

Progetto MATLAB
Fondamenti di Calcolo Numerico

Emanuele Intagliata
Prof. Carlo De Falco

A.A. 2018-2019

Indice

1	Introduzione	2
1.1	Organizzazione del paper	2
2	Trattazione teorica	3
2.1	Il problema e le ipotesi	3
2.1.1	Modellizzazione dei segnali audio come processi AR . . .	3
2.2	Presentazione del metodo di interpolazione	4
2.2.1	Calcolo dei coefficienti del modello AR	7
2.2.2	Calcolo dei campioni incogniti	7
3	L'algoritmo	8
3.1	Versione iterativa	8
3.2	Scelta del metodo di stima dei parametri del modello AR	8
3.3	chol Matlab vs algoritmo consigliato	9
3.4	Alcune considerazioni qualitative sull'errore	9
4	Test e risultati	10

1 Introduzione

Questo progetto vuole illustrare un algoritmo adattivo per la ricostruzione di sample mancanti in segnali audio campionati che possono essere localmente descritti attraverso processi autoregressivi.

In particolare, in questo elaborato discutiamo l'implementazione in ambiente MATLAB dell'algoritmo presentato nel paper [1].

Il metodo descritto nel paragrafo 3 usa l'informazione contenuta nei sample vicini noti. Inoltre esso è il primo passo per un algoritmo iterativo nel quale ad ogni passo la stima corrente dei sample mancanti è usata per calcolarne una nuova stima.

Il metodo è infine testato su segnali artificialmente generati, su segnali musicali e su segnali vocali.

1.1 Organizzazione del paper

...

2 Trattamento teorica

2.1 Il problema e le ipotesi

Il paper [1] tratta il problema della ricostruzione (o interpolazione) di sample mancanti in segnali campionati, in particolare in segnali audio. Viene presentato un algoritmo capace di ricostruire in modo soddisfacente i sample mancanti.

Le ipotesi che facciamo sono:

- La *posizione* dei sample mancanti è nota
- I sample mancanti sono circondati da un numero sufficiente ampio di sample dal valore noto
- Il segnale è a banda limitata¹

Le sequenze contenenti sample mancanti possono presentarsi in gruppi isolati (*bursts*) o in pattern sparsi e casuali.² L'algoritmo presentato risulta efficace in entrambi i casi.

La stima del valore dei sample mancanti è ottenuta minimizzando la somma dei quadrati dell'errore residuo. Un'analisi statistica contenuta nel paper [1, Section II.B] dimostra che, per un *burst* di sample mancanti, l'errore quadratico medio atteso per sample converge alla varianza del segnale quando la lunghezza della sequenza mancante tende a infinito.

2.1.1 Modellizzazione dei segnali audio come processi AR

Si assume che i segnali da interpolare possano essere modellizzati come *processi auto-regressivi (AR) di ordine finito*³.

Il metodo presentato nel paper [1] è *adattivo* nel senso che, dato un set finito di dati, per prima cosa stimiamo i parametri del modello AR. Noti questi ultimi, i sample ignoti possono essere ottenuti come *soluzione di un sistema lineare di equazioni*.

In realtà sia i parametri del modello AR che i sample ignoti potrebbero essere ottenuti in un unico passaggio minimizzando una funzione dipendente da entrambi. Tuttavia questa funzione contiene termini del quarto ordine e minimizzarla è un problema non banale. In questo progetto, come anche nel paper [1], adotteremo un **approccio sub-ottimale**. In questo approccio i parametri del modello AR sono stimati da un set incompleto di dati. Successivamente viene fatta una stima del valore dei sample mancanti. Questo può essere considerato il primo step di un algoritmo iterativo rapidamente convergente [cita].

Si assuma che s_k , $k = -\infty, \dots, +\infty$, sia una realizzazione di un processo

¹Ipotesi sempre verificata per segnali audio digitali

²Il numero di sample sconosciuti è da intendersi come numero totale dei sample mancanti.

³La scelta di modellizzare i segnali audio come processi AR può essere motivata dal fatto che molti segnali che incontriamo comunemente (segnali vocali, segnali audio tonali) possono essere così modellizzati.

auto-regressivo stazionario \tilde{s}_k , $k = -\infty, \dots, +\infty$. Ciò significa che esistono un intero positivo p , detto **ordine** del modello AR, e dei numeri a_1, \dots, a_{p+1} , con $a_1 = 1$ detti **coefficienti** del modello AR. Sia inoltre \tilde{e}_k , $k = -\infty, \dots, +\infty$ un processo di **rumore bianco** con varianza σ_e^2 tale che:

$$a_1 \tilde{s}_k + a_2 \tilde{s}_{k-1} + \dots + a_{p+1} \tilde{s}_{k-p} = \tilde{e}_k \quad (1)$$

con $k = -\infty, \dots, +\infty$.

Per convenzione si assuma che $a_k = 0$ per $k < 1$ o $k > p + 1$.

I **dati disponibili** consistono nel segmento s_k , $k = 1, \dots, N$ di una realizzazione del processo AR \tilde{s}_k , $k = -\infty, \dots, +\infty$. Come ipotizzato nell'introduzione, si assuma che i sample incogniti si presentino agli istanti di tempo noti $t(1), \dots, t(m)$, dove:

$$1 < (p + 1) \leq t(1) < \dots < t(m) \leq N - p$$

Il **problema** in esame consiste nella *stima* degli m sample incogniti $s_{t(1)}, \dots, s_{t(m)}$, dei p parametri del modello AR a_2, \dots, a_{p+1} e di σ_e^2 dai dati disponibili in modo che la sequenza ricostruita sia il più coerente possibile con il modello assunto. Quantitativamente la ricostruzione dovrà essere tale da minimizzare la somma dei quadrati dell'errore residuo e_{p+1}, \dots, e_N .

Esistono molti metodi per la stima dell'ordine di processi auto-regressivi (come riportato nell'articolo [2]), così come delle funzioni MATLAB già implementate. Nonostante ciò, il paper [1] in esame assume che, se p è incognito, verrà scelto p in funzione di m , numero dei sample incogniti:

$$p = 3m + 2 \quad (2)$$

Questa relazione piuttosto arbitraria si è dimostrata dare buoni risultati di interpolazione.

2.2 Presentazione del metodo di interpolazione

Adottiamo la seguente nomenclatura:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_{p+1} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} s_{t(1)} \\ \vdots \\ s_{t(m)} \end{bmatrix}$$

Inoltre siano $\hat{\mathbf{a}}$ la stima di \mathbf{a} e $\hat{\mathbf{x}}$ la stima di \mathbf{x} .

Come anticipato, la stima di \mathbf{a} e \mathbf{x} è formulata sotto forma di *problema di minimizzazione*. Sceglierò $\hat{\mathbf{a}}$ e $\hat{\mathbf{x}}$ tali che a funzione

$$Q(\mathbf{a}, \mathbf{x}) = \sum_{k=p+1}^N \left| \sum_{l=1}^{p+1} a_k s_{k-l} \right|^2 = \sum_{k=p+1}^N |e_k|^2 \quad (3)$$

sia minima in funzione di \mathbf{a} e \mathbf{x} .

Determinati $\hat{\mathbf{a}}$ e $\hat{\mathbf{x}}$, potrò stimare σ_e^2 come:

$$\hat{\sigma}_e^2 = \frac{1}{N - p - m} Q(\hat{\mathbf{a}}, \hat{\mathbf{x}}) \quad (4)$$

La particolare scelta di minimizzare la funzione $Q(\mathbf{a}, \mathbf{x})$ per ottenere le stime di \mathbf{a} e \mathbf{x} ha molteplici motivazioni, discusse nel dettaglio nel paper [1]. In particolare, nell'*Appendice A* si dimostra che, ipotizzando che i valori dei campioni abbiano una funzione di densità di probabilità di tipo Gaussiano, minimizzare $Q(\mathbf{a}, \mathbf{x})$ rispetto a \mathbf{x} sia lo stesso che trovare la minima stima della varianza di \mathbf{x} , noti p ed \mathbf{a} .

Si nota immediatamente come minimizzare la funzione $Q(\mathbf{a}, \mathbf{x})$ rispetto a \mathbf{a} e \mathbf{x} sia un problema tutt'altro che banale in quanto essa contiene termini del quarto ordine come $a_2^2 s_{t(m)}^2$. Useremo quindi un **approccio sub-ottimale**:

1. Scegliamo una stima iniziale $\hat{\mathbf{x}}^{(0)}$ per il vettore \mathbf{x} dei sample incogniti, per esempio $\hat{\mathbf{x}}^{(0)} = \mathbf{0}$.
2. Successivamente minimizziamo $Q(\mathbf{a}, \hat{\mathbf{x}}^{(0)})$ in funzione di \mathbf{a} per ottenere una stima di $\hat{\mathbf{a}}$.
3. Infine minimizziamo $Q(\hat{\mathbf{a}}, \mathbf{x})$ in funzione di \mathbf{x} per ottenere una stima di $\hat{\mathbf{x}}$ dei sample incogniti.

Entrambe le minimizzazioni sono possibili in quanto $Q(\mathbf{a}, \mathbf{x})$ è quadratica sia in funzione di $\mathbf{a} \in \mathbb{R}^p$ che in funzione di $\mathbf{x} \in \mathbb{R}^m$. Si può dimostrare che

$$Q(\mathbf{a}, \mathbf{x}) = \mathbf{a}^T C(\mathbf{x}) \mathbf{a} + 2\mathbf{a}^T \mathbf{c}(\mathbf{x}) + c_{11}(\mathbf{x}) \quad (5)$$

dove $C(\mathbf{x})$ è la *matrice di auto-covarianza* di dimensioni $p \times p$ definita come:

$$C(\mathbf{x}) = (c_{ij}(\mathbf{x}))_{i,j=2,\dots,p+1}$$

e

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} c_{1,2}(\mathbf{x}) \\ \vdots \\ c_{1,p+1}(\mathbf{x}) \end{bmatrix}$$

con

$$c_{ij}(\mathbf{x}) = \sum_{k=p+1}^N s_{k-i} s_{k-j}, \quad i, j = 1, 2, \dots, p+1.$$

Allo stesso modo, si può dimostrare che

$$Q(\mathbf{a}, \mathbf{x}) = \mathbf{x}^T B(\mathbf{a}) \mathbf{x} + 2\mathbf{x}^T \mathbf{z}(\mathbf{a}) + D(\mathbf{a}) \quad (6)$$

dove

$$B(\mathbf{a}) = (b_{t(i)-t(j)})_{i,j=1,\dots,m} \quad (7)$$

$$\mathbf{z}(\mathbf{a}) = \begin{bmatrix} z_1(\mathbf{a}) \\ \vdots \\ z_m(\mathbf{a}) \end{bmatrix}$$

con

$$\begin{aligned} z_i(\mathbf{a}) &= \sum_{k=-p}^p b_k s_{t(i)-k}, & i = 1, \dots, m, \\ b_l &= \sum_{k=0}^p a_k a_{k+l}, & l = -p, \dots, +p \end{aligned} \quad (8)$$

and $D(\mathbf{a}) \in \mathbb{R}$ dipendente da \mathbf{a} e dai sample noti. Quindi $\hat{\mathbf{a}}$ e $\hat{\mathbf{x}}$ saranno dati rispettivamente da

$$C(\hat{\mathbf{x}}^{(0)})\hat{\mathbf{a}} = -\mathbf{c}(\hat{\mathbf{x}}^{(0)}), \quad (9)$$

e

$$B(\hat{\mathbf{a}})\hat{\mathbf{x}} = -\mathbf{z}(\hat{\mathbf{a}}). \quad (10)$$

Il metodo appena descritto per il calcolo dei coefficienti del modello AR attraverso una sequenza di campioni è noto come **metodo dell'autocovarianza** [3]. Un metodo diverso, il metodo dell'auto-correlazione, verrà descritto nel paragrafo 3.

Sostituendo la (9) nella (4) segue che

$$\hat{\sigma}_e^2 = \frac{1}{N - p - m} (c_{1,1}(\hat{\mathbf{x}}) + \hat{\mathbf{a}}^T \mathbf{c}(\hat{\mathbf{x}})). \quad (11)$$

Il metodo di interpolazione appena descritto può essere considerato come il primo passo di un **algoritmo iterativo** nel quale, ad ogni passo, nuovi coefficienti $\hat{\mathbf{a}}$ del modello sono stimati attraverso la (9) usando, anziché $\hat{\mathbf{x}}^{(0)}$, il vettore $\hat{\mathbf{x}}$ dei campioni mancanti precedentemente stimato attraverso la (10). Questi coefficienti ottenuti potranno essere adesso usati per ottenere una nuova stima dei sample mancanti e così via. Sembra chiaro, almeno intuitivamente che $Q(\mathbf{a}, \mathbf{x})$ che decresca tendendo a qualche numero non-negativo. Si potrebbe sperare che la sequenza ottenuta converga nel punto in cui $Q(\mathbf{a}, \mathbf{x})$ raggiunge il suo minimo globale. Sfortunatamente, è estremamente difficile provare un risultato del genere.⁴

⁴Si può tuttavia dimostrare che questo procedimento iterativo di minimizzazione assomiglia molto all'algoritmo EM (Expectation-Maximisation) di stima di parametri per le distribuzioni, vedi[1, Appendice B].

2.2.1 Calcolo dei coefficienti del modello AR

Il calcolo di $\hat{\mathbf{a}}$ attraverso la (9) è, in realtà, un problema conosciuto. Noto come **metodo dell'auto-covarianza**, esso è ampiamente discusso nel dettaglio nel paper [3]. Nello stesso paper è esposto un algoritmo molto efficiente per il calcolo di $\hat{\mathbf{a}}$ attraverso la (9) in $O(p^2)$ operazioni: il **metodo dell'auto-correlazione**.⁵

Metodo dell'auto-correlazione In questo metodo anziché risolvere il sistema lineare (9) verrà risolto il sistema

$$\mathbf{R}\hat{\mathbf{a}} = -\mathbf{r}$$

dove \mathbf{R} è la *matrice di auto-correlazione* definita come:

$$\mathbf{R} = (r(i-j))_{i,j=2,\dots,p+1}$$

e

$$\mathbf{r} = \begin{bmatrix} r(2) \\ \vdots \\ r(p+1) \end{bmatrix}$$

dove

$$r(j) = \frac{1}{N} \sum_{k=1}^{N-|j|} s_k s_{k+|j|}, \quad j = -p, \dots, +p$$

è una stima pesata dell'autocorrelazione di j -esimo ritardo di auto-correlazione di $\hat{s}_k, k = -\infty, \dots, +\infty$. Per semplicità, noi risolveremo il sistema con la funzione `mldivide`, nativa in ambiente MATLAB.⁶

2.2.2 Calcolo dei campioni incogniti

Per il calcolo di $\hat{\mathbf{x}}$ attraverso la (10) è utile analizzare la matrice $B(\hat{\mathbf{a}})$ definita nel dettaglio in (9) e (8). Dalla (7) possiamo vedere che $\hat{\mathbf{x}}$ ha valori costanti b_0 sulla sua diagonale principale. Inoltre, la matrice $\hat{\mathbf{x}}$ è **definita positiva**. In ambiente MATLAB questa proprietà è facilmente dimostrabile con l'uso della funzione `chol`.⁷ Il fatto che $B(\hat{\mathbf{a}})$ sia definita positiva ci permette di utilizzare la *decomposizione di Cholesky*⁸ per calcolare $\hat{\mathbf{x}}$ in $O(m^3)$ operazioni.

Decomposizione della matrice $B(\hat{\mathbf{a}})$ Nella decomposizione di Cholesky la matrice $B(\hat{\mathbf{a}})$ è scomposta nel prodotto...

⁵Un calcolo approssimato di $\hat{\mathbf{a}}$ da comunque un risultato di interpolazione soddisfacente: dunque vari e altri metodi possono essere applicati indifferentemente.

⁶Il sistema può essere risolto in $O(p^2)$ operazioni dall'algoritmo di Levinson-Durbin. [4] Tuttavia non ci sono differenze significanti nei risultati di interpolazione ottenuti dall'autore attraverso differenti metodi.

⁷Nella Sezione III.B del paper [1] di riferimento è presente una dimostrazione matematica.

⁸Come illustrato in [1, Sezione III.B], esistono molteplici metodi per la risoluzione di $B(\hat{\mathbf{a}})$. Tuttavia in questo progetto (e nello stesso paper di riferimento) verrà esposta la sola decomposizione di Cholesky.

3 L'algoritmo

L'algoritmo MATLAB in allegato mette in pratica l'*approccio sub-ottimale* qui descritto nella Sezione 2.2. Per prima cosa l'algoritmo controlla che i dati in ingresso rispettino le ipotesi fatte nella Sezione 2.1. Dopo aver definito le variabili, inizia un ciclo iterativo in cui:

1. Viene calcolata una stima dei coefficienti del modello AR tramite la funzione **a_estimator**.
2. Viene calcolata una stima dei campioni incogniti tramite la funzione **x_estimator**. I campioni stimati sostituiscono i campioni incogniti nel segnale ricostruito.
3. Con il segnale così stimato si potrà iterare dal punto (1), trovando una nuova stima dei parametri del modello AR.

L'algoritmo si serve di alcune funzioni ausiliarie.

Funzione a_estimator [a] = a_estimator(sig, p, met)

Funzione x_estimator [sig] = x_estimator(a, t, sig)

L'algoritmo consente all'utente alcune scelte per la ricostruzione del segnale. L'utente può decidere:

- il *numero di iterazioni* per cui ripetere i passaggi descritti sopra
- il *metodo di stima dei parametri del modello AR* (vedi Paragrafo precedente)
- il metodo per il calcolo di σ_e^2

Inoltre, nella versione dimostrativa allegata, l'utente può scegliere quale segnale compromettere e poi ricostruire. I segnali a disposizione sono quelli presentati nei test della Sezione 4.

3.1 Versione iterativa

Appendice B.a

3.2 Scelta del metodo di stima dei parametri del modello AR

Nel nostro progetto, per il calcolo dei coefficienti del modello AR, abbiamo messo a confronto due differenti metodi: il **metodo dell'auto-covarianza** e il **metodo dell'auto-correlazione**. Il metodo dell'auto-covarianza, descritto

nel dettaglio nella Sezione 2.2, è per semplicità implementato attraverso la funzione `arcov` di MATLAB. Questa funzione, tramite appunto il metodo dell'auto-covarianza, stima i parametri di un modello AR di ordine p . Il metodo dell'auto-correlazione è invece interamente implementato come descritto nella Sezione 2.2.1.

Dai test della Sezione 4 è possibile vedere che, nella maggior parte dei casi, non c'è una notevole differenza tra i risultati di interpolazione ottenuti tramite il metodo dell'auto-covarianza o il metodo dell'auto-correlazione.

3.3 chol Matlab vs algoritmo consigliato

3.4 Alcune considerazioni qualitative sull'errore

Riassunto sezione II.B

4 Test e risultati

Text

References

- [1] A. Janssen, R. N. J. Veldhuis, and L. B. Vries, “Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, p. 317–330, Apr 1986.
- [2] H. Akaike, “A new look at the statistical model identification,” *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, December 1974.
- [3] S. M. Kay and S. L. Marple, “Spectrum analysis—a modern perspective,” *Proceedings of the IEEE*, vol. 69, pp. 1380–1419, Nov 1981.
- [4] J. Durbin, “The fitting of time-series models,” *Rev. Inst. Int. Stat.*, 1960.