# Compressed sensing

## Premessa

Con il termine ‘Compressed sensing’ o ‘Compressive sampling’ o ‘Sparse sampling’ si individua un nuovo paradigma di acquisizione/ricostruzione di segnali (o immagini) che supera i limiti imposti dallo schema classico del campionamento di Nyquist. Infatti con questo metodo è possibile ricostruire un segnale partendo da un numero di campioni inferiore rispetto a quelli imposti dal teorema di Nyquist.



Il meccanismo su cui si basa il paradigma richiede che il segnale sia sparso in qualche dominio opportunamente scelto, quale il dominio di fourier o il dominio wavelet. Una volta che il segnale è stato acquisito con un numero di campioni anche molto inferiore a quelli imposti da Nyquist, è possibile ricostruire il segnale senza perdita di informazioni.

Quindi il Compressed sensing si basa sul sottocampionamento di un segnale o immagine e sulla successiva ricostruzione utilizzando algoritmi robusti basati su un numero di campioni inferiori a quelli imposti dal teorema di Nyquist. Vedremo che il campionamento migliore per ottenere la migliore ricostruzione del segnale o dell’immagine è quello random.

## Matrice di acquisizione

La matrice di acquisizione deve avere le seguenti proprietà:

**Sparsità**: se il segnale (o l’immagine) è sparso in qualche dominio, la quantità di informazione necessaria per rappresentare il segnale in quel dominio è molto minore di quella prodotta dal campionamento di Nyquist, basato sulla larghezza di banda. Pertanto si cercherà una trasformazione che proietti il dominio denso nel dominio sparso, operando un sottocampionamento sulla trasformazione in modo da ridurre il numero di campioni e quindi velocizzare l’acquisizione.

**Incoerenza:** tra il dominio di acquisizione e il dominio sparso si deve avere una bassa correlazione. Tale requisito si impone in quanto non sempre il dominio di acquisizione coincide con quello sparso. In questa situazione può essere necessario definire una doppia trasformazione per passare dal dominio del segnale (denso) a quello sparso.

La matrice di acquisizione A (che può comprendere le due trasformazioni suddette) deve avere le seguenti proprietà:

* le colonne della matrice A devono avere norma unitaria. La matrice A più naturale è la matrice di Fourier, che è una matrice ortonormale. Tuttavia, per effetto del sottocampionamento la matrice A non è quadrata per cui il numero di righe *m* è minore di *N*, con colonne a norma unitaria. Essendo la nuova matrice A rettangolare, essa non può essere ortonormale. Quindi bisogna definire una metrica da utilizzare sulla matrice rettangolare per poter definire quando la matrice è adatta alla sparsificazione. In particolare dovremo verificare che il sottocampionamento è effettuato in modo da ottenere **l’incoerenza tra gli elementi di A.**

* data la matrice di acquisizione A si può definire la matrice (quadrata) di correlazione B = ATA, che è una matrice di Toeplitz. Tale matrice può essere utilizzata per valutare il grado di incorrelazione della matrice di acquisizione A e quindi la bontà del sottocampionamento (dato che la migliore ricostruzione necessita di un campionamento random).

## Ricostruzione

Una volta acquisito il segnale (o l’immagine) con i criteri sopra esposti si procede alla ricostruzione. L’operazione di ricostruzione diventa un problema ai minimi quadrati con vincolo.

Sia dato il seguente modello di acquisizione:



con n rumore additivo con , dove A è la matrice di acquisizione.

Come già discusso in precedenza, operando nel dominio sparso (wavelet) si deve minimizzare la seguente funzione: , dove: , . Sappiamo che la stima di x si ottiene mediante trasformazione inversa:

Il precedente problema ai minimi quadrati può essere risolto impiegando la versione iterativa dell’algoritmo di soft thresholding (Iterative Shrinkage Threshold Algorithm - ISTA). In questo caso la funzione di ‘soft thresholding’ può essere inserita nel classico algoritmo iterativo basato sul metodo del gradiente, ora non più nel dominio del segnale, ma nel dominio wavelet.

L’iterazione al passo k-esimo è data da:

 algoritmo iterativo classico,

 che include i seguenti step: passaggio al dominio wavelet, applicazione del soft thresholding e ritorno al dominio iniziale,

dove l’operazione di soft thresholding è data dalla seguente relazione:



I vantaggi del metodo ISTA sono la semplicità, specialmente quando il vettore da filtrare è separabile, cioè ciascuna componente del vettore può essere trattata separatamente dalle altre. In questo caso il problema del filtraggio si riduce ad un problema di minimizzazione monodimensionale.

Tuttavia l’algoritmo ISTA è lento. Soluzioni più veloci sono ottenute con l’algoritmo FISTA (Fast Iterative Shrinkage Threshold Algorithm) non trattato in questa dispensa, ma fornito dal docente sotto forma di programma di matlab.

##### Esercitazione n. 3.8: Sottocampionamento della matrice di Fourier

1° parte

* Generare la matrice di Fourier A;
* Generare la matrice ATA e verificare l’incorrelazione



* Effettuare l’undersampling uniforme lungo le colonne (eliminazione delle righe) della matrice di Fourier. Nella figura che segue in rosso è graficata la maschera di sottocampionamento.



* Calcolare la risposta impulsiva della matrice di sottocampionamento uniforme, ovvero la trasformata inversa di Fourier della maschera di sottocampionamento



* Rappresentare la matrice ATA nel caso di sottocampionamento regolare



* Effettuare l’undersampling random lungo le colonne della matrice di Fourier.



* Calcolare la risposta impulsiva della maschera non uniforme



* Rappresentare la ATA della matrice di Fourier sottocampionata in modo random (sinistra) e dopo aver eliminato la diagonale principale (destra)

 

2° parte

* Generare un vettore di zeri (N x 1) con N=1024;
* Scegliere 5 punti (*Nspikes*) in maniera casuale:   
  *idx=randperm(N);  
  x0(idx(1:Nspikes))=1*In questo modo è stato creato un vettore con sparsità uguale a 5, ovvero con 5 elementi non nulli nel dominio temporale.
* Creare la matrice di Fourier NxN e verificare che y0=F\*x0 è la trasformata di Fourier di x0. (ovvero plot(y0), hold on plot(fft(x0)/sqrt(N),’r’))
* Verificare che x1=FT\*y0 coincide con x0 (mediante la trasformata inversa di Fourier)  
  plot(x1),hold on plot(ifft(y0)\*sqrt(N),’r’)
* Confrontare i tempi di calcolo di y0=F\*x0 e di y0=fft(x0)/sqrt(N) utilizzando i comandi tic e toc
* Simulare un sottocampionamento di x0 nel dominio della frequenza (sottocampionamento della matrice di Fourier) con diversi gradi di coerenza della matrice di acquisizione. Per fare questo creare una matrice A(mxN con m=256) selezionando delle righe di F nel seguente modo:  
   *idx1=floor(linspace(1,N,m));%max coherence  
  A1=F(idx1,:);*

*A1 = A1/norm(A1(:,1));  
  
idx2=floor(linspace(1,N,floor(m/2)));  
tmp=1:N;  
tmp(idx2)=[];  
p=randperm(numel(tmp));  
idx2=[idx2,tmp(p(1:floor(m/2)))];   
idx2=sort(idx2);  
A2=F(idx2,:);  
A2 = A2/norm(A2(:,1));*

*idx3=randperm(N);  
idx3=idx3(1:m);  
idx3=sort(idx3);  
A3=F(idx3,:);%coerenza bassa*

*A3 = A3/norm(A3(:,1));*

* Per ogni matrice A creare la ATA e verificare che la matrice è di Toeplitz (usare imagesc(.))
* Per ogni ATA calcolare la coerenza valutando il massimo del modulo fuori dalla diagonale principale. Calcolare anche le risposte impulsive delle maschere di sottocampionamento e verificare che ad una bassa coerenza corrisponde una risposta impulsiva con lobi secondari ridotti.
* Applicare ciascuna delle ATA al vettore x0 (x\_hat\_i= (ATA)i\*x0), sapendo che ATA trasforma e antitrasforma il vettore x0, ed osservare l’effetto della coerenza sul vettore ricostruito x\_hat. Interpretare i risultati confrontandoli con l’andamento di x0.
* Ripetere il tutto per m=32, 64, 128. Osservare l’effetto del grado di sottocampionamento sulla coerenza e sugli artefatti da aliasing.
* Per il solo caso a coerenza bassa, creare un vettore y=A\*x0 e aggiungere un rumore di deviazione standard pari a 1e-2. Ricostruire il vettore con regolarizzazione di Tikhonov e con regolarizzazione in norma 1 (utilizzando l’algoritmo fornito FISTA(.)):

*x\_tikhonov=(A'\*A+lambda1\*eye(N))^-1\*A'\*yn*

*mask=zeros(N,1);*

*mask(idx)=1;*

*x\_l1=FISTA(mask,yn,lambda2,200);*  
Scegliere lambda1 nel range di 10^3 (alta regolarizzazione, il problema è molto mal condizionato). Scegliere lambda2 nel range di 10^-4 /10^-2

