UNIVERSITÀ DEGLI STUDI DI CAGLIARI

FACOLTÀ DI INGEGNERIA

Corso di Laurea in Ingegneria Elettronica

ALGORITMI PER LA BLIND AUDIO SOURCE SEPARATION

Tesi di Laurea Triennale

Relatore:
Prof. Giuseppe Rodriguez

Candidato:
Antonio Orrù

Anno Accademico 2013 - 2014

Grazie, siete tantissimi.

Indice

Introduzione

I segnali audio sono spesso un mix di diverse sorgenti sonore, quali voci, musica e rumore. La *Blind Audio Source Separation (BASS)* consiste nel recuperare una o più sorgenti sonore da una data *mixture,* ossia la “miscela” contenente l'insieme di tali sorgenti, senza avere informazioni dettagliate sul processo di mixing o sui vari componenti.

Lo studio dei metodi per la separazione è nato negli anni ’90, coinvolgendo diverse aree disciplinari, e tuttora continua ad essere oggetto di ricerca. Queste tecniche trovano applicazione in svariati campi, come le telecomunicazioni, l'analisi di segnali di natura medica e, soprattutto, nell'ambito dell'Audio Digital Signal Processing. Tra le applicazioni dirette in ambito audio, abbiamo la separazione in tempo reale del parlato per le traduzioni simultanee, la cancellazione della voce per il karaoke, il denoising, il remixing e il remastering di vecchi supporti musicali e il campionamento di suoni per la composizione di musica elettronica.

Questa tesi ha lo scopo di illustrare il problema della separazione delle sorgenti audio nel caso di voci e musica, analizzando due diversi algoritmi per la risoluzione del problema e mostrandone l'applicazione su dati sperimentali reali. Nel **Capitolo 1** descriverò le varie tipologie di mixtures sia da un punto di vista concettuale che da un punto di vista matematico. Nei **Capitoli 2** e **3** illustrerò gli strumenti e le ipotesi che serviranno per affrontare il problema della separazione, di cui fornirò la soluzione in due step, rispettivamente nei **Capitoli 4** e **5**. Infine, i **Capitoli 6** e **7** mostreranno le metriche per la valutazione della qualità della separazione e i risultati sperimentali ottenuti tramite l'implementazione dei due diversi algoritmi.

Capitolo 1: Classificazione delle mixtures audio

1.1 Cenni sulla natura dei segnali audio d'interesse

Al giorno d'oggi, la registrazione dei segnali audio è quasi totalmente affidata a dispositivi digitali. Il processo di registrazione consiste nella trasduzione di una pressione sonora in un segnale elettrico, nella pre-amplificazione del segnale analogico e, infine, nella conversione A/D, secondo lo schema riportato in Figura 1.

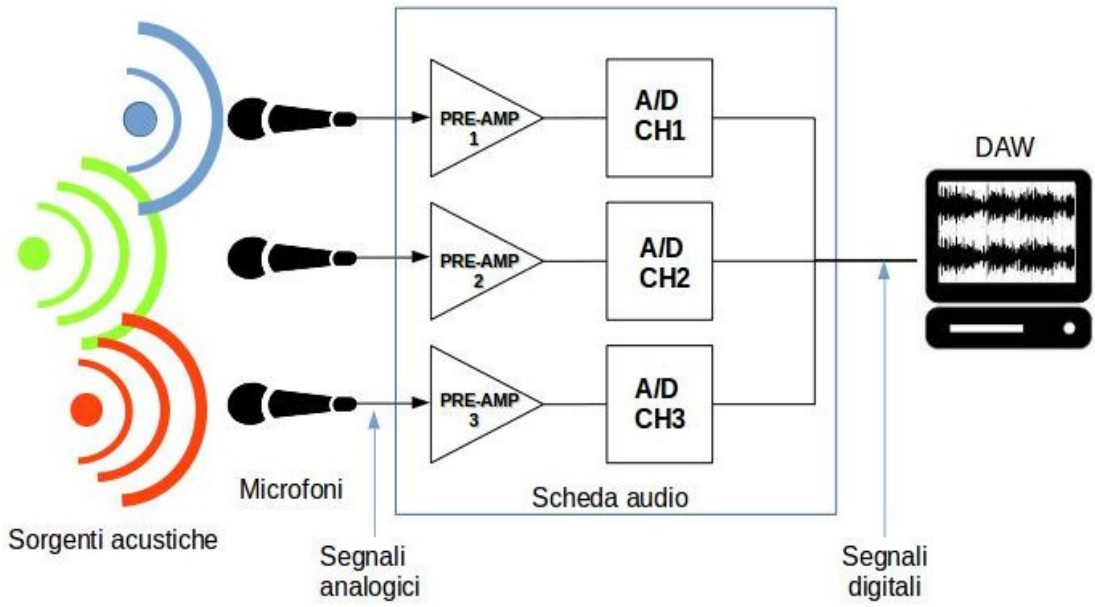


Figura 1. Schema di registrazione dei segnali audio.

La pressione prodotta dalle sorgenti sonore presenti nell'ambiente viene trasdotta dal microfono e il segnale elettrico prodotto varierà in funzione della posizione nello spazio delle sorgenti e dei microfoni, delle caratteristiche acustiche dello spazio circostante e, infine, della risposta del microfono stesso. Per questo motivo, se più sorgenti acustiche presenti nello stesso spazio vengano registrate da un solo microfono, l'interazione tra queste è di natura fisica e avviene a monte della trasduzione.

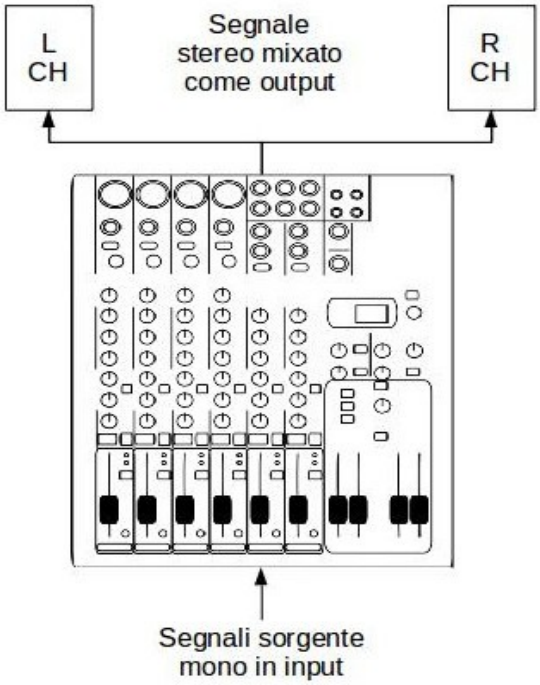


Figura 2. Schema mixaggio sintetico nella produzione di musica pop.

Generalmente, nella registrazione di musica pop, a ciascuno strumento corrispondono un microfono, un pre-amplificatore, un convertitore A/D e una traccia digitale nel software per la registrazione. In seguito, effettuando il mixing sintetico, tutte le singole sorgenti confluiscono all'interno di una traccia stereo [1]. Per via della diversa natura di una miscela ambientale di segnali acustici e di una miscela sintetica di segnali elettrici, a rigore, è scorretto parlare di microfoni e di canali di mixing in maniera analoga. Tuttavia, in letteratura sono presenti dei modelli che, ai fini della separazione, ben assimilano entrambe le situazioni. Per mantenere la generalità della trattazione mi riferirò a mixtures e sorgenti. Inoltre tutti i segnali audio considerati sono digitali.

1.2 Modelli di mixing nel dominio del tempo

Posto I il numero delle mixtures e J il numero delle sorgenti, il problema della separazione può essere [2]:

- Sotto-determinato, $I<J$ nel caso in cui il numero di mixtures sia inferiore al numero di sorgenti in esso presenti;
- Determinato o sovra-determinato, $I \geqslant J$ nel caso in cui il numero di mixtures sia maggiore o uguale al numero di sorgenti.

Riguardo all'ambiente di mixing o di registrazione, si può distinguere in misture di tipo *convolutivo*, in presenza di fenomeni di riverbero, e di tipo istantaneo, nel caso di anecoicità.

Indicando le *mixtures* con $x_i,\ i=1,..,I$ e con $s_j,\ j=1,..,J$ i segnali sorgente, un modello di mixing istantaneo può essere espresso nel dominio del tempo come:

$$x_i(t)=\sum_{j=1}^{J} a_{ij} s_j(t), \tag{1}$$

dove gli a_{ij} rappresentano gli elementi della matrice di mixing $\boldsymbol{A}$.

Infatti, il modello può essere espresso alternativamente in forma matriciale [3]:

$$\boldsymbol{x}(t)=\boldsymbol{A}\,\boldsymbol{s}(t), \tag{1.1}$$

dove

$$\boldsymbol{x}(t)=\begin{bmatrix} x_1(t) \\ \dots \\ x_I(t) \end{bmatrix},\quad \boldsymbol{s}(t)=\begin{bmatrix} s_1(t) \\ \cdots \\ \dots \\ s_J(t) \end{bmatrix} \text{ e } \boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \dots & \dots & & \dots \\ a_{I1} & a_{I2} & \dots & a_{1J} \end{bmatrix} \tag{1.2}$$

In ambiente *convolutivo*, il problema si può rappresentare come [2]:

$$x_i(t)=\sum_{j=1}^{J} \sum_{\tau=-\infty}^{+\infty} a_{ij}(t-\tau,\tau)\, s_j(t-\tau), \tag{2}$$

oppure, in forma più compatta:

$$\boldsymbol{x}(t) = \boldsymbol{A} * \boldsymbol{s}(t). \tag{2.1}$$

1.3 Esempi di mixtures reali

Esaminando il mondo reale, due esempi tipici di mixture sono:

- *le registrazioni ambientali* di discorsi in conferenze oppure di eventi musicali, come nel caso di orchestre nella musica classica [1]. In entrambi i casi, la registrazione audio è affidata a un numero di microfoni nettamente inferiore al numero di sorgenti sonore. L'impostazione spaziale dei microfoni e le caratteristiche degli stessi determinano la quantità di interferenze e di riverbero su ciascun canale di registrazione. Si tratta di mixtures di tipo sotto-determinato, convolutivo e tempo variante.
- *Le mixtures sintetiche.* I mix audio sintetici nascono dal processo di "mixdown" di più segnali, precedentemente registrati, che confluiscono all'interno di una traccia audio stereo (L/R) [1]. Tra gli effetti sintetici tipici del mixaggio abbiamo il *panning*, la compressione, l'equalizzazione e l'aggiunta di riverbero sintetico. Tali mixtures sono spesso di tipo sotto-determinato, istantaneo e tempo varianti.

1.4 Casi di studio

In questo lavoro di tesi, verrà esaminato il problema della separazione determinato e sotto-determinato di mixtures istantanee. Verranno studiati due algoritmi che si basano su due diversi approcci e, per mezzo della loro implementazione in ambiente *Matlab*, verranno applicati a dati reali.

Capitolo 2: Sparsità di un segnale

2.1 L'ipotesi alla base della separazione

La matrice di mixing, nel caso **(1.2)**, contiene le informazioni su come i vari segnali vengano miscelati in maniera costante nel tempo. Se, ad esempio, il processo di mixing coinvolgesse tre sorgenti e due mixtures, il modello risulterebbe [7]:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix} \qquad \textbf{(1.3)}$$

che evidentemente può essere descritto come una trasformazione lineare da uno spazio di dimensione *J* a uno spazio di dimensione *I*, in questo caso con *J=3* e *I=2*.

Dalla (1.3) è evidente che se una sola sorgente fosse attiva, supponiamo $s_1(t)$, le mixtures risultanti sarebbero:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} s_1(t) = \boldsymbol{a_1} s_1(t). \qquad \textbf{(1.4)}$$

e i punti dello *scatter plot*[1] di $x_1(t)$ su $x_2(t)$ giacerebbero su una linea orientata dal vettore $\boldsymbol{a_1}$. L'ipotesi alla base di questo modello è che le sorgenti abbiano una rappresentazione *sparsa* rispetto a una data base.

1. Per *scatter plot* o *grafico di dispersione* s'intende un tipo di grafico in cui due variabili di un set di dati sono riportate in uno spazio cartesiano. Questa rappresentazione è utile per determinare il grado di *correlazione ,* cioè di dipendenza lineare, tra le due variabili.

2.2 La sparsità nel dominio del tempo

Un segnale si dice *sparso* nel tempo, se la sua ampiezza è pari a zero durante la maggior parte della sua durata [3]. Più generalmente, un segnale è sparso se la sua ampiezza è pari a zero, o quasi, più di quanto ci si dovrebbe aspettare dalla sua *varianza*, definita nel caso discreto come:

$$\sigma_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \qquad \textbf{(3)}$$

dove n è il numero totale di osservazioni e $\bar{x}$ è il valore medio.

La *Sparse Component Analysis* è basata sull'ipotesi di sparsità dei segnali. Tuttavia, è stato dimostrato che segnali audio come i discorsi e le registrazioni musicali presentano una maggiore sparsità in frequenza piuttosto che nel tempo [3], dove il fenomeno è meno visibile, Figura 3. Questa considerazione sposta l'attenzione dal dominio del tempo al dominio della frequenza.

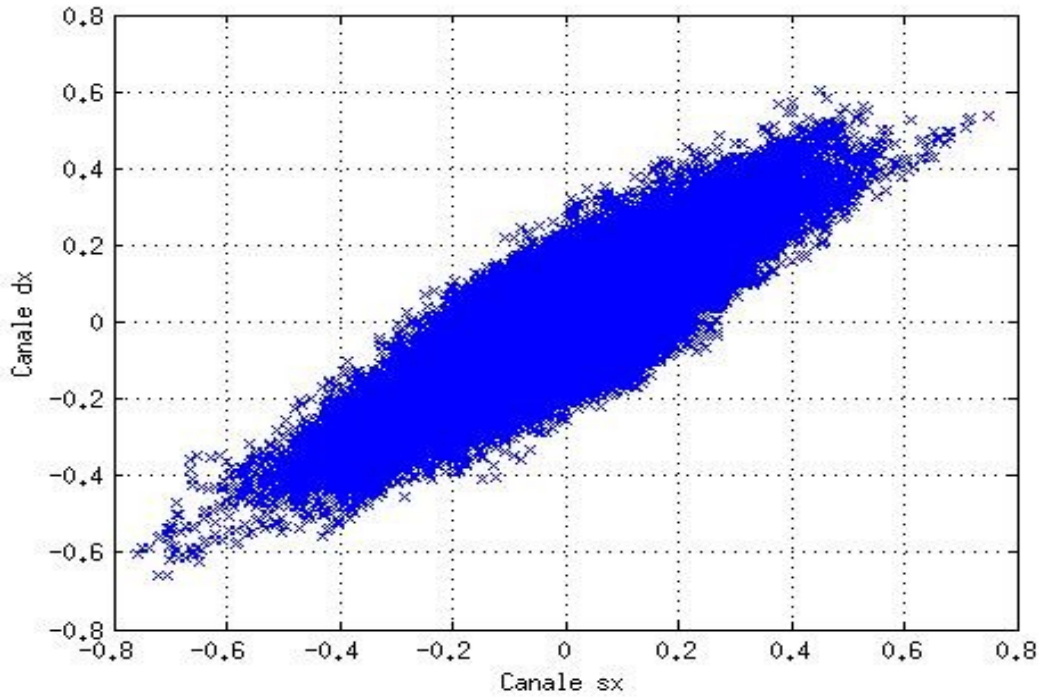


Figura 3. Scatter plot del canale sx e dx di un brano audio nel dominio del tempo.

2.3 Il dominio della frequenza

2.3.1 La Trasformata discreta di Fourier (DFT) e la FFT

La *trasformata discreta di Fourier* DFT associa al vettore x(n) con *n=0,..,N-1* il vettore *X(k)* con *k=0,1, ...,N-1* nel seguente modo:

$$X(k)=\sum_{n=0}^{N-1} x(n)e^{-i\frac{2\pi}{N}kn} \quad \textbf{(4)}$$

La trasformata discreta di Fourier è lineare e invertibile. La sua inversa è detta *antitrasformata,* IDFT, ed è data da:

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{i\frac{2\pi}{N}kn}. \quad \textbf{(5)}$$

La complessità di calcolo della DFT è di tipo $O(N^2)$. La Fast Fourier Transform (FFT) è un algoritmo ottimizzato che consente di calcolare la DFT con una complessità ridotta a $O(N\,log(N))$.

2.3.2 La Trasformata a tempo breve di Fourier (STFT).

La DFT evidenzia la presenza delle componenti armoniche ma non permette di ricavare facilmente informazioni su quando e come tali frequenze siano effettivamente presenti. Se i segnali sono stazionari, l'analisi con la DFT fornisce tutte le informazioni utili mentre nel caso di segnali non stazionari, la DFT potrebbe risultare inadeguata, per via della mancanza di informazioni temporali. Occorre inserire nella trasformazione una dipendenza dal tempo. Un modo per farlo consiste nel rendere locale la DFT, non operando più sull'intero segnale nel dominio del tempo ma su singole porzioni di esso, ottenute moltiplicandolo per una finestra temporale che trasla nel tempo. Questa particolare trasformata, prende il nome di *Short Time Discrete Fourier Transform* (STDFT) ed è definita da:

$$X(n,k)=\sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-i\frac{2\pi}{N}kn} \quad \textbf{(6)}$$

dove $f_n[m]=x[m]w[n-m]$ è una breve porzione temporale del segnale $\mathrm{x}[m]$ al tempo n. Inoltre si definisce lo spettrogramma come $S(n,k)=log(|(X(n,k))|)^{2.}$.

La scelta della dimensione della finestra $w[n]$ determinerà la risoluzione nel tempo e in frequenza, Figura, infatti:

- A una lunga finestra temporale, bassa risoluzione temporale, corrisponderà una maggiore risoluzione in frequenza e uno spettrogramma a bande strette, Figura 5 (c).
- A una breve finestra temporale e quindi una alta risoluzione temporale, corrisponderà una minore risoluzione in frequenza frequenza. Lo spettrogramma sarà costituito da bande larghe, Figura 5 (d).

Al limite $w \to \infty$ la STDFT corrisponde concettualmente alla DFT.

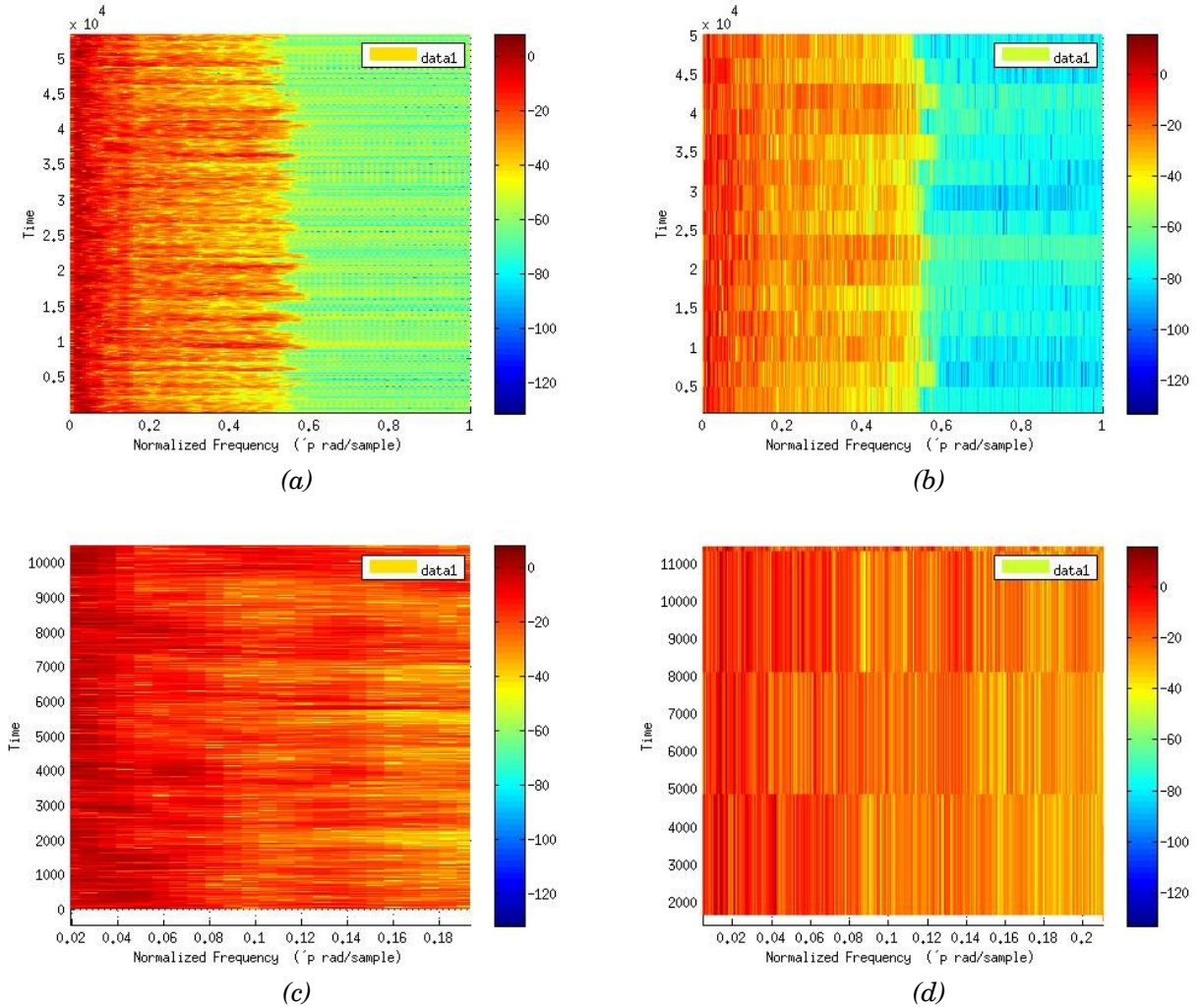


Figura 5. Spettrogramma di un sample audio, con finestra di Hanning, overlapping=128 samples (a) N=1024 samples (b) N=20384 samples, (c) dettaglio di (a), (d) dettaglio di (b).

Anche la scelta del tipo di finestra ha una grande importanza. Tuttavia, si è scelto di considerare solo la finestra di von Hann, Figura 6, come in [3],[4], che risulta definita da:

$$w[n]=0.5\left(1-\cos\left(\frac{2\pi\ n}{N-1}\right)\right) \qquad \textbf{(7)}$$

con $0\leq n\leq N-1$ e N è l'ampiezza della finestra, in numero di campioni.

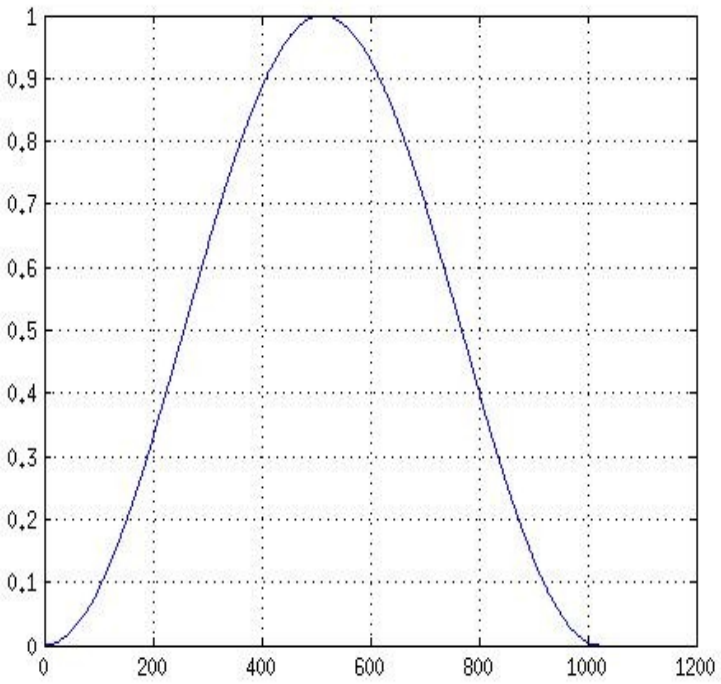


Figura 6. Finestra di von Hann o Hanning, con N=1024.

La STDFT è generalmente[2] invertibile e la sua inversa è data da:

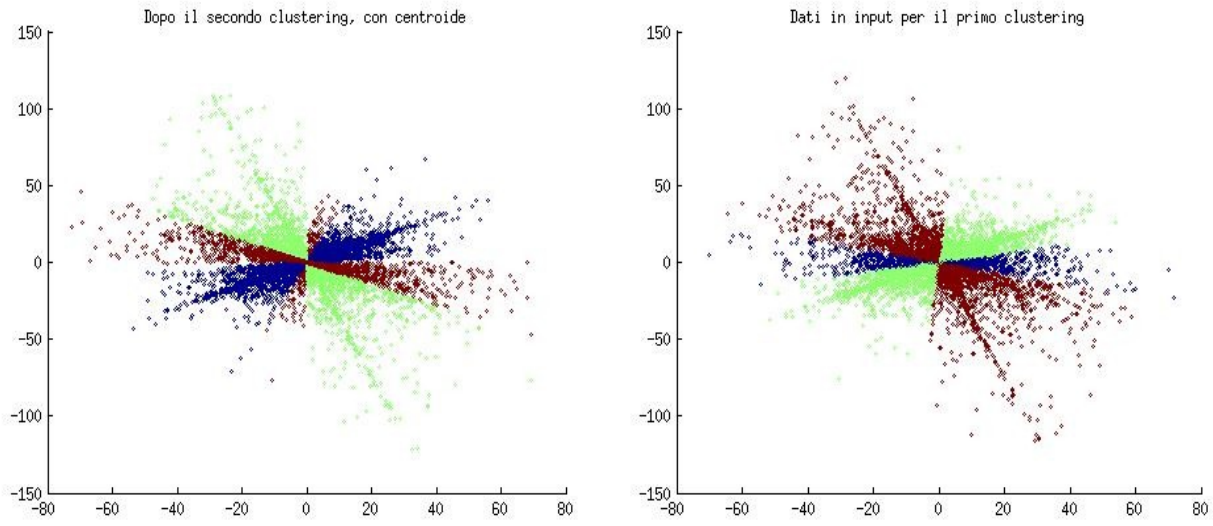
$$x(n)=\frac{1}{2\pi w[0]}\int_{-\pi}^{\pi} X(n,k)w(n-m)e^{i\frac{2\pi}{N}kn}\,dk \qquad \textbf{(8)}$$

2. Nel caso in cui *w[n]* sia a banda limitata, con banda B, se l'intervallo di campionamento $2\pi/N$ è maggiore di B, l'operazione di trasformazione non risulta invertibile.

2.4 La sparsità nel dominio della frequenza

Nell'ipotesi di sparsità, la probabilità che più componenti delle sorgenti abbiano contemporaneamente ampiezza diversa da zero è bassa e questo indica che questa situazione si verifica frequentemente [7].

Di conseguenza, sia la parte reale che quella immaginaria dei coefficienti della DFT delle sorgenti seguono delle distribuzioni distinte all'interno dello *scatter plot* delle mixtures. Queste distribuzioni possono essere approssimate da linee orientate, Figura 4. Ciascuna linea è collegata a una sorgente e, in particolare, la direzione di ogni linea corrisponde a una colonna della matrice di mixing [5]. Perciò, stimare le direzioni delle rette, equivale a stimare l'intera matrice $\boldsymbol{A}$.



(a) *(b)*

Figura 4. Scatter plot della parte reale (a) e immaginaria (b) dei coefficienti della DFT dei canali sx e dx di una mixture audio stereo composta da basso, batteria e voce.

In [6] è stato mostrato che in presenza di un alto livello di sparsità, analoghi risultati si possono ottenere basandosi sulla sola direzione del modulo dei coefficienti della DFT, Figura 5, in quanto accade che $|\boldsymbol{X}(n,k)| \simeq \boldsymbol{a}_j |S_j(n,k)|$.

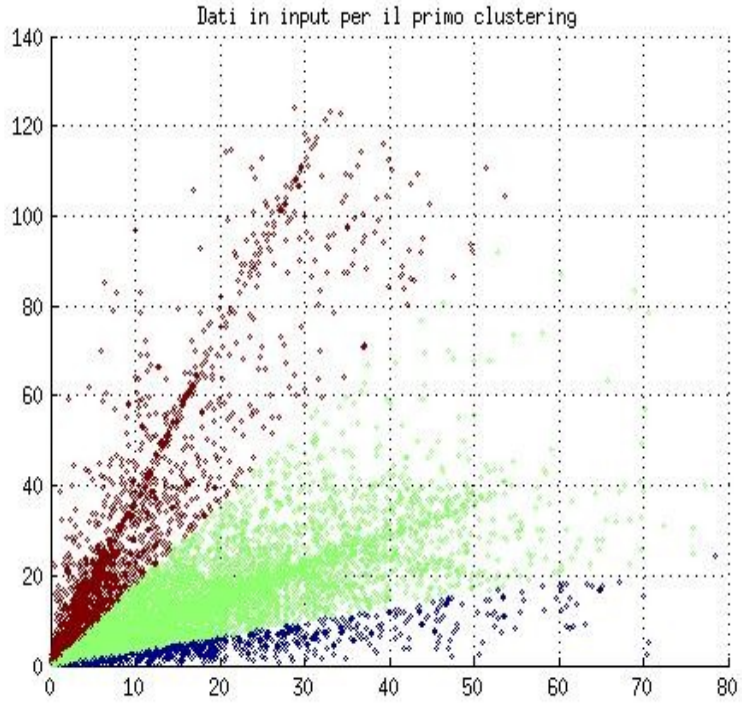


Figura 5. Scatter plot del modulo dei coefficienti della DFT dei canali sx e dx di una mixture audio composta da basso, batteria e voce.

Capitolo 3: Il problema della separazione

3.1 Definizione del problema

Nella grande maggioranza dei casi pratici, il numero di sorgenti è maggiore del numero di segnali mixati. Ma gli algoritmi studiati per la separazione sotto-determinata sono, generalmente, utilizzabili anche nel caso determinato [3]. Per questo motivo ho scelto di effettuare l'analisi di due metodi di separazione nati per il caso sotto-determinato.

Il processo di separazione sarà diviso in due parti, la prima consiste nella stima della matrice di mixing $\boldsymbol{A}$ e la seconda nel recupero dei segnali sorgente.

Per il calcolo della matrice di mixing ho applicato due algoritmi diversi, quello proposto da Namgook Cho, Yu Shiu and C.-C. Jay Kuo [4] e quello proposto da Reju, V. G., Koh, S. N., & Soon, I. Y. [3].

Per il recupero dei segnali sorgente, una volta determinata $\boldsymbol{A}$, ho affrontato solo il problema determinato, tracciando anche una possibile soluzione del caso sotto-determinato.

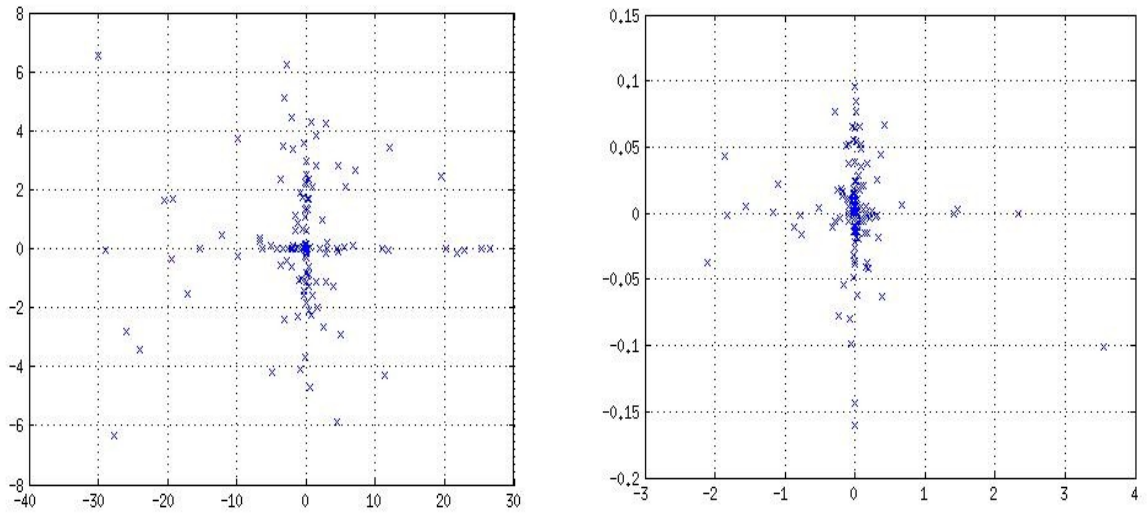
3.2 Considerazioni sui modelli basati sulla STFT

Partendo dal modello **(1.2)** e passando al dominio del tempo-frequenza tramite la STFT, si ottiene:

$$\boldsymbol{X}(n,k)=\boldsymbol{A}\,\boldsymbol{S}(n,k). \qquad \textbf{(9)}$$

con n e k rispettivamente l'indice dei frames temporali e delle bande di frequenza. Per poter ragionare sul modello nel tempo-frequenza occorre fare alcune considerazioni. Lo *scatter plot* di tutti i coefficienti della DFT di due mixtures, corrispondenti a tutti i frame temporali e a tutte le bande di frequenza, genera un grafico in cui si intravedono delle direzioni, ma in maniera molto confusa, Figura 4. Per avere una visione più nitida delle direzioni, ci si può concentrare su una singola banda, fissando, a titolo di esempio, $k=3$. La minore sovrapposizione di dati genera dei grafici più chiari, Figura 6(a), e di conseguenza il *fitting* dei dati sperimentali sarà meno affetto da errore. L'operazione di mixing lineare **(1.2)** è trasparente alla trasformazione in frequenza e di conseguenza le linee orientate dovrebbero mantenere gli stessi angoli di intersezione e la stessa "ampiezza" della nuvola di campioni circostanti in ogni banda di frequenza. Nella

realtà sperimentale si riscontra che per via del rumore e della simultaneità occasionale di occorrenza, la situazione cambia nelle varie bande, Figura 6 (a) *k=3*, (b) *k=100*.



(a) *(b)*

Figura 6. Scatter plot della parte reale della STDFT in tutti i frames temporali e nel bin frequenziale k=3 (a) e k=100 (b) dei canali sx e dx di una mixture audio stereo composta da basso, batteria e voce.

Riassumendo, se si utilizzano tutti i coefficienti, i dati risultano troppo confusi per essere elaborati ma se si utilizzano i coefficienti relativi a una sola banda di frequenza, si rischia di determinare un andamento scorretto. Pertanto, negli algoritmi che determineranno la direzione delle linee orientate, si seguirà la logica illustrata di nella Tabella 1.

Tabella 1

Algoritmo per il *fitting iterativo* dei dati nel tempo-frequenza

```
Step 1: Si parte con due mixtures nel dominio T-F del tipo X_1(n,k),
        X_2(n,k),con n=1:N, k=1:K.
Step 2: Inizializzo un vettore vuoto, che ad ogni iterazione
        concatenerà i risultati dell'iterazione in corso con quelle
        precedenti.
        g_new=[];
Step 3: Effettua il fitting dei dati di X in tutti i tempi e nella
        sola k-esima banda di frequenza, tenendo conto dei dati già
        "adattati" nell'iterazione precedente.
        for k=1:K
             g(1,:,:)=X_1(:,k); %dati relativi alla k-esima banda
             g(2,:,:)=X_2(:,k);
             g_new=[g_new g];  %concatena i dati fittati
                               %nell'iterazione precedente con quelli
                               %dell'iterazione in corso,
                               %non ancora elaborati.
                               %
             var_out=f(g_new); % f è la funzione che effettua il
                               % fitting dei dati
        end
Step 4: Ottieni var_out, che è il risultato del fitting iterativo dei
        dati in tutti i tempi e in tutte le bande di frequenza.
```

Alla luce delle considerazioni fatte nella Tabella 1, a patto di riferirci alla k-esima banda e quindi alla k-esima iterazione, la **(9)** può essere riscritta come:

$$\boldsymbol{X}(n)=\boldsymbol{A}\boldsymbol{S}(n) \qquad \textbf{(10)}$$

Questa formulazione risulterà più comoda della **(9)** per illustrare gli algoritmi che seguiranno.

Capitolo 4: Stima della matrice di mixing

4.1.1 Decomposizione agli autovalori

Sia $\boldsymbol{A}$ una matrice quadrata *nxn,* dotata di *n* autovettori linearmente indipendenti $\boldsymbol{q_i}$, $i=1,\ldots,n$. Allora $\boldsymbol{A}$ può essere fattorizzata come:

$$\boldsymbol{A}=\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1} \qquad \textbf{(11)}$$

dove **Q** è la matrice quadrata *nxn* che contiene come colonne gli autovettori di $\boldsymbol{A}$, mentre $\boldsymbol{\Lambda}$ è la matrice diagonale, i cui elementi diagonali corrispondono agli autovalori $\boldsymbol{\lambda_i}$, $i=1,\ldots,n$.

4.1.2 Metodo basato sulla PCA.

La *Principal Component Analysis* (PCA) è una tecnica che ha lo scopo primario di ridurre un numero più o meno elevato di variabili in alcune variabili latenti.
Ai fini della separazione, la PCA serve alla riduzione dei dati nelle loro componenti principali, stimate attraverso la decomposizione agli autovalori della ″variazione″ degli stessi dati. Più tecnicamente, è stato mostrato che l'orientazione di un cloud lineare di dati corrisponde agli autovettori principali della matrice di covarianza [8].
La matrice di covarianza rappresenta la variazione di ogni variabile rispetto alle altre e a se stessa. Data una popolazione di *n* elementi in cui sono rilevanti *k* caratteri quantitativi $\boldsymbol{X_i}$, $i=1,..,k,$ con elementi x_{hi}, $h=1,..,n,$ la matrice delle covarianze è definita come:

$$\boldsymbol{\Sigma}=\sigma_{ij}=\frac{1}{n}\sum_{h=1}^{n}(x_{hi}-\mu_i)(x_{hj}-\mu_j) \qquad \textbf{(12)}$$

dove μ_j è la media del *carattere j.*
L'algoritmo [4] prevede di conoscere preventivamente una stima delle direzioni delle linee orientate e di fornirne un'approssimazione migliore sfruttando la PCA.
Il problema della determinazione preliminare delle direzioni dei *cluod* verrà affrontato nel paragrafo successivo, per ora si partirà dal presupposto di avere già effettuato la stima.

Il metodo si divide in due parti, la prima consiste nell'assegnare dei pesi ai campioni delle mixtures in base alla corrispondenza delle loro direzioni con quelle stimate, mentre nella seconda si applica la decomposizione agli autovalori delle le matrici di covarianza dei campioni precedentemente pesati, determinando in maniera più precisa le direzioni.

Partendo dalla formulazione **(10)**, ipotizzo di avere due mixtures $\boldsymbol{X_1}(n)$ e $\boldsymbol{X_2}(n)$ e di considerare un unico vettore delle osservazioni $\boldsymbol{X}(n)=\left[\boldsymbol{X_1}(n)\,\boldsymbol{X_2}(n)\right]^{\mathrm{T}}$.

Dal momento che vi sono *J* sorgenti, nello scatter plot saranno presenti J distribuzioni di punti, approssimabili con delle rette che partono dall'origine.

Introduco i vettori normali $\boldsymbol{v_j}$ definiti come:

$$\boldsymbol{v_j}=\begin{bmatrix}\dfrac{-m_j}{\sqrt{1+m_j^2}}\\[2ex]\dfrac{1}{\sqrt{1+m_j^2}}\end{bmatrix} \quad \text{con} \quad j=1,\ldots,I \quad \text{e} \quad \|\boldsymbol{v_j}\|=1 \tag{13}$$

Definisco la distanza tra il vettore $\boldsymbol{v_j}$ e i vettori delle osservazioni $\boldsymbol{X}(n)$:

$$\boldsymbol{z_{j,n}}=dist\left(\boldsymbol{X}(n),\boldsymbol{v_j}\right)=\left|\left(\langle\boldsymbol{v_j^T},\boldsymbol{X}(n)\rangle\right)\right| \tag{14}$$

Se il punto campione $\boldsymbol{X}(n)$ appartenesse alla retta di direzione $\boldsymbol{v_j}$ la distanza sarebbe nulla.

Per dare un peso a tutti i campioni $\boldsymbol{X}(n)$ in base alle distanze $\boldsymbol{z_{j,n}}$, introduco:

$$\hat{\boldsymbol{z}}_{\boldsymbol{j,n}}=\frac{\boldsymbol{z}_{\boldsymbol{j,n}}^{-\tilde{m}}}{\sum_j \boldsymbol{z}_{\boldsymbol{j,n}}^{-\tilde{m}}}, \tag{15}$$

dove $\tilde{m}$ è un parametro di controllo della "morbidezza" dei pesi[3].

I pesi $\hat{\boldsymbol{z}}_{\boldsymbol{j,n}}$ variano in maniera inversa alla distanza, vale a dire che i punti allineati con il vettore $\boldsymbol{v_j}$ avranno uno *score* maggiore, mentre quelli che si discostano dall'andamento, avranno uno *score* minore.

3. Come mostrato in [4], il valore di $\tilde{m}$ ideale nei problemi tipici è compreso tra *1* e *2*.

Supponendo di conoscere approssimativamente la direzione di tre distribuzioni ideali di punti presenti nelle stesse osservazioni, Figura 7, calcolando gli $\hat{z}_{1,n}$ si otterrebbero valori molto elevati in corrispondenza delle osservazioni che seguono la direzione più simile a v_1 e valori molto bassi in corrispondenza delle altre osservazioni.

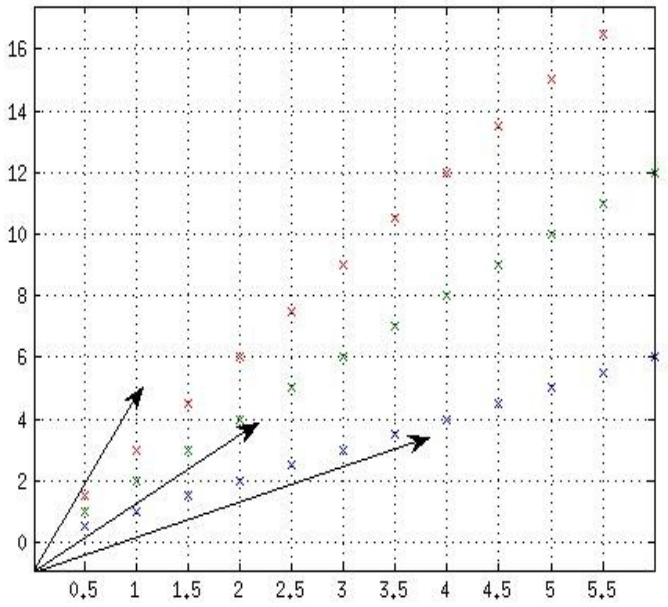


Figura 7. Esempio di tre distribuzioni di punti allineate e tre vettori che ne approssimano la direzione molto grossolanamente. Situazione prima dell'applicazione della PCA.

Definisco le matrici di covarianza dei campioni pesati associati alla linea *j*:

$$\Sigma_j = \frac{\sum_n \hat{z}_{j,n} \cdot (X(n) - \mu) \cdot (X(n) - \mu)^T}{\sum_n \hat{z}_{j,n}} \qquad \textbf{(16)}$$

dove μ è la media dei valori delle righe di $X(n)$.

Eseguendo la scomposizione agli autovalori delle matrici Σ_j, si ottiene:

$$\boldsymbol{\Sigma}_j = \boldsymbol{U}_j \cdot \boldsymbol{\Lambda}_j \cdot \boldsymbol{U}_j^{-1} \tag{17}$$

dove $\boldsymbol{U}_j$ contiene gli autovettori di Σ_j, e la matrice diagonale $\boldsymbol{\Lambda}_j$ contiene gli autovalori $\lambda_{1,\dots},\lambda_J$ associati agli autovettori. I nuovi vettori orientati corrispondono agli autovettori principali di Σ_j, $\boldsymbol{v}_j^{new} = \boldsymbol{u}_{j,max}$, dove $\boldsymbol{u}_{j,max}$ è l'autovettore principale corrispondente a λ_{max}. La stima attuale dei $\boldsymbol{v}_j$ è molto più precisa, Figura 8, grazie all'applicazione della PCA.
L'orientazione delle linee, come detto in precedenza, corrisponde alle colonne della matrice di mixing, perciò si ottiene la matrice di mixing stimata:

$$\hat{\boldsymbol{A}} = [\boldsymbol{v}_1^{new} \dots \boldsymbol{v}_j^{new} \dots \boldsymbol{v}_J^{new}]. \tag{18}$$

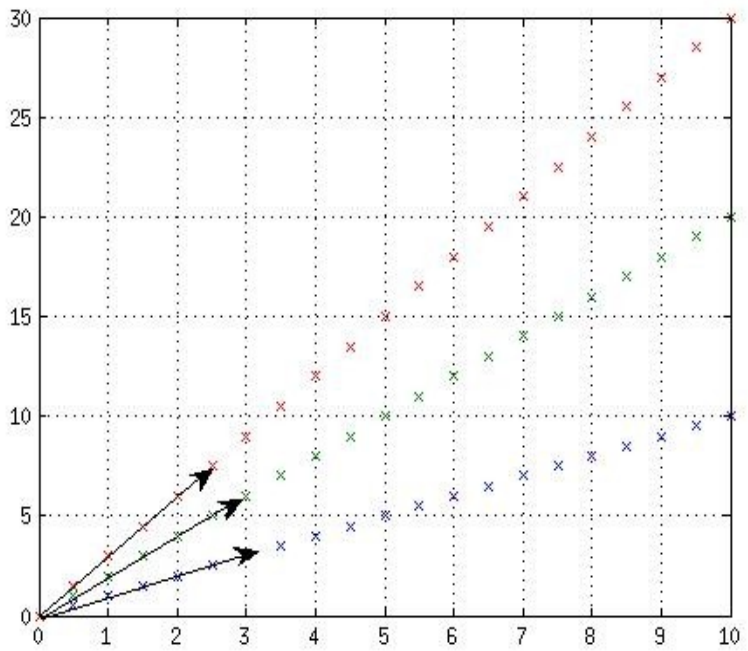


Figura 8. Dopo l'applicazione della PCA, i vettori stimati "fittano" quasi perfettamente il comportamento delle tre distribuzioni. Si tratta di dati sintetici sviluppati ad-hoc per l'esempio.

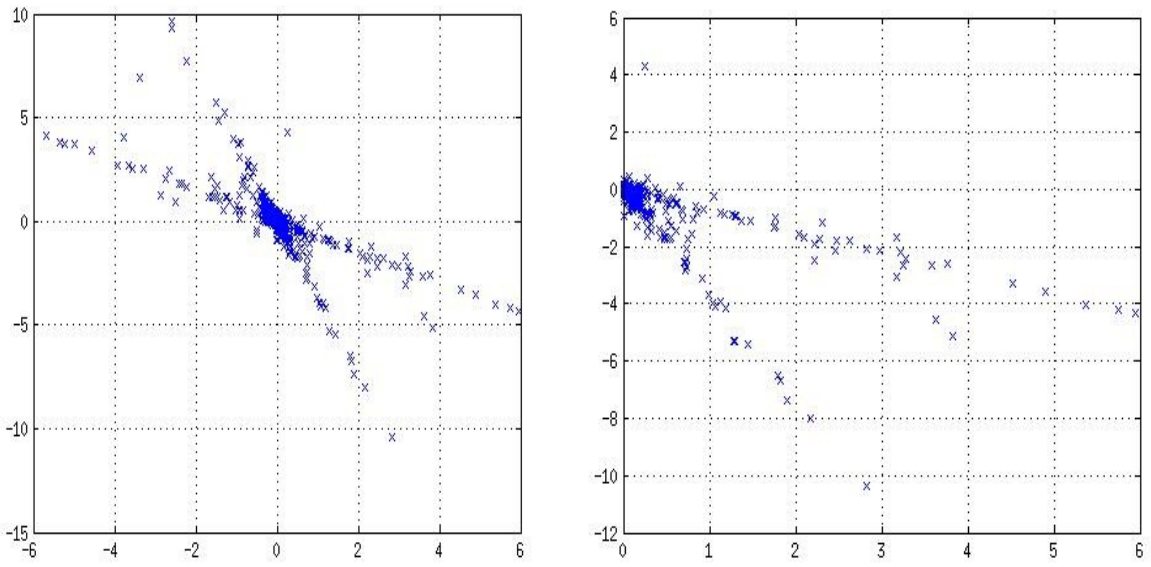
4.1.3 Stima preventiva delle direzioni.

In [7],[8] è stato sviluppato un algoritmo improntato su una base statistica e che, partendo da un'inizializzazione casuale dei vettori $\boldsymbol{v}_j$, applica iterativamente il metodo esposto nel paragrafo precedente, fino alla convergenza della soluzione.

Ho preferito affrontare il problema della stima preventiva delle direzioni utilizzando un approccio più euristico e senza ipotesi statistiche.

Inoltre, va precisato che l'algoritmo è stato pensato per un massimo di tre sorgenti.

Le distribuzioni delle linee da stimare sono sempre simmetriche rispetto all'asse verticale. Partendo da questa ipotesi, ho considerato solo i campioni di $\boldsymbol{X}(n)$ presenti nel *I* e *IV* quadrante, Figura 9. Chiamo $\boldsymbol{X}(\tilde{n})$ il nuovo insieme di campioni e, di questo sottoinsieme, considero solo i campioni con modulo maggiore di una certa soglia $\boldsymbol{X}(\tilde{n}) \leftarrow |\boldsymbol{X}(\tilde{n})| > 0.3$, coerentemente con quanto verificato in [3].



(a) *(b)*

Figura 9. Simmetria nello scatter plot della parte reale della STDFT di un brano audio, in un $\boldsymbol{k}$ *casuale. In ascissa canale sx e in ordinata canale dx. (a) Tutti i coefficienti, (b) solo quelli con ascissa>0.*

Calcolo i coefficienti angolari di tutti i *samples* $\boldsymbol{X}(\tilde{n})$ e li memorizzo, riordinati in ordine crescente, all'interno di un vettore $\boldsymbol{m}_{sort}$. Gli angoli relativi ai coefficienti massimo m_{max} e minimo m_{min} presenti in $\boldsymbol{m}_{sort}$ delimiteranno lo spazio necessario alla determinazione delle direzioni, Figura 10.

Introduco un vettore normale definito come:

$$\boldsymbol{v}_{scan} = \begin{bmatrix} \dfrac{-m_{scan}}{\sqrt{1+m_{scan}^2}} \\ \dfrac{1}{\sqrt{1+m_{scan}^2}} \end{bmatrix} \quad \text{con} \quad m_{scan} \in [m_{min}, m_{max}], \tag{19}$$

$$m_{scan} = tan(\theta_{scan}), \quad \theta_{scan} \in [atan(m_{min}), atan(m_{max})] \tag{20}$$

L'orientazione di $\boldsymbol{v}_{scan}$ è definita dal suo coefficiente angolare, che varierà tra un limite inferiore e superiore **(18)**. Durante il tragitto della retta scanner da sud a nord, si calcola la distanza, definita in **(13)**, tra il punto e i sample $\boldsymbol{X}(\tilde{n})$. La distanza minore si avrà in corrispondenza del centro della distribuzione di punti, che nel tre sorgenti coinciderà con la retta centrale, mentre nel caso di due sorgenti servirà solo a dividere lo spazio nelle due parti in cui saranno effettivamente presenti le rette.

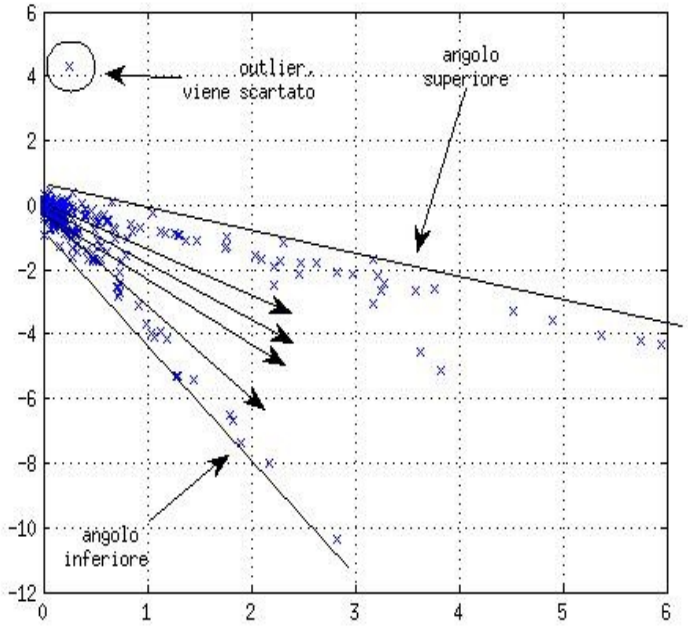


Figura 10. Angolo minimo, angolo massimo e vettore mobile nello scatter plot.

Con un ragionamento analogo, si ripete il calcolo nella parte superiore e inferiore comprese tra la retta scanner e i limiti superiore e inferiore, $\boldsymbol{v}_{scan} < \boldsymbol{v}_{scan_3} \leq \boldsymbol{v}_{super}$ e $\boldsymbol{v}_{inf} \leq \boldsymbol{v}_{scan_2} < \boldsymbol{v}_{scan}$. Questa volta il calcolo delle distanze verrà fatto sui campioni di $\boldsymbol{X}(\tilde{n})$ aventi rispettivamente $m_{scan} < m_{scan_3} \leq m_{super}$ e $m_{inf} \leq m_{scan_2} < m_{scan}$. Le tre rette stimate saranno una stima molto approssimativa degli andamenti reali, ma

l'applicazione della PCA, illustrata nel paragrafo precedente, consente comunque di ottenere buoni risultati.
L'algoritmo è schematizzato nella Tabella 2.

Tabella 2

Algoritmo per la stima preventiva delle direzioni delle rette.

```
Step 1: Sfrutto la simmetria dello scatter plot e scarto i dati
        inferiori alla soglia;
Step 2: Calcolo e i riordino tutti i coefficienti delle osservazioni,
        li memorizzo in ordine crescente in m_sort=[m_sort; ind_m],
        trovando anche m_min e m_max;
Step 3: Definisco m_scan=m_min:m_max;
Step 4: for i=1:length(m_scan)
                → definisco v_scan
                → calcolo dist(X(n),v_scan);
        end
Step 5:         → [~ ind_s]=min(dist(X(n),v_scan));
                → m_scan=m_scan(ind_s);
Step 6: Stesso procedimento per  m_inf, eccetto per il calcolo della
        distanza che va effettuata sui campioni appartenenti alla
        parte superiore e inferiore dello spazio
                → ind_m_inf=m_sort<m_scan;
                → considero solo X_inf=X(:,m_sort(2,ind_m_inf));
                ….
        trovo m_inf
 Step 7 Stesso procedimento,
                  → ind_m_sup=m_sort>m_scan;
                  → X_sup=X(:,m_sort(2,ind_m_sup));
                 ....
        trovo m_sup
```

4.2.1 Single-Source-Points e Multi-Source-Points.

Il modello di mixing istantaneo, trasformato nel dominio del Tempo-Frequenza attraverso la trasformata a tempo breve di Fourier, presente in **(10),** può essere riscritto come:

$$\boldsymbol{X}(n,k)=\boldsymbol{A}\,\boldsymbol{S}(n,k)=\sum_{j=1}^{J}\boldsymbol{a_j}\,S_j(n,k) \qquad \textbf{(21)}$$

dove $\boldsymbol{X}(n,k)=[X_1(n,k),..,X_I(n,k)]^T$ e $\boldsymbol{S}(n,k)=[S_1(n,k),..,S_J(n,k)]^T$ sono i coefficienti della STFT di mixture e sorgenti, nella banda di frequenza k-esima e al frame temporale t e $\boldsymbol{a_j}=[a_{1j},..,a_{Ij}]^T$ è la j-esima colonna della matrice di mixing $\boldsymbol{A}$.

Preso un generico punto (n_1,k_1) nel piano del Tempo-Frequenza, se in quel punto è presente solo la componente della sorgente s_1, allora $S_1(n_1,k_1)\neq 0$ e $S_j(n_1,k_1)=0,$ con $j=\{2,..,J\}$. Perciò l'equazione **(21)** diventerà:

$$\boldsymbol{X}(n_1,k_1)=\boldsymbol{a_1}S_1(n_1,k_1). \qquad \textbf{(22)}$$

Eguagliando la parte reale e immaginaria dei due termini, si ottiene:

$$\Re\{\boldsymbol{X}(n_1,k_1)\}=\boldsymbol{a_1}\,\Re\{S_1(n_1,k_1)\} \qquad \textbf{(23)}$$

$$\Im\{\boldsymbol{X}(n_1,k_1)\}=\boldsymbol{a_1}\,\Im\{S_1(n_1,k_1)\} \qquad \textbf{(24)}$$

Dalle equazioni precedenti, si può vedere come la direzione assoluta di $\Re\{\boldsymbol{X}(n_1,k_1)\}$ e di $\Im\{\boldsymbol{X}(n_1,k_1)\}$ sia la stessa e, in particolare, sia quella di $\boldsymbol{a_1}$.

In maniera analoga, preso un altro punto (t_2,k_2) del piano TF, se in quel punto è presente solo la sorgente s_2, avrò che $S_2(n_2,k_2)\neq 0$ e $S_j(n_2,k_2)=0,$ con $j=1\cup\{3,..,J\}$. In questo caso, scriverò:

$$\Re\{\boldsymbol{X}(n_2,k_2)\}=\boldsymbol{a_2}\,\Re\{S_2(n_2,k_2)\}, \qquad \textbf{(25)}$$

$$\Im\{\boldsymbol{X}(n_2,k_2)\}=\boldsymbol{a_2}\,\Im\{S_2(n_2,k_2)\}, \qquad \textbf{(26)}$$

Quindi in (n_2, k_2) la parte reale e immaginaria di $\boldsymbol{X}(n_2, k_2)$ avranno la stessa direzione di $\boldsymbol{a_2}$. I punti (n_1, k_1) e (n_2, k_2) sono dei *Single-Source-Point (SSP)*, ovvero punti nel piano TF dove è presente solo una sorgente.
Ora, si consideri un terzo punto (n_3, k_3) dove sono presenti i contributi di entrambe le sorgenti. In questo caso, si parla di *Multi-Source-Point*, ovvero di un punto nel piano TF delle mixtures in cui è presente più di un punto. In (n_3, k_3) le direzioni assolute di parte reale e immaginaria di $\boldsymbol{X}(n_3, k_3)$ saranno:

$$\begin{aligned} \Re\{\mathbf{X}(\mathrm{n}_3,\mathrm{k}_3)\} &= \mathbf{a_1}\Re\{\mathrm{S}_1(\mathrm{n}_3,\mathrm{k}_3)\} + \mathbf{a_2}\Re\{\mathrm{S}_2(\mathrm{n}_3,\mathrm{k}_3)\}, \\ \Im\{\boldsymbol{X}(n_3,k_3)\} &= \boldsymbol{a_1}\Im\{S_1(n_3,k_3)\} + \boldsymbol{a_2}\Im\{S_2(n_3,k_3)\}. \end{aligned} \tag{27}$$

Dalle ultime due equazioni, si nota come parte reale e immaginaria di $\boldsymbol{X}(n_3, k_3)$ possano avere la stessa direzione, solo se

$$\frac{\Re\{S_1(n_3,k_3)\}}{\Im\{S_1(n_3,k_3)\}} = \frac{\Re\{S_2(nt_3,k_3)\}}{\Im\{S_2(n_3,k_3)\}} \tag{28}$$

La probabilità che questo accada è molto basso, come dimostrato in [3].
Se la direzione assoluta di $\Re\{\boldsymbol{X}(n,k)\}$ e di $\Im\{\boldsymbol{X}(n,k)\}$ in un dato (n,k) è la stessa, allora questo sarà un *SSP*, diversamente si tratterà di un *MSP*.
Per il caso più generico, in presenza di *I* mixtures e *J* sorgenti, in un *MSP* (n,k), posso scrivere parte reale e immaginaria come:

$$\Re\{\boldsymbol{X}(n,k)\} = \sum_{j=1}^{J} \boldsymbol{a_j}\Re\{S_j(n,k)\}, \tag{29}$$

$$\Im\{\boldsymbol{X}(n,k)\} = \sum_{j=1}^{J} \boldsymbol{a_j}\Im\{S_j(n,k)\}, \tag{30}$$

e l'angolo tra esse compreso come:

$$\theta = cos^{-1}\left(\frac{\Re\{\boldsymbol{X}(n,k)\}^T \Im\{\boldsymbol{X}(n,k)\}}{\sqrt{\Re\{\boldsymbol{X}(n,k)\}^T \Re\{\boldsymbol{X}(n,k)\}}\sqrt{\Im\{\boldsymbol{X}(n,k)\}^T \Im\{\boldsymbol{X}(n,k)\}}}\right) \quad \textbf{(31)}$$

$$cos^{-1}\left(\frac{\sum_{i=1}^{I}\left(\left(\sum_{j=1}^{J} a_{ij}\Re\{S_j(n,k)\}\right)\left(\sum_{j=1}^{J} a_{ij}\Im\{S_j(n,k)\}\right)\right)}{\sqrt{\sum_{i=1}^{I}\left(\sum_{j=1}^{J} a_{ij}\Re\{S_j(n,k)\}\right)^2 \sum_{i=1}^{I}\left(\sum_{j=1}^{J} a_{ij}\Im\{S_j(n,k)\}\right)^2}}\right) \quad \textbf{(32)}$$

Nell'equazione precedente, θ diventerà *0°* o *180°* se

$$\frac{\Re\{S_1(n,k)\}}{\Im\{S_1(n,k)\}} = \ldots = \frac{\Re\{S_j(n,k)\}}{\Im\{S_j(n,k)\}} = \ldots = \frac{\Re\{S_J(n,k)\}}{\Im\{S_J(n,k)\}}. \quad \textbf{(33)}$$

Perciò se la direzione di $\Re\{\boldsymbol{X}(n,k)\}$ e $\Im\{\boldsymbol{X}(n,k)\}$ è la stessa in un punto del piano TF o il punto è un *SSP*, oppure il rapporto tra parte reale e immaginaria dei coefficienti della trasformata di Fourier di tutti i segnali deve essere lo stesso. Poiché, come già detto in precedenza, la seconda opzione è molto poco probabile e con l'aumento del numero di sorgenti la probabilità tenderà a diminuire ulteriormente, si può concludere che i *SSPs* sono i punti in cui la direzione assoluta di $\Re\{\boldsymbol{X}(n,k)\}$ e $\Im\{\boldsymbol{X}(n,k)\}$ è la stessa.

Nella pratica è difficile incontrare *SSPs* in situazioni in cui le ampiezze di tutte le sorgenti tranne una sono pari a zero. La condizione si può approssimare dicendo che l' ampiezza di una sorgente deve essere significativamente maggiore delle altre, o, in alternativa:

$$\left|\frac{\Re\{\boldsymbol{X}(n,k)\}^T \Im\{\boldsymbol{X}(n,k)\}}{\|\Re\{\boldsymbol{X}(n,k)\}\| \, \|\Im\{\boldsymbol{X}(n,k)\}\|}\right| > cos(\Delta\theta) \quad \textbf{(34)}$$

dove $\|\boldsymbol{y}\| = \sqrt{\boldsymbol{y}^T\boldsymbol{y}}$. I campioni che rispettano la condizione sopra, vengono considerati corrispondenti a *SSPs* , memorizzati nell'array $\widetilde{\boldsymbol{X}}$ e utilizzati per il clustering.

Per effettuare il clustering è sufficiente utilizzare o la parte reale o quella immaginaria di $\boldsymbol{X}$ nei *SSP*, perché le direzioni di una o dell'altra sono le stesse a meno di $\Delta\theta$.

In Tabella 3 è schematizzato l'algoritmo per la selezione dei campioni SSP.

Tabella 3

Algoritmo per la selezione dei Single-Source-Points

Step 1:	Trasformazione di $\boldsymbol{x}(t)$ in $\boldsymbol{X}(n,k)$.
Step 2:	Verifica della condizione (16).
Step 3:	Se la condizione **(34)** è soddisfatta, allora $\boldsymbol{X}(n,k)$ è un campione SSP che può essere utilizzato nella stima della matrice di mixing. Se la condizione non è soddisfatta, si scarta il punto.
Step 4:	I campioni SSP vengono memorizzati nell'array $\widetilde{\boldsymbol{X}}$.
Step 5:	Si ripetono gli Step 2, 3 e 4 finché non si ottiene un numero sufficiente di campioni corrispondenti a SSP.

4.2.2 Il clustering

Il clustering è un processo di raggruppamento di un insieme di oggetti fisici o astratti in classi di oggetti simili [11]. Un cluster è una collezione di oggetti simili tra loro, che sono dissimili rispetto agli oggetti degli altri cluster.

Nell'analisi dei dati, questi algoritmi vengono utilizzati per studiare come gli stessi dati si distribuiscono nello spazio.

Partendo da un insieme di istanze, descritte da un insieme di attributi, e una misura di similarità, lo scopo del clustering è trovare un insieme di classi tali che le istanze appartenenti alla stessa classe risultino simili e quelle appartenenti a classi differenti risultino dissimili.

Alla base di questo ragionamento c'è l'ipotesi che si possa definire una *funzione di similarità***,** che determini, appunto, la similarità tra due istanze.

Nel caso d'interesse, la similarità può essere interpretata come la distanza tra due cluster e il problema può essere riformulato come la ricerca di classi in cui le distanze intracluster siano minime e le distanze intercluster siano massime.

4.2.3 Clustering gerarchico bottom-up.

In questo particolare metodo, si parte da un cluster per ogni istanza e, ad ogni passo, si raggruppano due cluster, finché non c'è un solo cluster. In questo modo, si forma una gerarchia di suddivisioni, rappresentabile con un *dendrogramma.*

Tabella 4

Algoritmo di clustering gerarchico bottom-up

Step 1: Si parte con un cluster per ogni istanza.
Step 2: Determina i due cluster c_i e c_j più simili.
Step 3: Sostituisci c_i e c_j con un singolo cluster $c_i \cup c_j$.
Step 4: Ripeti gli Step 2 e 3 finché non c'è un solo cluster.

Nota la metrica di similarità tra due istanze, occorre definire come calcolare la similarità tra due cluster (*Linkage*).
In [3] è stato utilizzato l'*Average Linkage Clustering*, ovvero la la media delle distanze tra le istanze appartenenti alle due classi.

4.2.4 Stima della matrice di mixing attraverso il clustering

Come misura della distanza tra due istanze è stato utilizzata [3] la similarità del coseno $1-|(cos(\vartheta))|$ dove $cos(\vartheta)$ è il coseno dell'angolo compreso tra i vettori colonna campione *m*-esimo e *n*-esimo, $\tilde{\boldsymbol{X}}_m$, $\tilde{\boldsymbol{X}}_n \in \tilde{\boldsymbol{X}}$, definito da:

$$cos(\vartheta) = \frac{\tilde{\boldsymbol{X}}_m^T \tilde{\boldsymbol{X}}_n}{\|\tilde{\boldsymbol{X}}_m\| \|\tilde{\boldsymbol{X}}_n\|}. \qquad \textbf{(35)}$$

Nel clustering gerarchico, i dati sono partizionati in diverse classi attraverso la sezione del dendrogramma in corrispondenza della giusta distanza, come mostrato in Fig.2.

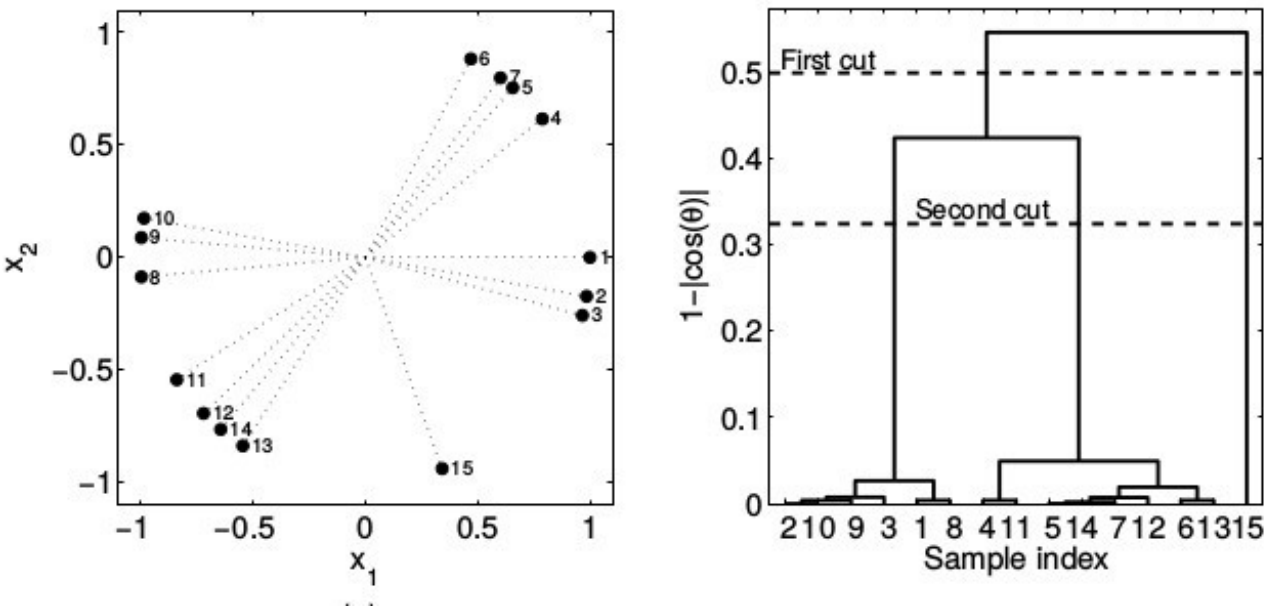


Figura 11. Clustering gerarchico (a) scatter plot (b) dendrogramma. Fonte [3]

In presenza di dati fuorvianti nella collezione iniziale da sottoporre al clustering, è necessario tener conto della corretta distanza in corrispondenza della quale verrà sezionato il dendrogramma e, di conseguenza, del corretto numero di clusters da considerare. Ad esempio, nella figura è mostrato un caso in cui il numero di clusters desiderati è due. Se si divide il dendrogramma in due, il primo cluster sarà costituito da un punto affetto da errore (punto *15*), mentre il secondo sarà costituito dai restanti punti. In questo caso per eliminare il dato errato, sarà necessario formare tre clusters e scartare quello a cui appartengono il numero minore di punti.

Più in generale, se sono presenti Q sorgenti, ci dovranno essere Q clusters validi. In [3] si assume che al di fuori dei clusters validi, il cluster con il numero minimo di campioni contenga almeno il *5%* del numero medio di campioni presenti nei rimanenti cluster validi. Si assume anche che il numero minimo di dati errati sia minore del *5%* del numero di campioni presenti nei cluster validi. Per risolvere il problema, occorre sezionare il dendrogramma in modo da ottenere Q clusters e, se questi non rispettano le condizioni discusse, il dendrogramma verrà sezionato per formare $Q+1$ clusters.

Il processo viene ripetuto finché le condizioni non vengono rispettate oppure non si verifica che il numero di cluster sia pari a due volte il numero di sorgenti.

Dal momento che $\tilde{\boldsymbol{X}}$ conterrà solo campioni di tipo *SSP*, lo scatter plot sarà quasi perfettamente orientato lungo le direzioni dei vettori colonna della matrice di mixing e i punti in $\tilde{\boldsymbol{X}}$ verranno suddivisi in Q gruppi.

Dopo aver effettuato il clustering, le colonne vettore della matrice di mixing vengono determinate calcolando il *centroide*[4] di ciascun cluster.

Nel caso di una mixture stereo, prima di procedere con il calcolo del centroide, i punti alla sinistra dell'asse verticale vengono cambiati di segno così da evitare valori molto piccoli dei vettori colonna della matrice di mixing.

L'errore nella stima della matrice di mixing può essere fortemente ridotto rimuovendo i punti che sono lontani dalla direzione media del cluster. L'i-esimo campione appartenente al q-esimo cluster viene rimosso se si verifica che

$$\left|\phi_q(i)-\mu_{\phi_q}\right|>\varepsilon\sigma_{\phi_q} \qquad \textbf{(36)}$$

dove ε è una costante, σ_{ϕ_q} è la deviazione standard delle direzioni dei campioni appartenenti al q-esimo cluster, $\phi_q(\mathrm{i})$ è la direzione assoluta del i-esimo campione nel q-esimo cluster e μ_{ϕ_q} è la media delle direzioni assolute dei campioni nel q-esimo cluster. Nelle applicazioni su dati reali è stata scelta $\varepsilon=0.5$.

4. Il *centroide* è definito come la media di tutti i punti appartenenti al cluster.

Capitolo 5: Ricostruzione delle sorgenti

5.1 Separazione determinata.

Nel caso di separazione determinata di misture di segnali istantanee e stazionarie, la matrice di mixing $\boldsymbol{A}$ sarà quadrata, di dimensione *IxI,* costante e a valori reali.
Il problema della ricostruzione delle sorgenti, si riduce al calcolo nel dominio del tempo di [7]:

$$\boldsymbol{s}(t)=(\boldsymbol{A})^{-1}\cdot\boldsymbol{x}(t). \tag{37}$$

Il problema è invertibile e la soluzione risulta determinata.

5.2 Separazione sotto-determinata.

5.2.1 Decomposizione ai valori singolari (SVD)

Sia $\mathbf{A}\in\mathbb{C}^{mxn}$, allora esistono due matrici unitarie $\boldsymbol{U}\in\mathbb{C}^{mxm}$ e $\boldsymbol{V}\in\mathbb{C}^{nxn}$ tali che:

$$\boldsymbol{U}^{*}\boldsymbol{A}\,\boldsymbol{V}=\boldsymbol{\Sigma}=diag(\sigma_{1,..},\sigma_{p},...,0,...,0)\in\mathbb{C}^{mxn} \tag{38}$$

Con $p=rank(\boldsymbol{A})$ e $\sigma_1>...>\sigma_p>0$. La fattorizzazione descritta da **(11)** prende il nome di SVD e i σ_i rappresentano i valori singolari [10]. La SVD permette di ridurre qualsiasi matrice in forma diagonale tramite delle moltiplicazioni per matrici unitarie.

5.2.2 La matrice pseudo-inversa

Nel caso in cui $I<J$ il problema è sotto-determinato e la matrice di mixing sarà rettangolare, di dimensione *IxJ* e non invertibile.
Tra i tanti approcci esistenti [2], si è scelto di risolvere il sistema:

$$\boldsymbol{s}(t)=\boldsymbol{A}^{+}\boldsymbol{x}(t), \tag{39}$$

in cui $\boldsymbol{A}^{+}$ è la matrice *pseudo-inversa* di Moore-Penrose, definita come [10]:

$$\boldsymbol{A}^{+}=\sum_{i}^{rank(A)}\boldsymbol{v_i}\sigma_i^{-1}\boldsymbol{u_i^T} \tag{40}$$

dove σ_i sono i valori singolari, $\boldsymbol{v_i}$ e $\boldsymbol{u_i}$ sono detti vettore singolare destro e sinistro definiti dalle relazioni $\boldsymbol{A}\boldsymbol{v_i}=\sigma\boldsymbol{u_i}$, con $i=1,..,M$ o $\boldsymbol{A}^{*}\boldsymbol{u_i}=\sigma_i\boldsymbol{v_i}$ con $i=1,..,n$. In

questo caso, ci sono infinite soluzioni e la pseudo-inversa seleziona la soluzione avente *norma-2* minima.

Un modo computazionalmente non troppo oneroso di calcolare la matrice pseudoinversa è legato alla Decomposizione ai Valori Singolari (SVD).

In maniera analoga, se $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\,\boldsymbol{V}^{*}$ è la decomposizione ai valori singolari di $\boldsymbol{A}$, allora $\boldsymbol{A}^{\dagger}=\boldsymbol{U}\boldsymbol{\Sigma}^{\dagger}\boldsymbol{V}^{*}$. Nel caso di una matrice diagonale rettangolare come $\boldsymbol{\Sigma}$ la pseudo-inversa si ottiene calcolando il reciproco di ciascun elemento diverso da zero presente nella diagonale, lasciando gli zeri nelle loro posizioni e trasponendo la matrice. Nel calcolo numerico, si definisce una soglia e tutti gli elementi con valori maggiori della soglia, vengono considerati come diversi da zero, mentre gli altri vengono rimpiazzati da zeri.

Capitolo 6: Qualità della separazione

6.1 Misure della qualità nella stima della matrice di mixing

La misura della qualità nella stima della matrice di mixing [3] viene effettuata attraverso l'applicazione del *Normalized Mean Square Error (NMSE)*, definito come:

$$NMSE = 10 \log_{10} \frac{\sum_{p,q} (\hat{a}_{pq} - a_{pq})^2}{\sum_{p,q} (a_{pq})^2} \tag{41}$$

dove gli $\hat{a}_{pq}$ indicano gli elementi della matrice stimata.

6.2 Misure della qualità globale della separazione

Per la valutazione della performance della separazione, ho utilizzato il toolbox *BSS EVAL* [12]. La misura della performance si basa sul principio secondo cui una data sorgente stimata possa essere scomposta come:

$$\hat{s}(t) = s(t) + e_{interf}(t) + e_{noise}(t) + e_{artif}(t) \tag{42}$$

dove $s(t)$ è la sorgente originale, $e_{interf}(t)$ è il rumore dovuto all'interferenza con altre sorgenti, $e_{noise}(t)$ è un termine di perturbazione dovuto al rumore, tipicamente Gaussiano e $e_{artif}(t)$ è il rumore dovuto agli artefatti creati dal processo di separazione. Il rumore introdotto da ciascuno dei termini della (**42**) viene stimato come:

1. *Source-to-Artifact Ratio (SAR)*, che determina il livello di artefatti presenti nella separazione:

$$SAR = 10 \log_{10} \frac{\|s + \varepsilon_i + \varepsilon_n\|^2}{\|\varepsilon_a\|^2} \tag{43}$$

2. *Source-to-Interferences-Ratio (SIR),* che misura il livello di interferenza tra la sorgente stimata e le altre:

$$SIR = 10 \log_{10} \frac{\|s\|^2}{\|\varepsilon_i\|^2} \quad \textbf{(44)}$$

3. *Source-to-Distortion-Ratio (SDR),* che fornisce un indice globale sulla performance del criterio di separazione:

$$SDR = 10 \log_{10} \frac{\|s\|^2}{\|\varepsilon_a + \varepsilon_i + \varepsilon_n\|^2} \quad \textbf{(45)}$$

Tutti gli indici presenti sopra sono espressi in *dB*.

Capitolo 7: Risultati sperimentali

7.1 Considerazioni sul codice e sui test

Entrambi gli algoritmi proposti sono stati sviluppati in ambiente Matlab. Il metodo basato sul clustering, descritto nel Capitolo 4.2, [3], è stato implementato dallo stesso autore dell'articolo [13]. Nella realizzazione del mio codice ho preso spunto dalla sua implementazione, integrando alcune parti al resto del progetto da me sviluppato. Tutte le mixtures utilizzate nei test sono state create a partire da sample audio scaricati da *FreeSound* [14], nel formato formato .WAV a 44100Hz e 16 bit. La durata temporale di tutti i samples e delle risultanti mixtures è di *4s*.

I parametri relativi alla STFT, definiti nel paragrafo 2.3.2, sono stati fissati ai valori *K=1024 samples* e *overlapping*[5]*=128 samples*.

Nel primo algoritmo, paragrafo 4.1.2, ho fissato il valore $\tilde{m}=2$ e la condizione sui moduli $\boldsymbol{X}(\tilde{n}) \leftarrow |\boldsymbol{X}(\tilde{n})| > 0.3$.

Nel secondo algoritmo, paragrafo 4.2.1, ho impostato come angolo soglia $\Delta\theta = 0.2°$, condizione **(34),** mentre per la condizione sui moduli ho fissato una soglia identica a quella del primo algoritmo.

Nel caso determinato, ho calcolato l'errore nella stima della matrice basandomi su un set di tre mixtures stereo, costituite rispettivamente da una chitarra e un basso, (*mixture 1)* una voce e una batteria (*mixture 2)* e un sintetizzatore e un coro (*mixture 3).*

Nel caso sotto-determinato, ho eseguito i test considerando tre mixtures stereo formate da una chitarra, un basso e una batteria (*mixture 1.2),* un coro, un sintetizzatore e una voce femminile (*mixture 2.2*) e una chitarra, una voce femminile e un coro (*mixture 3.2*).

La separazione completa è stata effettuata nel caso di *due mixtures e due sorgenti* e di conseguenza anche il calcolo degli indici *SAR, SIR* e *SDR* verrà riportato per il solo caso *2x2.*

Tutti i test sono stati fatti senza l'aggiunta di rumore.

5. Per *overlapping* si intende il numero di campioni che si sovrappongono da una finestra all'altra.

7.2 Numero di condizionamento di una matrice

Dato un sistema lineare $\boldsymbol{Ax}=\boldsymbol{b}$, con $\boldsymbol{A}\in\mathbb{R}^{nxn}$ e $\boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^{n}$, si definisce *numero di condizionamento di una matrice* la quantità [9]:

$$k(\boldsymbol{A})=\|\mathbf{A}\|\|\mathbf{A}^{-1}\| \tag{46}$$

Nel caso di matrici non quadrate, la **(46)** diventa:

$$k(\boldsymbol{A})=\|\boldsymbol{A}\|\|\boldsymbol{A}^{\dagger}\| \tag{47}$$

dove $\boldsymbol{A}^{\dagger}$ è la matrice *pseudo-inversa di Moore-Penrose* definita in **(40)**.

7.3 Errore nella matrice di mixing stimata

Nel calcolo della matrice di mixing ha notevole importanza il suo condizionamento. Per un valore di $k(\boldsymbol{A})$ alto si avranno dei vettori colonna della matrice **A** quasi linearmente dipendenti. Di conseguenza, le linee orientate presenti nello scatter plot saranno più difficili da determinare. Tuttavia, come mostreranno i risultati dei test, l'errore sulla stima della matrice di mixing risulta poco influenzato dal valore di $k(\boldsymbol{A})$, che invece inciderà sul processo di recupero delle sorgenti. Per mettere a confronto i due algoritmi, ho effettuato delle prove con un numero di condizionamento alto[6] e uno basso.

7.3.1 NMSE: due mixtures e due sorgenti.

Tabella 5.

	$k(\boldsymbol{A})\simeq 3.08$			$k(\boldsymbol{A})\simeq 17.94$		
	$mixture_1$	$mixture_2$	$mixture_3$	$mixture_1$	$mixture_2$	$mixture_3$
NMSE PCA	-43.23dB	-19.29dB	-57.37dB	-51.39dB	-30.60dB	-68.99dB
NMSE Clustering	-49.63dB	-76.21dB	-64.64dB	-56.46dB	-73.21dB	-20.36dB

6. “alto” e “basso” relativamente a valori tipici, come quelli usati in [4].

Tabella 6.

	$k(\boldsymbol{A}) \simeq 3.08$			$k(\boldsymbol{A}) \simeq 17.94$		
	$mixture_1$	$mixture_2$	$mixture_3$	$mixture_1$	$mixture_2$	$mixture_3$
$t_{calcolo}$ *PCA*	0.09s	0.55s	0.16s	0.36s	5.25s	0.80s
$t_{calcolo}$ *Clustering*	0.45s	1.36s	0.79s	1.94s	1.49s	9.10s

7.3.2 NMSE: due mixtures e tre sorgenti.

Il sistema assume la forma:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix} \quad \textbf{(1.3)}$$

I risultati sperimentali sono riportati di seguito:

Tabella 7

	$k(\boldsymbol{A}) \simeq 5.48$			$k(\boldsymbol{A}) \simeq 19.84$		
	$mixture_{1.2}$	$mixture_{2.2}$	$mixture_{3.2}$	$mixture_{1.2}$	$mixture_{2.2}$	$mixture_{3.2}$
NMSE PCA	-35.53dB	-44.63dB	-48.03dB	-39.01dB	-43.18dB	-45.76dB
NMSE Clustering	-11.07dB	-12.58dB	-16.16dB	-20.48dB	-19.94dB	-24.38dB

Tabella 8

	$k(\boldsymbol{A}) \simeq 5.48$			$k(\boldsymbol{A}) \simeq 19.84$		
	$mixture_{1.2}$	$mixture_{2.2}$	$mixture_{3.2}$	$mixture_{1.2}$	$mixture_{2.2}$	$mixture_{3.2}$
$t_{calcolo}$ *PCA*	0.26s	0.70s	0.50s	1.92s	1.02s	0.93s
$t_{calcolo}$ *Clustering*	0.12s	0.44s	0.34s	0.92s	2.66s	2.27s

7.4 Prestazioni ottenute nella ricostruzione delle sorgenti

Tabella 9

$k(\boldsymbol{A}) \simeq 3$	*PCA*						*Clustering*					
	$mixture_1$		$mixture_2$		$mixture_3$		$mixture_1$		$mixture_2$		$mixture_3$	
	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$
*SIR (*dB)	12.18	1.26	-3.9	-1.36	5.52	8.39	12.30	1.37	8.12	5.82	5.56	8.38
SAR (dB)	276.73	270.80	270.71	273.76	271.40	274,53	283.73	282.14	278.93	290.93	274.21	279.89
SDR (dB)	18.18	1.26	-3.9	-1.36	5.52	8.39	12.30	1.37	8.12	5.82	5.56	8.38

Tabella 10

$k(\boldsymbol{A}) \simeq 18$	*PCA*						*Clustering*					
	$mixture_1$		$mixture_2$		$mixture_3$		$mixture_1$		$mixture_2$		$mixture_3$	
	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$	$\hat{s}_1$	$\hat{s}_2$
*SIR (*dB)	-2.12	3.52	-0.59	-9.65	2.92	-1.38	-1.72	3.61	0.34	-2.82	13.76	-29.25
SAR (dB)	247.62	255.53	251.31	257.32	253.1	251.25	255.79	252.61	263.18	254.04	255.64	255.09
SDR (dB)	-2.12	3.52	-0.59	-9.65	2.92	-1.38	-1.72	3.06	0.34	-2.82	13.76	-29.25

7.5 Analisi dei risultati

Nel caso di mixtures composte da due sorgenti, Tabella 5, il clustering ha permesso di ottenere una migliore stima della matrice di mixing, a discapito di un tempo di calcolo molto maggiore di quello impiegato dall'altro algoritmo, Tabella 6. Di conseguenza, la ricostruzione delle sorgenti, Tabella 9 e 10, è stata effettuata in maniera più accurata nel caso del clustering, salvo qualche eccezione. In generale, si è ottenuto un livello quasi inesistente di artefatti, a cui corrisponde un alto valore dell'indice SAR. Come conseguenza, si ottengono valori quasi identici di SDR e SIR. La stima della matrice di mixing non viene influenzata dal cattivo condizionamento, che però ha conseguenze negative nel recupero delle sorgenti per la risoluzione del problema inverso presentato nel Capitolo 5.

Nel caso di mixtures composte da tre sorgenti, Tabella 7, la PCA ha permesso di ottenere risultati migliori rispetto al clustering, anche se impiegando un tempo di calcolo leggermente maggiore, Tabella 8.

Conclusioni

Con questa tesi ho analizzato dei metodi per la *Blind Audio Source Separation* di mixtures istantanee, fornendo una soluzione completa nel caso determinato e affrontando parzialmente il caso sotto-determinato. In questo modo ho delineato le caratteristiche principali di una tecnica vasta e complessa, che negli anni ha subito continui mutamenti. Oggi la tendenza si sta spostando verso algoritmi *semi-Blind* [2], basati sulla stessa struttura studiata in questo lavoro ma nei quali il processo di separazione viene guidato da una serie di informazioni relative al problema specifico, dando luogo a risultati più precisi e fruibili.

Un possibile sviluppo di questa tesi è legato al completamento della separazione nel caso sotto-determinato e alla sua applicazione in chiave *semi-Blind* alle mixtures reali di tipo istaneo, ad esempio nel restauro di vecchi supporti musicali.

Bibliografia

[1] Bruce Bartlett, Jenny Bartlett. *Tecniche di registrazione*, Apogeo, (2010), pp. 319-342.

[2] Emmanuel Vincent, Nancy Bertin, Rémi Gribonval, Frédéric Bimbot. *From blind to guided audio source separation: How models and side information can improve the separation of sound,* IEEE Signal Processing Magazine, Institute of Electrical and Electronics Engineers (IEEE), 31 (3), pp.107-115, (2014).

[3] Reju, V. G., Koh, S. N., & Soon, I. Y. *An algorithm for mixing matrix estimation in instantaneous blind source separation. Signal Processing,* 89, 1762-1773. (2009).

[4] Namgook Cho, Yu Shiu and C.-C. Jay Kuo, *An Improved Technique for Blind Audio Source Separation ,* Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06) , (2006).

[5] P. Bofill and M. Zibulevsky. *Blind separation of more sources than mixtures using sparsity of their short-time fourier transform.* In Int. Conf. Independent Component Anal., pages 87–92, Helsinki, Finland, (2000).

[6] Pau Bofill. *Identifying Single Source Data for Mixing Matrix Estimation in Instantaneous Blind Source Separation,* Proceedings of the 18th international conference on Artificial Neural Networks, Part I (ICANN '08), Springer-Verlag, (2008).

[7] Paul D. O'Grady, Barak A. Pearlmutter, *The LOST Algorithm: Finding Lines and Separating Speech Mixtures,* EURASIP Journal on Advances in Signal Processing Article ID 784296, (2008).

[8] Paul D. O'Grady and Barak A. Pearlmutter, *Soft-LOST: EM on a Mixture of Oriented Lines.* ICA2004,Granada, Spain, pp 430-436 (2004).

[9] Giuseppe Rodriguez, *Algoritmi Numerici*, Pitagora Editrice, (2008)

[10] Per Christian Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, (2000), pp. 19-22

[11] J. Han, M. Kamber, A. K. H. Tung, *Spatial Clustering Methods in Data Mining: A Survey*, H. Miller and J. Han editions, Geographic Data Mining and Knowledge Discovery, Taylor and Francis, (2001).

[12] C. Févotte, R. Gribonval and E. Vincent, *BSS EVAL Toolbox User Guide*, IRISA Technical Report 1706, Rennes, France, April (2005). *http://www.irisa.fr/metiss/bss eval/*.

[13] Reju, V, *Mixing matrix estimation in instantaneous blind source separation,* (2010). *http://www.mathworks.com/matlabcentral/fileexchange/28537-mixing-matrix-estimation-in-instantaneous-blind-source-separation*

[14] *FreeSound*, *http://www.freesound.org*, Universitat Pompeu Fabra Barcelona, Music Technology Group