimi quadrati;
- il terzo capitolo è un'introduzione al campo dei metodi matching pursuit, con teoremi riguardo l'unicità ed esistenza di soluzioni sparse e descrizione dei metodi MP e OMP;
- nel quarto capitolo verranno mostrate le applicazioni dei metodi matching pursuit per la classificazione di immagini usando tre data set diversi.

Notazioni

In questa tesi verranno usate le seguenti notazioni:

- $\mathbf{v} \in \mathbb{R}^n$ è un vettore. La i -esima componente di \mathbf{v} è v_i ;
- $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ è una matrice le cui colonne sono i vettori $\mathbf{a}_1, \dots, \mathbf{a}_n$.
L'elemento nella posizione i, j di \mathbf{A} è $\mathbf{A}_{i,j}$;
- \mathbf{A}^T è la trasposta della matrice \mathbf{A} ;
- $\mathbf{0}$ è la matrice in cui tutte le componenti sono zero;
- \mathbf{I} è la matrice identità;
- $\text{spec}(\mathbf{A})$ è lo spettro di una matrice \mathbf{A} ;
- $\text{rank}(\mathbf{A})$ è il rango di una matrice \mathbf{A} ;
- il prodotto scalare di due vettori \mathbf{x} e \mathbf{y} con le stesse dimensioni è $\mathbf{x} \cdot \mathbf{y}$ o $\mathbf{x}^T \mathbf{y}$.

Tutti i valori considerati sono reali.

Indice

Introduzione	i
Notazioni	iii
1 Definizioni preliminari	1
2 Ortogonalizzazione di Gram-Schmidt ed applicazioni	3
2.1 Ortogonalizzazione di Gram-Schmidt	3
2.1.1 Procedura modificata di Gram-Schmidt	5
2.2 Proprietà della fattorizzazione QR	6
2.3 Applicazioni: problema ai minimi quadrati a rango massimo	6
3 Metodi matching pursuit per la rappresentazione sparsa	11
3.1 Il problema della rappresentazione sparsa	11
3.2 Rappresentazione sparsa con minimizzazione rispetto alla norma l_0	12
3.2.1 Unicità della soluzione mediante spark	12
3.2.2 Unicità della soluzione mediante mutua coerenza	13
3.3 Algoritmi MP e OMP	14
3.4 Condizione di recupero esatto	19
4 Applicazioni dei metodi matching pursuit	23
4.1 Il problema della classificazione di immagini	23
4.2 Illustrazioni computazionali	24
4.3 Conclusioni	27
Bibliografia	29

Capitolo 1

Definizioni preliminari

Definizione 1.1. La precisione macchina μ di un calcolatore è il numero positivo più piccolo che sommato ad 1 col calcolatore da un numero diverso da 1.

Definizione 1.2. Sia $\mathbf{v} \in \mathbb{R}^n$ un vettore, $p \in \mathbb{R}$ tale che $0 < p < +\infty$. Si definisce la norma l_p del vettore come

$$\|\mathbf{v}\|_p = \left(\sum_{i=1}^n |v_i|^p \right)^{\frac{1}{p}}.$$

Osservazione 1.3. Per $0 < p < 1$, la norma l_p appena definita non è una norma ma una seminorma siccome non vale la proprietà della disuguaglianza triangolare.

Definizione 1.4. Sia $\mathbf{v} \in \mathbb{R}^n$ un vettore. Si definisce la 'norma' l_0 del vettore come il seguente limite:

$$\|\mathbf{v}\|_0 = \lim_{p \rightarrow 0} \|\mathbf{v}\|_p^p = \lim_{p \rightarrow 0} \sum_{i=1}^n |v_i|^p = \#\{i : v_i \neq 0\}.$$

Osservazione 1.5. La 'norma' l_0 appena definita non è una norma siccome non vale la seguente proprietà: $\|\lambda \mathbf{x}\|_0 = |\lambda| \|\mathbf{x}\|_0$ per ogni scalare λ , per ogni vettore \mathbf{x} . Da ora in poi ci riferiremo alla 'norma' l_0 omettendo le virgolette.

Definizione 1.6. Sia $\mathbf{v} \in \mathbb{R}^n$ un vettore. Si definisce la norma l_∞ del vettore nel seguente modo:

$$\|\mathbf{v}\|_\infty = \max_{i \in \{1, \dots, n\}} |v_i|.$$

Definizione 1.7. Sia $\|\cdot\|$ una norma vettoriale. La norma matriciale indotta da tale norma vettoriale della matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ è il numero:

$$\|\mathbf{A}\| = \max_{\|\mathbf{x}\| \leq 1} \|\mathbf{A}\mathbf{x}\| = \max_{\|\mathbf{x}\| \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}.$$

Definizione 1.8. Sia $\mathbf{A} \in \mathbb{R}^{m \times n}$ di rango massimo. La pseudo-inversa della matrice \mathbf{A} è l'unica matrice \mathbf{A}^+ che soddisfa le quattro condizioni di Moore-Penrose:

$$(1) \mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}; \quad (3) (\mathbf{A}\mathbf{A}^+)^T = \mathbf{A}\mathbf{A}^+;$$

$$(2) \mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+; \quad (4) (\mathbf{A}^+\mathbf{A})^T = \mathbf{A}^+\mathbf{A}.$$

Teorema 1.9. Sia $\mathbf{A} \in \mathbb{R}^{m \times n}$ una matrice. $\mathbf{A}^T\mathbf{A}$ è definita positiva se e solo se \mathbf{A} ha rango massimo.

Teorema 1.10 (Unicità ed esistenza della fattorizzazione di Cholesky). Sia $\mathbf{A} \in \mathbb{R}^{n \times n}$ una matrice simmetrica e definita positiva. Allora esiste ed è unica la fattorizzazione $\mathbf{A} = \mathbf{L}\mathbf{L}^T$, dove \mathbf{L} è triangolare inferiore non singolare con $L_{i,i} > 0$.

Definizione 1.11. Siano $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{R}^n$. La matrice di Gram di tale insieme è la matrice il cui elemento in posizione i, j è $\mathbf{v}_i \cdot \mathbf{v}_j$.

Definizione 1.12. Sia $\mathbf{A} \in \mathbb{R}^{n \times n}$. \mathbf{A} è una matrice a dominanza diagonale stretta per righe se

$$|\mathbf{A}_{i,i}| > \sum_{j=1, j \neq i}^n |\mathbf{A}_{i,j}|, \quad i = 1, \dots, n.$$

Definizione 1.13 (Cerchi di Gerschgorin). Sia $\mathbf{A} \in \mathbb{C}^{n \times n}$. I cerchi del piano complesso

$$\mathcal{G}_1^{(r)} = \left\{ z \in \mathbb{C} \text{ t.c. } |z - \mathbf{A}_{i,i}| \leq \sum_{j=1, j \neq i}^n |\mathbf{A}_{i,j}| \right\}, \quad i = 1, \dots, n$$

di centro $\mathbf{A}_{i,i}$ e raggio $\sum_{j=1, j \neq i}^n |\mathbf{A}_{i,j}|$ sono detti cerchi di Gerschgorin per righe.

Teorema 1.14 (Primo teorema di Gerschgorin). Sia $\mathbf{A} \in \mathbb{C}^{n \times n}$. Allora

$$\text{spec}(\mathbf{A}) \subset \bigcup_{i=1}^n \mathcal{G}_i^{(r)},$$

cioè gli autovalori di \mathbf{A} sono contenuti nell'unione dei cerchi di Gerschgorin per righe.

Osservazione 1.15. Una conseguenza di questo teorema è che una matrice a dominanza diagonale stretta per righe che ha solo valori positivi sulla diagonale è definita positiva.

Definizione 1.16. Sia $\mathbf{v} \in \mathbb{R}^n$. Diciamo che il vettore è sparso se alcune sue componenti sono uguali a zero.

Capitolo 2

Ortogonalizzazione di Gram-Schmidt ed applicazioni

Per scrivere questo capitolo sono state usate le dispense del corso di Calcolo Numerico [4] e il libro "*Matrix Computations*" [5].

Consideriamo il sistema lineare:

$$\mathbf{B}\mathbf{x} = \mathbf{b},$$

con $\mathbf{B} \in \mathbb{R}^{n \times n}$ non singolare, $\mathbf{b} \in \mathbb{R}^n$. Una tipologia di metodi risolutivi per sistemi lineari consiste nel determinare una fattorizzazione della matrice \mathbf{B} del tipo $\mathbf{B} = \mathbf{C}\mathbf{D}$, per poi risolvere i seguenti sistemi in ordine: $\mathbf{C}\mathbf{y} = \mathbf{b}$, $\mathbf{D}\mathbf{x} = \mathbf{y}$. Tali metodi risolutivi sono efficaci fintanto che le matrici \mathbf{C} e \mathbf{D} ottenute dalla fattorizzazione hanno una forma che rendano facile e veloce risolvere i sistemi lineari appena enunciati. Un esempio di tale metodo è la fattorizzazione QR ridotta che si può ottenere mediante il processo di ortogonalizzazione di Gram-Schmidt.

2.1 Ortogonalizzazione di Gram-Schmidt

Supponiamo di avere una base di un spazio vettoriale di \mathbb{R}^n di grado m . Sia $\{\mathbf{a}_1, \dots, \mathbf{a}_m\}$ una base di tale spazio. L'ortogonalizzazione di Gram-Schmidt permette di ottenere una base ortonormale $\{\mathbf{q}_1, \dots, \mathbf{q}_m\}$ dalla base iniziale. L'algoritmo di Gram-Schmidt per

ottenere una base ortonormale $\{\mathbf{q}_1, \dots, \mathbf{q}_m\}$ è il seguente:

$$\begin{aligned}\mathbf{q}_1 &= \mathbf{a}_1 / \|\mathbf{a}_1\|_2 \\ \tilde{\mathbf{q}}_2 &= \mathbf{a}_2 - \mathbf{q}_1(\mathbf{q}_1^T \mathbf{a}_2) \\ \mathbf{q}_2 &= \tilde{\mathbf{q}}_2 / \|\tilde{\mathbf{q}}_2\|_2 \\ \tilde{\mathbf{q}}_3 &= \mathbf{a}_3 - \mathbf{q}_1(\mathbf{q}_1^T \mathbf{a}_3) - \mathbf{q}_2(\mathbf{q}_2^T \mathbf{a}_3) \\ &\vdots \quad \quad \quad \vdots\end{aligned}$$

Da cui si ha:

$$\begin{aligned}\mathbf{a}_1 &= \mathbf{q}_1 \|\mathbf{a}_1\|_2 \\ \mathbf{a}_2 &= \mathbf{q}_1(\mathbf{q}_1^T \mathbf{a}_2) + \mathbf{q}_2 \|\tilde{\mathbf{q}}_2\|_2 \\ \mathbf{a}_3 &= \mathbf{q}_1(\mathbf{q}_1^T \mathbf{a}_3) + \mathbf{q}_2(\mathbf{q}_2^T \mathbf{a}_3) + \mathbf{q}_3 \|\tilde{\mathbf{q}}_3\|_2 \\ &\vdots \quad \quad \quad \vdots\end{aligned}$$

In forma matriciale si ha quindi:

$$[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3 \dots] = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \dots] \begin{bmatrix} \|\mathbf{a}_1\|_2 & \mathbf{q}_1^T \mathbf{a}_2 & \mathbf{q}_1^T \mathbf{a}_3 & \dots \\ 0 & \|\tilde{\mathbf{q}}_2\|_2 & \mathbf{q}_2^T \mathbf{a}_3 & \dots \\ 0 & 0 & \|\tilde{\mathbf{q}}_3\|_2 & \dots \\ 0 & 0 & 0 & \ddots \end{bmatrix} = \mathbf{Q}_1 \mathbf{R}_1.$$

L'insieme di vettori $\{\mathbf{q}_1, \dots, \mathbf{q}_n\}$ ottenuto è un insieme ortonormale di vettori. Infatti, dati due vettori $\mathbf{v}, \mathbf{q} \in \mathbb{R}^n$, con \mathbf{q} di norma unitaria, l'operazione $\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{q}(\mathbf{q}^T \mathbf{v})$ elimina dal vettore \mathbf{v} la direzione del vettore \mathbf{q} , quindi si avrà $\mathbf{q}^T \tilde{\mathbf{v}} = 0$.

Supponiamo ora di avere una matrice $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m \geq n$ di rango massimo. Si può usare la procedura appena illustrata per determinare due matrici \mathbf{Q}_1 e \mathbf{R}_1 tale che:

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1, \quad (2.1)$$

con $\mathbf{Q}_1 \in \mathbb{R}^{m \times n}$ matrice ortogonale e $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$ matrice triangolare superiore.

In aritmetica esatta, tale procedura restituisce una matrice ortogonale. In aritmetica floating point, se le norme dei \mathbf{q}_i sono dell'ordine di $O(\mu)$, dove μ è la precisione macchina, la matrice \mathbf{A} numericamente ha rango inferiore a n . In particolare, se per un certo j si ha $\|\tilde{\mathbf{q}}_j\| = 0$, la procedura si interrompe poichè non riuscirà a determinare \mathbf{q}_j . Inoltre, anche se il processo non si interrompe, in aritmetica floating point la matrice \mathbf{Q}_1 potrebbe

essere non ortogonale. Infatti, usando il metodo di Gram-Schmidt classico si può avere una perdita di ortogonalità tra i vettori colonna di \mathbf{Q}_1 . Un esempio è la seguente matrice:

$$\mathbf{A} = \alpha \mathbf{I} + \mathbf{H},$$

con $n = 200$, $\alpha = 0.00001$, $\mathbf{I} \in \mathbb{R}^{n \times n}$ la matrice identità, $\mathbf{H} \in \mathbb{R}^{n \times n}$ la matrice di Hilbert che è definita nel seguente modo: $(\mathbf{H}_{i,j})_{i,j=1,\dots,n} = (\frac{1}{i+j-1})$. Sia \mathbf{Q}_1 la matrice ottenuta usando il metodo classico di Gram-Schmidt, si ha che:

$$\|\mathbf{I} - \mathbf{Q}_1^T \mathbf{Q}_1\|_2 = 0.5461,$$

il che significa che le colonne della matrice \mathbf{Q}_1 non sono più ortogonali tra di loro. Ciò è dovuto all'accumulazione di errori di arrotondamento.

Per ovviare a questi problemi, è preferibile usare la procedura modificata di Gram-Schmidt.

2.1.1 Procedura modificata di Gram-Schmidt

Un riordinamento delle operazioni del metodo classico di Gram-Schmidt permette di ottenere un algoritmo che è più stabile dal punto di vista numerico rispetto al metodo classico. Tale algoritmo è l'algoritmo modificato di Gram-Schmidt (MGS). I passaggi del metodo MGS sono riassunti di seguito.

Algorithm 1. Algoritmo modificato di Gram-Schmidt

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$ di rango massimo, $m \geq n$.

Inizializzazioni: $\mathbf{R}_{1,1} = \|\mathbf{a}_1\|_2$, $\mathbf{q}_1 = \mathbf{a}_1 / \mathbf{R}_{1,1}$.

Corpo dell'algoritmo: per $j = 2, \dots, n$

$$\hat{\mathbf{q}} = \mathbf{a}_j$$

per $i = 1, \dots, j - 1$

$$\mathbf{R}_{i,j} = \hat{\mathbf{q}} \cdot \mathbf{q}_i$$

$$\hat{\mathbf{q}} = \hat{\mathbf{q}} - \mathbf{R}_{i,j} \mathbf{q}_i$$

$$\mathbf{R}_{j,j} = \|\hat{\mathbf{q}}\|_2$$

$$\mathbf{q}_j = \hat{\mathbf{q}} / \mathbf{R}_{j,j}$$

Output: \mathbf{Q}, \mathbf{R} .

Riprendendo la matrice \mathbf{A} definita precedentemente, sia \mathbf{Q}_1 la matrice ottenuta dalla fattorizzazione QR di \mathbf{A} usando il metodo di Gram-Schmidt modificato. Si ha che:

$$\|\mathbf{I} - \mathbf{Q}_1^T \mathbf{Q}_1\|_2 = 1.2189 \cdot 10^{-11},$$

perciò rispetto al metodo classico tale procedura modificata è più accurata dal punto di vista numerico.

2.2 Proprietà della fattorizzazione QR

Per quanto dimostrato nella sezione precedente, sia $\mathbf{A} \in \mathbb{R}^{m \times n}$ di rango massimo con $m \geq n$. Allora esistono una matrice \mathbf{R}_1 triangolare superiore, $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$, e una matrice \mathbf{Q}_1 con colonne ortonormali, $\mathbf{Q}_1 \in \mathbb{R}^{m \times n}$, tale che:

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1.$$

In particolare, l'algoritmo di Gram-Schmidt restituisce una fattorizzazione QR ridotta poichè, rispetto alla normale fattorizzazione QR, non vengono generate completamente le matrici $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2] \in \mathbb{R}^{m \times m}$, $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{m \times n}$, ma solo $\mathbf{Q}_1 \in \mathbb{R}^{m \times n}$, $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$. Inoltre, valgono i seguenti teoremi.

Teorema 2.1. *Sia $\mathbf{A} \in \mathbb{R}^{m \times n}$ di rango massimo, $m \geq n$. Se $\mathbf{A} = \mathbf{QR}$ è la fattorizzazione QR di una matrice a rango pieno e $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_m]$, allora:*

$$\text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\} = \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}$$

per $k = 1, \dots, n$.

Dimostrazione. Considerando l'ortogonalizzazione di Gram-Schmidt, si ha che

$$\mathbf{a}_k = \sum_{i=1}^k \mathbf{R}_{i,k} \mathbf{q}_i \in \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}.$$

Quindi, $\text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\} \subseteq \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}$. Siccome \mathbf{A} ha rango n , segue che $\text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\}$ ha dimensione k e perciò deve essere uguale a $\text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}$. \square

Teorema 2.2. *Sia $\mathbf{A} \in \mathbb{R}^{m \times n}$ di rango massimo, $m \geq n$. Allora la fattorizzazione QR ridotta $\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1$ è unica se tutti gli elementi sulla diagonale di \mathbf{R}_1 sono positivi.*

Dimostrazione. Si ha che $\mathbf{A}^T \mathbf{A} = (\mathbf{Q}_1 \mathbf{R}_1)^T (\mathbf{Q}_1 \mathbf{R}_1) = \mathbf{R}_1^T \mathbf{R}_1$. Siccome \mathbf{A} ha rango massimo, $\mathbf{A}^T \mathbf{A}$ è una matrice simmetrica definita positiva. Essendo $\mathbf{R}_1^T \mathbf{R}_1$ la fattorizzazione di Cholesky di $\mathbf{A}^T \mathbf{A}$, per l'unicità della fattorizzazione di Cholesky, \mathbf{R}_1 è unica. Dato che $\mathbf{Q}_1 = \mathbf{A} \mathbf{R}_1^{-1}$, anche \mathbf{Q}_1 è unica, da cui la tesi. \square

2.3 Applicazioni: problema ai minimi quadrati a rango massimo

Un problema molto comune in vari ambiti applicativi è il problema ai minimi quadrati. Data una matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m \geq n$ di rango massimo e un vettore $\mathbf{b} \in \mathbb{R}^m$, il

sistema $\mathbf{Ax} = \mathbf{b}$ è sovradeterminato, ossia il numero di equazioni è superiore al numero di incognite. In generale un sistema sovradeterminato non ha una soluzione esatta.

L'obiettivo quindi è scegliere il vettore $\mathbf{x} \in \mathbb{R}^n$ che minimizzi il residuo $\mathbf{r} := \mathbf{b} - \mathbf{Ax}$ rispetto ad una norma prestabilita. In particolare, il problema ai minimi quadrati consiste nel minimizzare il residuo rispetto alla norma l_2 , quindi il problema da risolvere è il seguente:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{b} - \mathbf{Ax}\|_2. \quad (2.2)$$

Un metodo risolutivo consiste nel trasformare tale problema in un sistema lineare detto equazione normale che ha solo una soluzione nel caso \mathbf{A} abbia rango massimo. Per poter parlare di equazione normale ci serve dimostrare questo lemma.

Lemma 2.3. *Una matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$ con $m \geq n$ ha rango massimo se e solo se la matrice $\mathbf{A}^T \mathbf{A}$ non è singolare.*

Dimostrazione. Sia $\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1$ la fattorizzazione QR ridotta di \mathbf{A} . Dalla procedura di Gram-Schmidt segue che le colonne di \mathbf{A} sono linearmente indipendenti se e solo se la matrice \mathbf{R}_1 è non singolare. Inoltre si ha che $\mathbf{A}^T \mathbf{A} = \mathbf{R}_1^T \mathbf{Q}_1^T \mathbf{Q}_1 \mathbf{R}_1 = \mathbf{R}_1^T \mathbf{R}_1$. Quindi, $\mathbf{A}^T \mathbf{A}$ è non singolare se e solo se $\mathbf{R}_1^T \mathbf{R}_1$ è non singolare, cioè se e solo se \mathbf{R}_1 è non singolare. \square

Avendo dimostrato tale lemma, possiamo enunciare il seguente teorema.

Teorema 2.4. *Sia X l'insieme dei vettori $\mathbf{x} \in \mathbb{R}^n$ che soddisfa (2.2). Allora valgono le seguenti affermazioni:*

1. $\mathbf{x} \in X$ se e solo se \mathbf{x} è una soluzione dell'equazione normale

$$\mathbf{A}^T \mathbf{Ax} = \mathbf{A}^T \mathbf{b}; \quad (2.3)$$

2. X ha un solo elemento se e solo se \mathbf{A} ha rango massimo.

Dimostrazione. Siano $R = \text{range}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{R}^m \text{ tale che } \mathbf{y} = \mathbf{Ax}, \mathbf{x} \in \mathbb{R}^n\}$, $R^\perp = \{\mathbf{z} \in \mathbb{R}^m \text{ tale che } \mathbf{z}^T \mathbf{y} = 0, \mathbf{y} \in R\}$ il suo ortogonale. Si può scrivere $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$, con $\mathbf{b}_1 \in R$, $\mathbf{b}_2 \in R^\perp$.

1. Si ha $\mathbf{r} = \mathbf{b} - \mathbf{Ax} = \mathbf{b}_1 - \mathbf{Ax} + \mathbf{b}_2$, con $\mathbf{b}_1 - \mathbf{Ax} \in R$ e $\mathbf{b}_2 \in R^\perp$, quindi:

$$\|\mathbf{r}\|_2^2 = \|\mathbf{b}_1 - \mathbf{Ax}\|_2^2 + \|\mathbf{b}_2\|_2^2.$$

Per ottenere il minimo di $\|\mathbf{r}\|_2$ è sufficiente operare sul primo termine dato che il secondo non contiene \mathbf{x} . Perciò il minimo di $\|\mathbf{r}\|_2$ sarà raggiunto se e solo se esiste

\mathbf{x} tale che $\mathbf{b}_1 = \mathbf{Ax}$. Seguirà inoltre che $\mathbf{r} = \mathbf{b}_2$, quindi si avrà $\mathbf{r} \in R^\perp$. Una conseguenza diretta di questo è la seguente equazione:

$$\mathbf{0} = \mathbf{A}^T \mathbf{r} = \mathbf{A}^T (\mathbf{b} - \mathbf{Ax}) = \mathbf{A}^T \mathbf{b} - \mathbf{A}^T \mathbf{Ax}.$$

2. Per il lemma precedente, la matrice \mathbf{A} ha rango massimo se e solo se $\mathbf{A}^T \mathbf{A}$ è non singolare, cioè se e solo se l'equazione normale ha una e una sola soluzione.

□

Supponendo \mathbf{A} di rango massimo, ossia supponendo che $\mathbf{A}^T \mathbf{A}$ sia definita positiva, un metodo per risolvere (2.2) consiste nel calcolare la matrice $\mathbf{A}^T \mathbf{A}$, calcolare la matrice \mathbf{L} della fattorizzazione di Cholesky $\mathbf{A}^T \mathbf{A} = \mathbf{LL}^T$ e risolvere in ordine i seguenti sistemi triangolari:

$$\begin{aligned} \mathbf{Ly} &= \mathbf{A}^T \mathbf{b}, \\ \mathbf{L}^T \mathbf{x} &= \mathbf{y}. \end{aligned}$$

Tale metodo è costoso e non molto stabile dal punto di vista numerico. Infatti, colonne che sono "quasi" linearmente dipendenti in aritmetica esatta possono causare problemi in aritmetica con precisione finita. In tali casi, il metodo dell'equazione normale non può essere usato. Consideriamo il seguente esempio:

$$\mathbf{A} = \begin{pmatrix} 3 & 3 \\ 4 & 4 \\ 0 & \alpha \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

con $\alpha \in \mathbb{R}$ tale che $\mu \leq \alpha < 1$ e $\alpha^2 < \mu$, dove μ indica la precisione macchina. Calcolando esplicitamente $\mathbf{A}^T \mathbf{A}$ e $\mathbf{A}^T \mathbf{b}$ si ha:

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 25 & 25 \\ 25 & 25 + \alpha^2 \end{pmatrix}, \quad \mathbf{A}^T \mathbf{b} = \begin{pmatrix} 7 \\ 7 + \alpha \end{pmatrix}.$$

In aritmetica esatta la soluzione dell'equazione è $\mathbf{x} = \left(\frac{7}{25} - \frac{1}{\alpha}, \frac{1}{\alpha}\right)^T \in \mathbb{R}^n$. Dato che $\alpha^2 < \mu$, la rappresentazione di $\mathbf{A}^T \mathbf{A}$ sul calcolatore è

$$fl(\mathbf{A}^T \mathbf{A}) = \begin{pmatrix} 25 & 25 \\ 25 & 25 \end{pmatrix}$$

che ha due colonne uguali, quindi $fl(\mathbf{A}^T \mathbf{A})$ è singolare, cioè l'equazione normale in aritmetica con precisione finita non ha soluzioni. In questo caso il metodo dell'equazione normale non risolve il problema dei minimi quadrati.

Siccome il metodo dell'equazione normale non funziona in tutti i casi, è necessario un metodo alternativo per la risoluzione del problema dei minimi quadrati. Un tale metodo prevede l'uso della fattorizzazione QR.

Supponendo $m \geq n$, siano \mathbf{Q}, \mathbf{R} le matrici della fattorizzazione QR di \mathbf{A} , $\mathbf{A} = \mathbf{QR}$. Allora si ha:

$$\begin{aligned} \|\mathbf{b} - \mathbf{Ax}\|_2^2 &= \|\mathbf{b} - \mathbf{QRx}\|_2^2 = \|\mathbf{Q}(\mathbf{Q}^T \mathbf{b} - \mathbf{Rx})\|_2^2 \\ &= \left\| \mathbf{Q}^T \mathbf{b} - \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} \right\|_2^2 = \left\| \begin{bmatrix} \mathbf{Q}_1^T \mathbf{b} - \mathbf{R}_1 \mathbf{x} \\ \mathbf{Q}_2^T \mathbf{b} \end{bmatrix} \right\|_2^2 = \|\mathbf{Q}_1^T \mathbf{b} - \mathbf{R}_1 \mathbf{x}\|_2^2 + \|\mathbf{Q}_2^T \mathbf{b}\|_2^2. \end{aligned}$$

Riassumendo, il problema ai minimi quadrati si riduce a minimizzare $\mathbf{Q}_1^T \mathbf{b} - \mathbf{R}_1 \mathbf{x}$ siccome $\mathbf{Q}_2^T \mathbf{b}$ è costante. In particolare, dato che \mathbf{A} ha rango massimo, \mathbf{R}_1 è non singolare, quindi il sistema $\mathbf{Q}_1^T \mathbf{b} - \mathbf{R}_1 \mathbf{x} = \mathbf{0}$ ha un'unica soluzione \mathbf{x} , e tale \mathbf{x} è la soluzione del problema ai minimi quadrati.

Riprendendo l'esempio usato precedentemente, la fattorizzazione QR ridotta della matrice \mathbf{A} è:

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1 = \begin{pmatrix} \frac{3}{5} & 0 \\ \frac{4}{5} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 & 5 \\ 0 & \alpha \end{pmatrix},$$

quindi usando la fattorizzazione QR il problema ai minimi quadrati è ancora risolvibile.

In particolare, se aggiungessimo una colonna alla matrice \mathbf{A} , si potrebbe ottenere la fattorizzazione QR ridotta della nuova matrice \mathbf{A} a partire dalla fattorizzazione QR ridotta della matrice \mathbf{A} iniziale.

Infatti, sia $\mathbf{A} \in \mathbb{R}^{m \times n}$ una matrice a rango massimo con $m > n$, \mathbf{Q}, \mathbf{R} le matrici ottenute dalla fattorizzazione QR ridotta di \mathbf{A} , consideriamo un vettore $\mathbf{y} \in \mathbb{R}^m$ che non è combinazione lineare delle colonne di \mathbf{A} . Sia $\tilde{\mathbf{A}} = [\mathbf{A}, \mathbf{y}]$, la fattorizzazione QR ridotta della matrice $\tilde{\mathbf{A}}$ si ottiene nel seguente modo:

$$\begin{aligned} \tilde{\mathbf{y}} &= \mathbf{Q}^T \mathbf{y}; \\ \tilde{\mathbf{R}}_{1:n, n+1} &= \tilde{\mathbf{y}}; \\ \tilde{\mathbf{q}} &= \mathbf{y} - \mathbf{Q} \tilde{\mathbf{y}}; \\ \tilde{\mathbf{R}}_{n+1, n+1} &= \|\tilde{\mathbf{q}}\|_2; \\ \mathbf{q} &= \tilde{\mathbf{q}} / \|\tilde{\mathbf{q}}\|_2; \\ \tilde{\mathbf{Q}} &= [\mathbf{Q}, \mathbf{q}]. \end{aligned}$$

Questa proprietà sarà molto utile nei capitoli successivi.

Capitolo 3

Metodi matching pursuit per la rappresentazione sparsa

Per scrivere questo capitolo sono stati utilizzati il libro "*Sparse and Redundant Representations, From Theory to Applications in Signal and Image Processing*" [6] e l'articolo "*A Survey of Sparse Representation: Algorithms and Applications*" [7].

3.1 Il problema della rappresentazione sparsa

Siano dati i vettori $\mathbf{d}_1, \dots, \mathbf{d}_n \in \mathbb{R}^m$. La matrice $\mathbf{D} \in \mathbb{R}^{m \times n}$ che ha come colonne i vettori \mathbf{d}_i è un dizionario.

Definizione 3.1. *Un dizionario è una matrice $\mathbf{D} \in \mathbb{R}^{m \times n}$. Le colonne di un dizionario sono dette campioni. Se $m < n$ e ogni vettore $\mathbf{v} \in \mathbb{R}^m$ può essere scritto come combinazione lineare delle colonne di \mathbf{D} , \mathbf{D} è un dizionario sovrappieno.*

Sia $\mathbf{y} \in \mathbb{R}^m$ un campione da analizzare, e supponiamo che $\mathbf{D} \in \mathbb{R}^{m \times n}$ sia un dizionario sovrappieno. Allora si avrà:

$$\mathbf{y} = \mathbf{d}_1\alpha_1 + \dots + \mathbf{d}_n\alpha_n,$$

per qualche α_i . Per semplicità si può scrivere l'equazione precedente nel seguente modo:

$$\mathbf{y} = \mathbf{D}\boldsymbol{\alpha}, \tag{3.1}$$

dove $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_n]$, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^T$. Per le ipotesi che abbiamo posto, tale sistema è sottodeterminato, quindi non avrà una soluzione unica. Se vogliamo avere una soluzione unica bisogna imporre delle restrizioni. Tuttavia, la scelta dei vincoli influenza molto il risultato ottenuto. Una categoria di metodi per risolvere il problema (3.1) è la categoria dei metodi di rappresentazione sparsa con minimizzazione rispetto alla norma l_0 .

3.2 Rappresentazione sparsa con minimizzazione rispetto alla norma l_0

Minimizzando rispetto alla norma l_0 si ottiene la soluzione più sparsa del sistema. In particolare, risolvere il sistema con tale metodo equivale a risolvere il seguente problema di ottimizzazione:

$$\hat{\alpha} = \arg \min \|\alpha\|_0 \quad t.c. \quad \mathbf{y} = \mathbf{D}\alpha. \quad (3.2)$$

Tuttavia, questo metodo presenta due problemi:

1. sotto quali condizioni esiste una soluzione unica?
2. se abbiamo un candidato per la soluzione del sistema $\mathbf{y} = \mathbf{D}\alpha$, come possiamo verificare che tale soluzione sia il minimo globale rispetto alla norma l_0 ?

Inoltre, anche se sapessimo il numero di elementi non zero della soluzione, trovare tale soluzione tramite una ricerca sulle combinazioni delle colonne di \mathbf{D} è un problema NP-difficile. Quindi sorge un'altra domanda: è possibile risolvere il problema (3.2) in modo efficiente? Nelle prossime pagine proveremo a trattare questi quesiti.

3.2.1 Unicità della soluzione mediante spark

Per lo studio dell'unicità, un concetto riguardante le matrici che è cruciale è lo spark di una matrice.

Definizione 3.2. Sia $\mathbf{D} \in \mathbb{R}^{m \times n}$ una matrice. Lo spark di \mathbf{D} è il più piccolo numero di colonne di \mathbf{D} linearmente dipendenti, e si indica con $\text{spark}(\mathbf{D})$.

Osservazione 3.3. Sia $m < n$, $\mathbf{D} \in \mathbb{R}^{m \times n}$ una matrice che non ha colonne nulle. Allora si ha: $2 \leq \text{spark}(\mathbf{D}) \leq n + 1$.

Segue immediatamente dalla definizione di spark che i vettori α diversi da $\mathbf{0}$ che soddisfano $\mathbf{D}\alpha = \mathbf{0}$ devono soddisfare anche $\|\alpha\|_0 \geq \text{spark}(\mathbf{D})$. Infatti, questi vettori combinano linearmente colonne di \mathbf{D} per dare $\mathbf{0}$, e almeno $\text{spark}(\mathbf{D})$ colonne sono necessarie per definizione di spark per avere un'insieme di colonne linearmente dipendenti. Grazie a questa proprietà si ha il seguente teorema.

Teorema 3.4 (Unicità). Se $\mathbf{D}\alpha = \mathbf{y}$ ammette una soluzione α che soddisfa $\|\alpha\|_0 < \text{spark}(\mathbf{D})/2$, allora tale soluzione è la più sparsa possibile.

Dimostrazione. Sia \mathbf{v} una soluzione diversa da $\boldsymbol{\alpha}$. Allora $\boldsymbol{\alpha} - \mathbf{v}$ appartiene al nucleo di \mathbf{D} , ossia $\mathbf{D}(\boldsymbol{\alpha} - \mathbf{v}) = \mathbf{0}$. Dato che $\boldsymbol{\alpha} - \mathbf{v} \neq \mathbf{0}$, per la definizione di spark si ha:

$$\|\boldsymbol{\alpha}\|_0 + \|\mathbf{v}\|_0 \geq \|\boldsymbol{\alpha} - \mathbf{v}\|_0 \geq \text{spark}(\mathbf{D}).$$

Siccome $\|\boldsymbol{\alpha}\|_0 < \text{spark}(\mathbf{D})/2$, $\|\mathbf{v}\|_0 > \text{spark}(\mathbf{D})/2$. Quindi, $\boldsymbol{\alpha}$ è la soluzione più sparsa possibile \square

Per poter usare questo risultato per risolvere (3.2) è necessario conoscere $\text{spark}(\mathbf{D})$. Tuttavia, per conoscere lo spark di una matrice è necessario effettuare una ricerca sulle combinazioni delle colonne di \mathbf{D} che è molto costoso. Si possono stabilire dei limiti superiori e inferiori al valore di $\text{spark}(\mathbf{D})$ che permettono in modo vago di studiare l'unicità della soluzione.

3.2.2 Unicità della soluzione mediante mutua coerenza

Siccome calcolare lo spark di una matrice è molto costoso, metodi più semplici per studiare l'unicità della soluzione sono d'interesse. Uno di questi metodi sfrutta la mutua coerenza di una matrice.

Definizione 3.5. Sia $\mathbf{D} \in \mathbb{R}^{m \times n}$ una matrice. Si definisce la mutua coerenza di \mathbf{D} come il più grande prodotto scalare normalizzato rispetto alla norma l_2 in valore assoluto tra colonne diverse di \mathbf{D} , ossia $\mu(\mathbf{D}) = \max_{1 \leq k, j \leq n, i \neq j} \frac{|\mathbf{d}_k^T \mathbf{d}_j|}{\|\mathbf{d}_k\|_2 \|\mathbf{d}_j\|_2}$.

Osservazione 3.6. La mutua coerenza è un modo per analizzare la dipendenza delle colonne di una matrice \mathbf{D} . Infatti, la mutua coerenza di una matrice con colonne ortogonali è 0 essendo le colonne ortogonali tra di loro, mentre generalmente la mutua coerenza di una matrice è strettamente positiva. Più la mutua coerenza di una matrice è bassa, più tale matrice si comporta come una matrice ortogonale.

La mutua coerenza di una matrice \mathbf{D} è semplice da calcolare e permette di determinare un limite inferiore allo spark che è difficile da calcolare. Infatti vale il seguente lemma.

Lemma 3.7. Per ogni matrice $\mathbf{D} \in \mathbb{R}^{m \times n}$, vale la seguente relazione:

$$\text{spark}(\mathbf{D}) \geq 1 + \frac{1}{\mu(\mathbf{D})}.$$

Dimostrazione. Sia $\tilde{\mathbf{D}}$ la matrice ottenuta normalizzando le colonne della matrice \mathbf{D} rispetto alla norma l_2 . Tale operazione preserva sia lo spark della matrice sia la sua mutua coerenza. Consideriamo quindi la matrice $\mathbf{G} = \tilde{\mathbf{D}}^T \tilde{\mathbf{D}}$. Tale matrice soddisfa le seguenti proprietà per come è stata definita:

1. $\mathbf{G}_{k,k} = 1$ per ogni k , $1 \leq k \leq n$;
2. $|\mathbf{G}_{j,k}| \leq \mu(\mathbf{D})$ per ogni j, k , $1 \leq j, k \leq n$, $j \neq k$.

Poniamo $\mu(\mathbf{D}) = \mu$. Consideriamo una sottomatrice di \mathbf{G} di dimensioni $p \times p$ che viene costruita scegliendo p colonne di $\tilde{\mathbf{D}}$ e calcolando la loro matrice di Gram. Per la definizione di dischi di Gerschgorin, se tale matrice di Gram è a dominanza diagonale stretta, ossia $\sum_{j \neq i} |\mathbf{G}_{i,j}^{(p)}| < |\mathbf{G}_{i,i}^{(p)}| = 1$, allora la sottomatrice considerata è definita positiva, quindi le colonne di $\tilde{\mathbf{D}}$ sono linearmente indipendenti. La condizione $1 > (p-1)\mu$ che equivale a $p < 1 + 1/\mu$ implica che tutti i minori di dimensione $p \times p$ sono definiti positivi siccome $\mu \geq |\mathbf{G}_{i,j}^{(p)}|$ per definizione di μ . Perciò, il minimo numero naturale p tale che $p \geq 1 + 1/\mu$ è il più piccolo numero di colonne che possono essere linearmente dipendenti. Quindi, $\text{spark}(\mathbf{D}) \geq 1 + \frac{1}{\mu}$. \square

Analogamente ai precedenti teoremi abbiamo il seguente risultato.

Teorema 3.8 (Unicità). *Se un sistema $\mathbf{D}\boldsymbol{\alpha} = \mathbf{y}$ ha una soluzione che soddisfa $\|\boldsymbol{\alpha}\|_0 < \frac{1}{2}(1 + 1/\mu(\mathbf{D}))$, questa soluzione è necessariamente la più sparsa possibile.*

3.3 Algoritmi MP e OMP

Nella sezione precedente abbiamo dimostrato che una soluzione sufficientemente sparsa del sistema (3.1) è necessariamente la più sparsa possibile tra tutte le soluzioni. Una conseguenza immediata di ciò è che ogni soluzione sufficientemente sparsa è anche soluzione di (3.2), quindi ora possiamo concentrarci su metodi per trovare tale soluzione.

Esistono diverse categorie di metodi per trovare la soluzione più sparsa del sistema. Tra questi, sono di interesse gli algoritmi matching pursuit che sono algoritmi greedy.

Definizione 3.9. *Un algoritmo greedy è un algoritmo che ad ogni passo effettua la scelta che in quel momento appare migliore e non torna mai indietro.*

I metodi matching pursuit permettono di trovare una soluzione approssimata del problema (3.2). Due algoritmi appartenenti a questa categoria sono l'algoritmo Matching Pursuit (MP) e l'algoritmo Orthogonal Matching Pursuit (OMP).

L'idea alla base del metodo MP è di scegliere iterativamente dal dizionario \mathbf{D} il miglior campione secondo una certa misura di similarità per ottenere la soluzione più sparsa. Di seguito sono descritti i passaggi dell'algoritmo MP.

Sia $\mathbf{y} = \mathbf{r}_0$ il residuo iniziale, $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_n] \in \mathbb{R}^{d \times n}$ un dizionario sovrappieno le cui colonne sono normalizzate rispetto alla norma l_2 . Per approssimare \mathbf{y} , l'algoritmo

MP sceglie la colonna \mathbf{d}_{λ_0} in modo che valga la seguente condizione:

$$|\mathbf{r}_0 \cdot \mathbf{d}_{\lambda_0}| = \max_{i \in \{1, \dots, n\}} |\mathbf{r}_0 \cdot \mathbf{d}_i|,$$

dove λ_0 è uno degli indici delle colonne di \mathbf{D} . Perciò \mathbf{y} può essere scomposta nella seguente equazione:

$$\mathbf{y} = (\mathbf{r}_0 \cdot \mathbf{d}_{\lambda_0}) \mathbf{d}_{\lambda_0} + \mathbf{r}_1.$$

Siccome $(\mathbf{r}_0 \cdot \mathbf{d}_{\lambda_0}) \mathbf{d}_{\lambda_0}$ rappresenta la proiezione ortogonale di \mathbf{y} su \mathbf{d}_{λ_0} , \mathbf{d}_{λ_0} e \mathbf{r}_1 saranno ortogonali, ossia si avrà:

$$\|\mathbf{y}\|^2 = |\mathbf{y} \cdot \mathbf{d}_{\lambda_0}|^2 + \|\mathbf{r}_1\|^2.$$

Alla t -esima iterazione il miglior campione approssimante è \mathbf{d}_{λ_t} e si ha:

$$\mathbf{r}_t = (\mathbf{r}_t \cdot \mathbf{d}_{\lambda_t}) \mathbf{d}_{\lambda_t} + \mathbf{r}_{t+1},$$

con \mathbf{d}_{λ_t} che soddisfa:

$$|\mathbf{r}_t \cdot \mathbf{d}_{\lambda_t}| = \max_{i \notin \{\lambda_0, \dots, \lambda_{t-1}\}} |\mathbf{r}_t \cdot \mathbf{d}_i|.$$

Siccome \mathbf{d}_{λ_t} è ortogonale a \mathbf{r}_{t+1} , si ha:

$$\|\mathbf{r}_t\|^2 = |\mathbf{r}_t \cdot \mathbf{d}_{\lambda_t}|^2 + \|\mathbf{r}_{t+1}\|^2.$$

Mettendo insieme tutti i passaggi si ha:

$$\mathbf{y} = \sum_{i=0}^{N-1} (\mathbf{r}_i \cdot \mathbf{d}_{\lambda_i}) \mathbf{d}_{\lambda_i} + \mathbf{r}_N.$$

Se $\|\mathbf{r}_N\|^2$ è sufficientemente piccolo si ha:

$$\mathbf{y} \approx \sum_{i=0}^{N-1} (\mathbf{r}_i \cdot \mathbf{d}_{\lambda_i}) \mathbf{d}_{\lambda_i},$$

dove $N \ll n$. Perciò, il campione da esaminare può essere rappresentato da un numero piccolo di elementi di un dizionario sovrappieno. I passaggi principali dell'algoritmo MP sono riassunti di seguito.

Algorithm 2. Matching pursuit algorithm

Input: \mathbf{y} campione da esaminare, \mathbf{D} dizionario sovrappieno, $maxit$ numero massimo di iterazioni consentite, τ una costante positiva piccola.

Inizializzazioni: $\mathbf{r}_0 = \mathbf{y}$ residuo iniziale, $\mathcal{I} = \emptyset$ insieme degli indici, $t = 1$.

Corpo dell'algoritmo: finchè $\|\mathbf{r}_t\| > \tau$ e $t \leq maxit$:

Passo 1: trova il miglior campione approssimante \mathbf{d}_{λ_t} mediante $\lambda_t = \arg \max_{j \notin \mathcal{I}} |\mathbf{d}_j^T \mathbf{r}_{t-1}|$.

Passo 2: aggiorna il coefficiente di sparsità $\boldsymbol{\alpha}(\lambda_t) = \mathbf{d}_{\lambda_t}^T \mathbf{r}_{t-1}$ e l'insieme degli indici $\mathcal{I} = \mathcal{I} \cup \lambda_t$.

Passo 3: aggiorna il residuo, $\mathbf{r}_t = \mathbf{y} - \mathbf{D}_{:, \mathcal{I}} \boldsymbol{\alpha}$.

Passo 4: $t = t + 1$.

Output: $\boldsymbol{\alpha}, \mathcal{I}$.

L'algoritmo OMP è una versione migliorata dell'algoritmo MP. Questo miglioramento deriva dal fatto che il metodo OMP risolve ad ogni iterazione un problema dei minimi quadrati per garantire la direzione ortogonale della proiezione rispetto alle colonne di \mathbf{D} già scelte. I passaggi principali dell'algoritmo OMP sono riassunti di seguito.

Algorithm 3. Orthogonal matching pursuit algorithm

Input: \mathbf{y} campione da esaminare, \mathbf{D} dizionario sovrappieno, $maxit$ numero massimo di iterazioni consentite, τ una costante positiva piccola.

Inizializzazioni: $\mathbf{r}_0 = \mathbf{y}$ residuo iniziale, $\mathcal{I} = \emptyset$ insieme degli indici, $t = 1$.

Corpo dell'algoritmo: finchè $\|\mathbf{r}_t\| > \tau$ e $t \leq maxit$:

Passo 1: trova il miglior campione approssimante \mathbf{d}_{λ_t} mediante $\lambda_t = \arg \max_{j \notin \mathcal{I}} |\mathbf{d}_j^T \mathbf{r}_{t-1}|$.

Passo 2: aggiorna l'insieme degli indici $\mathcal{I} = \mathcal{I} \cup \lambda_t$.

Passo 3: calcola il coefficiente di sparsità $\boldsymbol{\alpha} = \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{I}|}} \|\mathbf{y} - \mathbf{D}_{:, \mathcal{I}} \boldsymbol{\alpha}\|_2^2$.

Passo 4: aggiorna il residuo, $\mathbf{r}_t = \mathbf{y} - \mathbf{D}_{:, \mathcal{I}} \boldsymbol{\alpha}$.

Passo 5: $t = t + 1$.

Output: $\boldsymbol{\alpha}, \mathcal{I}$.

Al quarto passaggio di OMP si ha la risoluzione di un problema ai minimi quadrati. Per risolvere tale problema si può usare la fattorizzazione QR ottenuta mediante il metodo di Gram-Schmidt modificato enunciato nel capitolo precedente. Dato che la matrice $\mathbf{D}_{:, \mathcal{I}}$ viene modificata ad ogni iterazione aggiungendo una colonna, si può calcolare la nuova fattorizzazione QR partendo dalla precedente fattorizzazione QR. Inoltre, tale passaggio garantisce anche l'ortogonalità della nuova direzione rispetto a quelle precedenti.

L'algoritmo MP è meno efficiente dell'algoritmo OMP se le colonne del dizionario usato non sono ortogonali e il numero massimo di iterazioni non è basso. Infatti, essendo un algoritmo greedy, il metodo MP sceglie ad ogni passo la colonna che meglio approssima il residuo in quel momento. Siccome non viene risolto un problema dei minimi quadrati nel metodo MP, non è garantito che la nuova direzione della proiezione sia ortogonale a tutte quelle precedenti, quindi è possibile che l'algoritmo MP faccia una scelta sbagliata. Se il metodo effettua una scelta sbagliata in uno dei primi passi, potrebbe spendere il resto del tempo a sistemare l'errore causato da tale scelta all'inizio.

Esiste anche un'implementazione dell'algoritmo OMP che calcola il residuo senza risolvere un problema dei minimi quadrati ad ogni iterazione [8]. Essa funziona nel seguente modo:

Dati in input \mathbf{y} campione da esaminare e \mathbf{D} dizionario sovrappieno, supponiamo dopo un certo numero di iterazioni di avere un'insieme \mathcal{I} di colonne di \mathbf{D} e $\boldsymbol{\alpha}$ tale che

$$\boldsymbol{\alpha} = \arg \min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}_{:, \mathcal{I}} \boldsymbol{\alpha}\|_2^2.$$

Sia $\mathbf{D}_{:, \mathcal{I}} = \mathbf{Q}_1 \mathbf{R}_1$ la fattorizzazione QR ridotta di $\mathbf{D}_{:, \mathcal{I}}$, allora per il residuo vale:

$$\mathbf{r} = \mathbf{y} - \mathbf{D}_{:, \mathcal{I}} \boldsymbol{\alpha} = \mathbf{Q}_1 \mathbf{R}_1 \mathbf{R}_1^{-1} \mathbf{Q}_1^T \boldsymbol{\alpha} = (\mathbf{I} - \mathbf{Q}_1 \mathbf{Q}_1^T) \mathbf{y},$$

con \mathbf{r} ortogonale a \mathbf{Q}_1 . Sia \mathbf{d} una nuova colonna che deve essere aggiunta. Nella prima implementazione di OMP verrebbe risolto il problema ai minimi quadrati

$$\tilde{\boldsymbol{\alpha}} = \arg \min_{\tilde{\boldsymbol{\alpha}}} \|\mathbf{y} - [\mathbf{D}_{:, \mathcal{I}}, \mathbf{d}] \tilde{\boldsymbol{\alpha}}\|_2.$$

In questa implementazione dell'OMP, sia:

$$[\mathbf{D}_{:, \mathcal{I}}, \mathbf{d}] = [\mathbf{Q}_1, \tilde{\mathbf{q}}] \tilde{\mathbf{R}}_1, \quad \tilde{\mathbf{R}}_1 = \begin{bmatrix} \mathbf{R}_1 & s_1 \\ \mathbf{0} & s_2 \end{bmatrix}$$

la fattorizzazione QR ridotta di $[\mathbf{D}_{:, \mathcal{I}}, \mathbf{d}]$ aggiornata dalla fattorizzazione QR precedente. Il nuovo residuo può essere scritto nel seguente modo:

$$\tilde{\mathbf{r}} = \mathbf{y} - [\mathbf{D}_{:, \mathcal{I}}, \mathbf{d}] \tilde{\boldsymbol{\alpha}} = \dots = (\mathbf{I} - [\mathbf{Q}_1, \tilde{\mathbf{q}}][\mathbf{Q}_1, \tilde{\mathbf{q}}]^T) \mathbf{y} = \mathbf{y} - \mathbf{Q}_1 \mathbf{Q}_1^T \mathbf{y} - \tilde{\mathbf{q}} \tilde{\mathbf{q}}^T \mathbf{y} = \mathbf{r} - \tilde{\mathbf{q}} \tilde{\mathbf{q}}^T \mathbf{y}.$$

Siccome $\tilde{\mathbf{r}}$ è ortogonale a $[\mathbf{Q}_1, \tilde{\mathbf{q}}]$ e quindi a $\tilde{\mathbf{q}}$, si ha:

$$\|\mathbf{r}\|_2^2 = \|\tilde{\mathbf{r}} + \tilde{\mathbf{q}} \tilde{\mathbf{q}}^T \mathbf{y}\|_2^2 = \|\tilde{\mathbf{r}}\|_2^2 + \|\tilde{\mathbf{q}} \tilde{\mathbf{q}}^T \mathbf{y}\|_2^2$$

da cui

$$\|\tilde{\mathbf{r}}\|_2^2 = \|\mathbf{r}\|_2^2 - \|\tilde{\mathbf{q}} \tilde{\mathbf{q}}^T \mathbf{y}\|_2^2 = \|\mathbf{r}\|_2^2 - |\tilde{\mathbf{q}}^T \mathbf{y}|^2.$$

Per quanto dimostrato, si può ottenere la norma al quadrato del nuovo residuo senza calcolarlo. Quindi, possiamo controllare se le condizioni per uscire dal ciclo sono soddisfatte.

Assumendo di aver calcolato $\mathbf{p} = \mathbf{D}^T \mathbf{r}$, si ha:

$$\tilde{\mathbf{p}} = \mathbf{D}^T \tilde{\mathbf{r}} = \mathbf{D}^T (\mathbf{r} - \tilde{\mathbf{q}} \tilde{\mathbf{q}}^T \mathbf{y}) = \mathbf{p} - \mathbf{D}^T \tilde{\mathbf{q}} (\tilde{\mathbf{q}}^T \mathbf{y}),$$

con $(\tilde{\mathbf{q}}^T \mathbf{y})$ che è già stato calcolato. Tale procedura può essere ripetuta per le successive iterazioni. Per evitare che colonne scelte vengano scelte di nuovo, si azzerano le componenti di $\tilde{\mathbf{p}}$ i cui indici corrispondono alle colonne già scelte.

Quando il residuo è sufficientemente basso o il numero massimo di iterazioni è stato raggiunto, la soluzione $\boldsymbol{\alpha}$ viene ottenuta risolvendo un solo problema dei minimi quadrati nel seguente modo: sia \mathbf{Q} la matrice ortogonale che si ha alla fine delle iterazioni e $\mathbf{D}_{:, \mathcal{I}}$ la matrice costituita dalle colonne scelte dal metodo OMP. Si prende la matrice $\mathbf{R} = \text{triu}(\mathbf{Q}^T \mathbf{D}_{:, \mathcal{I}})$, dove $\mathbf{R} = \text{triu}(\mathbf{D})$ è la matrice le cui componenti lungo la diagonale e sopra di essa corrispondono a quelle della matrice \mathbf{D} , le componenti sotto la diagonale sono 0, e si risolve il sistema lineare $\tilde{\mathbf{p}} = \mathbf{R} \boldsymbol{\alpha}$.

Queste due implementazioni dell'algoritmo OMP ottengono la stessa rappresentazione sparsa, ma la versione che risolve un solo problema ai minimi quadrati è più efficiente.

Negli algoritmi MP e OMP abbiamo considerato delle generiche matrici \mathbf{D} che non hanno colonne normalizzate. Possiamo normalizzare le colonne della matrice \mathbf{D} moltiplicandola per la matrice diagonale \mathbf{W} tale che $\mathbf{W}_{i,i} = 1/\|\mathbf{d}_i\|_2$, 0 nelle altre posizioni. Consideriamo quindi $\tilde{\mathbf{D}} = \mathbf{D}\mathbf{W}$, il risultato degli algoritmi MP e OMP saranno differenti se usiamo la matrice $\tilde{\mathbf{D}}$ al posto di \mathbf{D} ? La risposta è data dal seguente teorema.

Teorema 3.10. *Sia $\mathbf{D} \in \mathbb{R}^{m \times n}$, sia $\tilde{\mathbf{D}}$ la matrice \mathbf{D} le cui colonne sono normalizzate rispetto alla norma l_2 . Usare \mathbf{D} o $\tilde{\mathbf{D}}$ non influenza la scelta degli indici degli algoritmi MP e OMP.*

Dimostrazione. Dimostriamo il teorema solo per l'algoritmo OMP.

Supponiamo di essere al t -esimo ciclo dell'algoritmo OMP. Trovare l'indice successivo j_0 equivale a trovare l'indice j_0 tale che $\epsilon(j) = \|\mathbf{r}_{t-1}\|_2^2 - (\mathbf{a}_j^T \mathbf{r}_{t-1})^2 / \|\mathbf{a}_j\|_2^2$ sia minimo. Ponendo $\tilde{\mathbf{a}}_j = \mathbf{a}_j / \|\mathbf{a}_j\|_2$, si ha che $\|\tilde{\mathbf{a}}_j\| = 1$, perciò abbiamo:

$$\epsilon(j) = \|\mathbf{r}_{t-1}\|_2^2 - \left(\frac{\mathbf{a}_j^T \mathbf{r}_{t-1}}{\|\mathbf{a}_j\|_2} \right)^2 = \|\mathbf{r}_{t-1}\|_2^2 - (\tilde{\mathbf{a}}_j^T \mathbf{r}_{t-1})^2 = \|\mathbf{r}_{t-1}\|_2^2 - \left(\frac{\tilde{\mathbf{a}}_j^T \mathbf{r}_{t-1}}{\|\tilde{\mathbf{a}}_j\|_2} \right)^2,$$

quindi usare colonne normalizzate porta alla stessa scelta dell'indice j_0 . Dimostriamo ora che il residuo trovato in seguito alla risoluzione del problema ai minimi quadrati è

lo stesso usando \mathbf{D} e $\tilde{\mathbf{D}}$. Sia \mathbf{D}_t la matrice costituita dalle colonne scelte di \mathbf{D} , siccome la soluzione al problema dei minimi quadrati $\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}_t \boldsymbol{\alpha}\|_2^2$ è $\boldsymbol{\alpha} = (\mathbf{D}_t^T \mathbf{D}_t)^{-1} \mathbf{D}_t^T \mathbf{y}$ il residuo al t -esimo passo è:

$$\mathbf{r}_t = \mathbf{y} - \mathbf{D}_t \boldsymbol{\alpha} = [\mathbf{I} - \mathbf{D}_t (\mathbf{D}_t^T \mathbf{D}_t)^{-1} \mathbf{D}_t^T] \mathbf{y}.$$

Considerando la matrice $\tilde{\mathbf{D}}_t = \mathbf{D}_t \mathbf{W}_t$, dove \mathbf{W}_t è la porzione di \mathbf{W} che si riferisce alle colonne degli indici scelti dall'algoritmo OMP, si ha:

$$\begin{aligned} \tilde{\mathbf{r}}_t &= [\mathbf{I} - \tilde{\mathbf{D}}_t (\tilde{\mathbf{D}}_t^T \tilde{\mathbf{D}}_t)^{-1} \tilde{\mathbf{D}}_t^T] \mathbf{y} \\ &= [\mathbf{I} - \mathbf{D}_t \mathbf{W}_t (\mathbf{W}_t^T \mathbf{D}_t^T \mathbf{D}_t \mathbf{W}_t)^{-1} \mathbf{W}_t \mathbf{D}_t^T] \mathbf{y} \\ &= [\mathbf{I} - \mathbf{D}_t \mathbf{W}_t \mathbf{W}_t^{-1} (\mathbf{D}_t^T \mathbf{D}_t)^{-1} \mathbf{W}_t^{-1} \mathbf{W}_t \mathbf{D}_t^T] \mathbf{y} \\ &= [\mathbf{I} - \mathbf{D}_t (\mathbf{D}_t^T \mathbf{D}_t)^{-1} \mathbf{D}_t^T] \mathbf{y} = \mathbf{r}_t. \end{aligned}$$

Abbiamo appena dimostrato che il residuo è invariante rispetto alla normalizzazione della matrice. Siccome è il residuo a influenzare la scelta dell'indice successivo, l'algoritmo non è influenzato dalla normalizzazione di \mathbf{D} . \square

La normalizzazione della matrice \mathbf{D} non ha un effetto sulla scelta degli indici, tuttavia essa ha un effetto sul risultato ottenuto $\boldsymbol{\alpha}$. Per rimediare a ciò, la soluzione ottenuta dagli algoritmi MP e OMP deve essere denormalizzata. Sia quindi $\tilde{\boldsymbol{\alpha}}_k$ la soluzione ottenuta al passo k , $\tilde{\mathbf{D}} = \mathbf{W}\mathbf{D}$ la matrice normalizzata, si ha:

$$\mathbf{y} = \tilde{\mathbf{D}} \tilde{\boldsymbol{\alpha}}_k = \mathbf{D} \mathbf{W} \tilde{\boldsymbol{\alpha}}_k = \mathbf{D} \boldsymbol{\alpha}_k$$

ponendo $\boldsymbol{\alpha}_k = \mathbf{W} \tilde{\boldsymbol{\alpha}}_k$.

3.4 Condizione di recupero esatto

Nelle sezioni precedenti abbiamo trattato il problema (3.2) ed in particolare l'unicità della sua soluzione ed alcuni algoritmi per determinare tale soluzione. Per concludere tale discussione parleremo della condizione di recupero esatto o Exact Recovery Condition (ERC) che ci permetterà di analizzare ulteriormente le performance degli algoritmi MP e OMP. Per parlare dell'ERC, dobbiamo dare la definizione di supporto di una soluzione.

Definizione 3.11. *Sia $\mathbf{D}\boldsymbol{\alpha} = \mathbf{y}$ un sistema sottodeterminato, sia $\hat{\boldsymbol{\alpha}}$ una sua soluzione. Si definisce supporto della soluzione $\hat{\boldsymbol{\alpha}}$ l'insieme S degli indici degli elementi non zero di $\hat{\boldsymbol{\alpha}}$.*

Osservazione 3.12. L'insieme S appena definito può anche essere visto come l'insieme degli indici delle colonne di \mathbf{D} che compongono la soluzione a cui il supporto si riferisce.

Denotando \mathbf{D}_S la sottomatrice di \mathbf{D} composta dalle colonne individuate dal supporto, possiamo definire l'ERC.

Definizione 3.13. *Dato un supporto S e una matrice \mathbf{D} , l'ERC è soddisfatta se vale la seguente disuguaglianza:*

$$ERC(\mathbf{D}, S) : \max_{i \notin S} \|\mathbf{D}_S^+ \mathbf{d}_i\|_1 < 1. \quad (3.3)$$

Considerando per ogni $i \notin S$ il sistema $\mathbf{D}_S \mathbf{x} = \mathbf{d}_i$, l'ERC richiede che le soluzioni di tutti questi sistemi abbiano norma l_1 minore di 1.

L'importanza dell'ERC deriva dal suo collegamento con il successo dell'algoritmo OMP.

Teorema 3.14. *Sia $\boldsymbol{\alpha}$ con supporto S una soluzione sparsa del sistema lineare $\mathbf{D}\boldsymbol{\alpha} = \mathbf{y}$. Se l'ERC è soddisfatta l'algoritmo OMP restituirà la soluzione $\boldsymbol{\alpha}$.*

Dimostrazione. Supponiamo di essere al primo passo del ciclo. Affinchè il ciclo finisca in modo soddisfacente bisogna richiedere che il valore $\|\mathbf{D}_S^T \mathbf{y}\|_\infty$ sia maggiore del massimo prodotto interno di \mathbf{y} con colonne i cui indici non appartengono al supporto. Tale prodotto è $\|\mathbf{D}_{S^c}^T \mathbf{y}\|_\infty$, perciò richiediamo

$$\rho = \frac{\|\mathbf{D}_{S^c}^T \mathbf{y}\|_\infty}{\|\mathbf{D}_S^T \mathbf{y}\|_\infty} < 1.$$

Siccome $\mathbf{y} = \mathbf{D}_S \boldsymbol{\alpha}_S$, si ha:

$$\begin{aligned} (\mathbf{D}_S^T)^+ \mathbf{D}_S^T \mathbf{y} &= \mathbf{D}_S (\mathbf{D}_S^T \mathbf{D}_S)^{-1} \mathbf{D}_S^T \mathbf{y} \\ &= \mathbf{D}_S (\mathbf{D}_S^T \mathbf{D}_S)^{-1} \mathbf{D}_S^T \mathbf{D}_S \boldsymbol{\alpha}_S \\ &= \mathbf{D}_S \boldsymbol{\alpha}_S \\ &= \mathbf{y}. \end{aligned}$$

Perciò,

$$\rho = \frac{\|\mathbf{D}_{S^c}^T \mathbf{y}\|_\infty}{\|\mathbf{D}_S^T \mathbf{y}\|_\infty} = \frac{\|\mathbf{D}_{S^c}^T (\mathbf{D}_S^T)^+ \mathbf{D}_S^T \mathbf{y}\|_\infty}{\|\mathbf{D}_S^T \mathbf{y}\|_\infty} \leq \|\mathbf{D}_{S^c}^T (\mathbf{D}_S^T)^+\|_\infty.$$

La norma l_∞ indotta è definita nel seguente modo:

$$\|\mathbf{D}\|_\infty = \max_i \sum_j |D_{i,j}| = \max_j \sum_i |D_{i,j}^T| = \|\mathbf{D}^T\|_1,$$

con quest'ultima che è la norma l_1 indotta. L'operazione di pseudo-inversione e trasposizione commutano tra di loro. Quindi si ha:

$$\|\mathbf{D}_{S^c}^T (\mathbf{D}_S^T)^+ \|_\infty = \|\mathbf{D}_S^+ \mathbf{D}_{S^c}\|_1.$$

Richiedendo che $\|\mathbf{D}_S^+ \mathbf{D}_{S^c}\|_1 < 1$ otteniamo che $\rho < 1$. Tale richiesta equivale a soddisfare la ERC.

Per quanto dimostrato sopra, si ha che la prima colonna scelta all'inizio dell'algoritmo OMP appartiene al supporto della soluzione, quindi il residuo appartiene al sottospazio vettoriale generato dalle colonne di \mathbf{D}_S . Per i cicli successivi si applicano i ragionamenti enunciati sopra sostituendo \mathbf{r}_k a \mathbf{y} . Dopo $|S|$ passi si avrà la soluzione $\boldsymbol{\alpha}$ \square

L'ERC permette di stabilire delle condizioni per il successo dell'algoritmo OMP, tuttavia tale teorema non è costruttivo poichè si può usare solo se si conosce il supporto della soluzione. Inoltre, se conosciamo solo la cardinalità del supporto bisognerà testare la condizione enunciata nel teorema per tutti i $\binom{d}{|S|}$ possibili supporti che ha un costo proibitivo. Per risolvere questo problema si può usare la mutua coerenza.

Teorema 3.15. *Per una matrice \mathbf{D} con mutua coerenza μ , se $k < \frac{1}{2} \left(1 + \frac{1}{\mu}\right)$, allora per tutti i supporti con cardinalità uguale o minore a k la ERC è soddisfatta.*

Dimostrazione. Per un'arbitraria colonna i -esima non del supporto richiediamo che $\|\mathbf{D}_S^+ \mathbf{d}_i\|_1 = \|(\mathbf{D}_S^T \mathbf{D}_S)^{-1} \mathbf{D}_S^T \mathbf{d}_i\|_1 < 1$. Possiamo sostituire questa condizione con una più forte richiedendo che $\|(\mathbf{D}_S^T \mathbf{D}_S)^{-1}\|_1 \cdot \|\mathbf{D}_S^T \mathbf{d}_i\|_1 < 1$. Il vettore $\mathbf{D}_S^T \mathbf{d}_i$ ha componenti nell'intervallo $[-\mu, \mu]$, perciò $\|\mathbf{D}_S^T \mathbf{d}_i\|_1 \leq |S|\mu$.

La matrice $\mathbf{D}_S^T \mathbf{D}_S$ è a stretta dominanza diagonale, quindi si può dimostrare che

$$\|(\mathbf{D}_S^T \mathbf{D}_S)^{-1}\|_1 \leq \frac{1}{1 - (|S| - 1)\mu}.$$

La dimostrazione si trova in [9]. Combinando tutto si ha:

$$\|(\mathbf{D}_S^T \mathbf{D}_S)^{-1}\|_1 \cdot \|\mathbf{D}_S^T \mathbf{d}_i\|_1 \leq \frac{|S|\mu}{1 - (|S| - 1)\mu} < 1,$$

ossia $|S| < \frac{1}{2} \left(1 + \frac{1}{\mu}\right)$. Siccome questa condizione vale per tutte le colonne \mathbf{d}_i non appartenenti al supporto, concludiamo che se $|S| < \frac{1}{2} \left(1 + \frac{1}{\mu}\right)$, allora la ERC è soddisfatta. \square

Capitolo 4

Applicazioni dei metodi matching pursuit

Per scrivere questo capitolo è stato utilizzato l'articolo "*A Survey of Sparse Representation: Algorithms and Applications*" [7].

Nel capitolo precedente abbiamo parlato del problema della rappresentazione sparsa, in particolare con minimizzazione rispetto alla norma l_0 . Le tecniche per risolvere tale problema hanno numerose applicazioni nel mondo reale per l'analisi di immagini. Infatti, un'immagine può essere vista come un vettore che ha come componenti i pixel. In questa tesi discuteremo una di tali applicazioni: la classificazione di immagini.

4.1 Il problema della classificazione di immagini

Supponiamo di avere c classi di immagini. Sia $D = [\mathbf{d}_1, \dots, \mathbf{d}_n]$ la matrice avente come colonne n immagini di confronto o training samples appartenenti alle c classi, siano D_i le matrici contenenti i campioni di confronto della classe i , e sia \mathbf{y} l'immagine da classificare o test sample appartenente ad una delle c classi. Il problema della classificazione di immagini consiste nel determinare la classe a cui l'immagine \mathbf{y} appartiene. Un metodo generale che usa la minimizzazione rispetto alla norma l_0 per risolvere tale problema è il seguente:

1. si normalizzano tutti i campioni rispetto alla norma l_2 ;
2. si sfruttano tutti i campioni di confronto per rappresentare l'immagine da analizzare risolvendo il problema di ottimizzazione

$$\boldsymbol{\alpha} = \arg \min \|\boldsymbol{\alpha}\|_0 \quad t.c. \quad \mathbf{y} = D\boldsymbol{\alpha}.$$

3. si calcola il residuo r_i per ogni classe ponendo $r_i = \|\mathbf{y} - \mathbf{D}_i\boldsymbol{\alpha}\|_2^2$, dove $\boldsymbol{\alpha}_i$ è il coefficiente di sparsità associato alla i -esima classe;
4. si restituisce la classe che ha il residuo r_i più piccolo rispetto alla norma l_2 .

Siccome nel punto 2 vogliamo la soluzione minima rispetto alla norma l_0 , possiamo usare algoritmi come MP e OMP per trovare $\boldsymbol{\alpha}$.

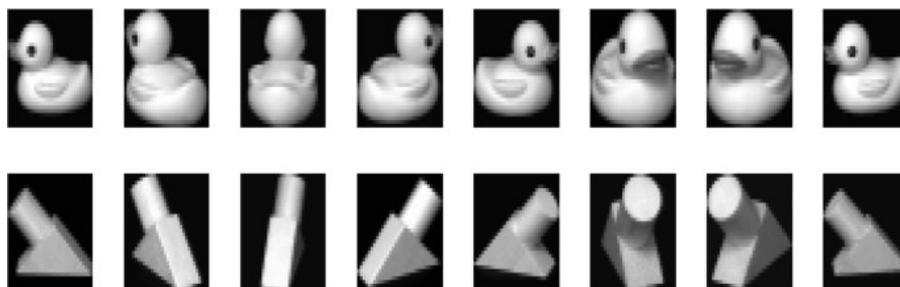
4.2 Illustrazioni computazionali

Per testare le performance degli algoritmi MP e OMP sono stati scelti tre data set di immagini. Le immagini possono essere viste come matrici $n_1 \times n_2$ o come vettori in \mathbb{R}^{n_i} , dove $n_i = n_1 n_2$ è il numero di pixels dell'immagine. I data set usati sono elencati di seguito.

1. il data set ORL [10] contiene foto di 40 persone diverse a scala di grigi. Ogni persona costituisce una classe. Per ogni classe ci sono 10 immagini di dimensioni 112×92 pixels. Le immagini sono state riprese con diverse illuminazioni ed espressioni facciali;



2. il data set coil20 [11] contiene immagini di 20 oggetti a scala di grigi. Ogni oggetto costituisce una classe. Per ogni classe ci sono 72 immagini di dimensioni 128×128 pixels. Tra un'immagine e l'altra, l'oggetto viene ruotato di 5 gradi;



3. il data set yalefaces [12] contiene le foto di 15 persone a scala di grigi. Ogni persona costituisce una classe. Per ogni persona sono state fatte 11 foto di dimensioni 243×320 pixels. Tra le foto ci sono differenze di illuminazione ed espressione facciale.



Le descrizioni dei tre data set sono riassunte nella tabella 4.1.

	n_1	n_2	n_i	num. classi	num. immagini
ORL	112	92	10304	40	10
coil20	128	128	16384	20	72
yalefaces	243	320	77760	15	11

Tabella 4.1: Descrizioni dei tre data set.

Per ogni data set, gli algoritmi MP e OMP verranno testati scegliendo a caso $\#Tr$ immagini di confronto da ogni classe del data set per creare il training set. Le rimanenti immagini delle classi costituiranno il test set.

Da ora in poi, con OMP ci riferiremo all'implementazione del metodo OMP che non risolve un problema ai minimi quadrati ad ogni iterazione, con OMP_LS invece ci riferiremo all'implementazione con la risoluzione di un problema ai minimi quadrati ad ogni iterazione.

Per gli algoritmi MP, OMP e OMP_LS useremo due versioni che variano per la scelta del numero massimo di iterazioni: la prima versione pone il numero massimo di iterazioni a 5, la seconda versione pone il numero massimo di iterazioni a 10. Per ogni versione degli algoritmi usata e numero di immagini di confronto usate per classe sono state effettuate 5 prove con training set e test set diversi.

Il confronto tra le varie versioni degli algoritmi si baserà sulla percentuale media di successo di classificazione delle immagini e sul tempo medio usato per classificarle avendo già definito i training set e test set.

I risultati della classificazione ed il tempo impiegato sono riassunti nelle tabelle 4.2 e 4.3.

Da queste due tabelle possiamo notare svariate cose:

Data set(#Tr)	MP_1	MP_2	OMP_LS_1	OMP_LS_2	OMP_1	OMP_2
ORL(1)	64.61	64.61	67.89	67.22	67.89	67.22
ORL(2)	75.63	72.69	78.44	79.44	78.44	79.44
ORL(3)	83.71	80.00	87.14	87.00	87.14	87.00
ORL(4)	89.50	85.58	91.67	91.50	91.67	91.50
ORL(5)	92.20	87.80	94.40	94.90	94.40	94.90
yalefaces(1)	51.07	51.07	55.60	55.60	55.60	55.60
yalefaces(2)	76.74	73.48	81.19	83.26	81.19	83.26
yalefaces(3)	82.17	76.00	88.50	89.33	88.50	89.33
yalefaces(4)	77.71	70.48	84.58	83.43	84.58	83.43
yalefaces(5)	84.44	76.22	91.56	94.44	91.56	94.44
coil20(3)	75.55	74.29	75.93	75.80	75.93	75.80
coil20(5)	81.67	79.42	83.03	82.57	83.03	82.57
coil20(7)	89.15	84.45	90.14	90.02	90.14	90.02
coil20(9)	90.98	85.30	92.33	92.95	92.33	92.95
coil20(11)	90.28	82.21	91.52	91.33	91.52	91.33

Tabella 4.2: Percentuali medie di successo della classificazione.

Data set(#Tr)	MP_1	MP_2	OMP_LS_1	OMP_LS_2	OMP_1	OMP_2
ORL(1)	0.92	1.05	1.22	2.05	1.09	1.55
ORL(2)	0.97	1.25	1.38	2.34	1.22	1.91
ORL(3)	1.24	1.82	1.59	2.90	1.59	2.36
ORL(4)	1.64	2.70	1.92	3.27	1.90	3.15
ORL(5)	1.67	2.72	1.96	3.39	2.00	3.13
yalefaces(1)	1.21	1.69	2.66	6.20	1.95	3.59
yalefaces(2)	2.21	3.04	3.34	6.39	2.90	4.45
yalefaces(3)	2.81	3.97	3.79	6.84	3.70	5.17
yalefaces(4)	3.07	4.28	3.83	6.49	3.68	5.26
yalefaces(5)	3.26	4.63	3.97	6.74	3.86	5.63
coil20(3)	7.04	8.40	8.75	13.47	8.46	11.15
coil20(5)	10.05	13.77	10.96	16.12	11.31	14.77
coil20(7)	11.67	15.20	12.17	16.61	13.30	16.35
coil20(9)	21.33	25.65	22.07	25.96	23.35	27.00
coil20(11)	24.43	29.60	24.70	30.05	26.72	30.88

Tabella 4.3: Tempi medi in secondi.

1. per un numero basso di iterazioni, gli algoritmi MP e OMP hanno simili percentuali di successo di classificazione. Se alziamo il numero di iterazioni, la percentuale di successo del metodo MP decresce mentre quella del metodo OMP generalmente cresce. Questo succede perchè il metodo OMP risolve un problema ai minimi quadrati ad ogni iterazione per garantire che ad ogni passo si proceda in direzione ortogonale;
2. la versione dell'algoritmo OMP che risolve un solo problema ai minimi quadrati dopo aver effettuato tutti i cicli è generalmente più efficiente della versione che risolve un problema ai minimi quadrati ad ogni iterazione;
3. aumentare il numero di immagini del training set generalmente migliora la percentuale di successo dei metodi OMP e MP, ma aumenta anche il tempo impiegato;
4. anche se ci sono variazioni di illuminazione e di posizione delle persone e oggetti nelle immagini analizzate, i metodi MP e OMP hanno ottime percentuali di successo.

4.3 Conclusioni

In questa tesi abbiamo visto risultati teorici e applicazione pratiche dei metodi matching pursuit.

Per quanto illustrato nei capitoli precedenti, i metodi matching pursuit hanno buone performance per quanto riguarda la classificazione di immagini. Questo continua a valere anche se tra immagine e immagine ci sono variazioni non indifferenti. Infatti, tra le immagini del data base coil20 ci sono differenze per quanto riguarda l'orientazione degli oggetti rappresentati, e tra le facce del database ORL e yalefaces ci sono differenze di espressioni ed illuminazione. Ciò significa che i metodi MP e OMP sono robusti rispetto a queste variazioni.

Tuttavia, questi non sono gli unici problemi che si possono presentare nella classificazione di immagini. Infatti, certe immagini potrebbero risultare corrotte o sfocate. Per complicare la classificazione nel caso le immagini siano foto di persone, le persone fotografate potrebbero anche indossare accessori che coprono il volto come occhiali da sole o sciarpe.

Per ottenere performance migliori, nel corso degli anni sono stati ideati molti altri metodi matching pursuit come il Stage-wise Orthogonal Matching Pursuit (StOMP) o Compressive Sampling Matching Pursuit (CoSaMP).

Bibliografia

- [1] S. G. Mallat and Zhifeng Zhang, *Matching pursuits with time-frequency dictionaries*, in IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3397-3415, Dec. 1993, DOI: 10.1109/78.258082.
- [2] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, *Atomic Decomposition by Basis Pursuit*, in SIAM Journal on Scientific Computing, vol. 20, issue 1, 1998, DOI: 10.1137/S1064827596304010
- [3] David L. Donoho and Michael Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization*, PNAS, 100 (5) 2197-2202, Feb. 2003, DOI: 10.1073/pnas.0437847100
- [4] Davide Palitta e Valeria Simoncini: *Dispense del corso di Calcolo Numerico*, pp. 51-60, II edizione, v.6, 20 settembre 2021.
- [5] Gene H. Golub and Charles F. Van Loan: *Matrix computations*, Johns Hopkins University Press, pp. 223-236, III edizione, 1996.
- [6] Michael Elad: *Sparse and Redundant Representations, From Theory to Applications in Signal and Image Processing*, Springer New York, pp. 17-75, 2010, DOI: 10.1007/978-1-4419-7011-4.
- [7] Z. Zhang, Y. Xu, J. Yang, X. Li and D. Zhang, *A Survey of Sparse Representation: Algorithms and Applications*, in IEEE Access, vol. 3, pp. 490-530, 2015, DOI: 10.1109/ACCESS.2015.2430359
- [8] Valeria Simoncini: *Derivation of optimized OMP*, https://virtuale.unibo.it/pluginfile.php/1852622/mod_resource/content/0/notes_omp.pdf.
- [9] J.M. Varah, *A lower bound for the smallest singular value of a matrix*, Linear Algebra Appl. 11:3-5, 1975.

- [10] AT&T Laboratories Cambridge, *The ORL Database of Faces*, <https://cam-orl.co.uk/facedatabase.html>
- [11] S. A. Nene, S. K. Nayar e H. Murase, *Columbia Object Image Library (COIL-20)*, <https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>
- [12] *Yale Face Database*, <http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html>