



UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI SCIENZE STATISTICHE

Corso di Laurea Triennale in Statistica per le Tecnologie e le Scienze



Relazione Finale
**MODELLI E METODI STATISTICI
PER L'ANALISI MUSICALE**

Laureando

Elena Bortolato

Matricola n

1125724

Relatore

Prof.ssa

Alessandra R. Brazzale

Correlatore

Prof. Lorenzo Finesso

ANNO ACCADEMICO 2017/2018

“Ogni spiegazione musicale deve essere anche scientifica.”

A. Schoenberg

Indice

Introduzione	xi
1 Musica e Statistica	1
1.1 Un mondo ordinato, un approccio quantitativo	1
1.2 Caratteristiche e limiti dei dati	5
2 Modelli di Markov e modelli di Markov nascosti	7
2.1 Catene di Markov	7
2.1.1 Definizioni	7
2.1.2 Inferenza	9
2.1.3 Processi Markoviani di ordini superiore	10
2.2 Modelli markoviani nascosti	10
2.2.1 Notazione	10
2.3 Problemi e inferenza	12
2.3.1 Stima di massima verosimiglianza	12
2.3.2 Scelta del modello	15
2.4 HMM per sequenze multivariate	16
2.5 Software	17
3 Confrontare gli HMM	19
3.1 Il caso multivariato	19
3.2 Divergenza di Kullback-Leibler	21
3.3 Mappare i dati	22
3.3.1 Scaling Multidimensionale	22
3.3.2 L'algoritmo t-SNE	24
4 Esempi di applicazione	27
4.1 Dati audio digitali: il formato MIDI	27
4.2 Il pacchetto tuneR	30
4.3 I brani considerati	30
4.4 Calcolo delle similarità	75

4.4.1	Soluzione “naive” e soluzione alternativa	77
4.5	Risultati ottenuti	78
4.6	Un confronto con t-SNE	85
5	Conclusioni	89
A	Codice R utilizzato	91

Elenco delle figure

1.1	Lunghezza della corda in relazione all'altezza dei suoni.	2
1.2	Incisione del Canone Cancrizzante sul nastro di Moebius	3
1.3	Il grafo <i>Tonnetz</i> di Eulero	4
1.4	Struttura circolare del sistema notazionale	6
2.1	Rappresentazione di una catena di Markov mediante un grafo. . .	8
2.2	Grafo di un processo markoviano di secondo ordine	10
2.3	Grafo di un HMM	12
3.1	Rotazione di $5/6\pi$ (5 semitoni) e passaggio dall'accordo di <i>mi minore</i> all'accordo di <i>la minore</i>	20
4.1	Relazione tra frequenza delle onde sonore (Hz) e codifica .MIDI .	29
4.2	Dataset ottenuto con le funzioni del pacchetto tuneR	30
4.3	Linea del tempo e anno di composizione dei brani analizzati . . .	32
4.4	J. S. Bach, Cantata BWV 140, Corale I, tempo e altezza delle note	33
4.5	J. S. Bach, Cantata BWV 140, dataset relativo alle sequenze simultanee	33
4.6	J. S. Bach, Cantata BWV 140, Corale I, diagramma a barre e diagramma a scatola circolare	34
4.7	J. S. Bach, Cantata BWV 140. A sinistra, distribuzione delle sequenze distinte, a destra, rappresentazione delle successioni ordinate e frequenza di missing value	35
4.8	J. S. Bach, Cantata BWV 140, log-verosimiglianza massimizzata e criterio di Akaike al variare del numero di stati nascosti stimati .	36
4.9	Pseudo-Mozart 1, tempo e altezza delle note	38
4.10	Pseudo-Mozart 1, diagramma a barre e diagramma a scatola circolare	38
4.11	Pseudo-Mozart 1, distribuzione delle sequenze distinte	39
4.12	Pseudo-Mozart 1, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	39

4.13	Pseudo-Mozart 2, tempo e altezza delle note	40
4.14	Pseudo-Mozart 2, diagramma a barre e diagramma a scatola circolare	41
4.15	Pseudo-Mozart 2, distribuzione delle sequenze distinte	41
4.16	Pseudo-Mozart 2, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	42
4.17	Pseudo-Mozart 3, tempo e altezza delle note	43
4.18	Pseudo-Mozart 3, diagramma a barre e diagramma a scatola circolare	44
4.19	Pseudo-Mozart 3, distribuzione delle sequenze distinte	44
4.20	Pseudo-Mozart 3, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	45
4.21	Danza Russa, tempo e altezza delle note	46
4.22	Danza Russa, barplot	46
4.23	Danza Russa, distribuzione delle sequenze distinte	47
4.24	Danza Russa, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	47
4.25	Preludio e Fuga BWV 884, J. S. Bach, tempo e altezza delle note	48
4.26	Preludio e Fuga BWV 884, J. S. Bach, diagramma a barre e diagramma a scatola circolare	48
4.27	Preludio e Fuga BWV 884, J. S. Bach, distribuzione delle sequenze distinte	49
4.28	Preludio e Fuga BWV 884, J. S. Bach, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	49
4.29	Tre pezzi per piano, op. 11 n 1, A. Schoenberg, tempo e altezza delle note	50
4.30	Tre pezzi per piano, op. 11 n 1, A. Schoenberg, diagramma a barre e diagramma a scatola circolare	50
4.31	Tre pezzi per piano, op. 11 n 1, A. Schoenberg, distribuzione delle sequenze distinte	51
4.32	Tre pezzi per piano, op. 11 n 1, A. Schoenberg, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	51
4.33	Quartetto per archi n. 12, F. Schubert, tempo e altezza delle note	52
4.34	Quartetto per archi n. 12, F. Schubert, diagramma a barre e diagramma a scatola circolare	52
4.35	Quartetto per archi n. 12, F. Schubert, distribuzione delle sequenze distinte	53
4.36	Quartetto per archi n. 12, F. Schubert, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	53
4.37	Improvviso op. 90 n. 2, F. Schubert, tempo e altezza delle note	54

4.38	Improvviso op. 90 n. 2, F. Schubert, diagramma a barre e diagramma a scatola circolare	55
4.39	Improvviso op. 90 n. 2, F. Schubert, distribuzione delle sequenze distinte	55
4.40	Improvviso op. 90 n. 2, F. Schubert, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	56
4.41	Arabesque op. 18 in do maggiore, R. Schumann, tempo e altezza delle note	57
4.42	Arabesque op. 18 in do maggiore, R. Schumann, diagramma a barre e diagramma a scatola circolare	58
4.43	Arabesque op. 18 in do maggiore, R. Schumann, distribuzione delle sequenze distinte	58
4.44	Arabesque op. 18 in do maggiore, R. Schumann, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	59
4.45	Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, tempo e altezza delle note	60
4.46	Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, diagramma a barre e diagramma a scatola circolare	61
4.47	Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, distribuzione delle sequenze distinte	61
4.48	Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	62
4.49	Notturmo op. 9 n. 2, F. Chopin, tempo e altezza delle note	63
4.50	Notturmo op. 9 n. 2, F. Chopin, diagramma a barre e diagramma a scatola circolare	64
4.51	Notturmo op. 9 n. 2, F. Chopin, distribuzione delle sequenze distinte	64
4.52	Notturmo op. 9 n. 2, F. Chopin, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	65
4.53	The snow is dancing, C. Debussy, tempo e altezza delle note	66
4.54	The snow is dancing, C. Debussy, diagramma a barre e diagramma a scatola circolare	67
4.55	The snow is dancing, C. Debussy, distribuzione delle sequenze distinte	67
4.56	The snow is dancing, C. Debussy, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati	68
4.57	La fille au cheveux de lin, C. Debussy, tempo e altezza delle note	69
4.58	La fille au cheveux de lin, C. Debussy diagramma a barre e diagramma a scatola circolare	70
4.59	La fille au cheveux de lin, C. Debussy, distribuzione delle sequenze distinte	70

4.60	La fille au cheveux de lin, C. Debussy, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati . . .	71
4.61	Danza in ritmo bulgaro n. 1, B. Bartok, tempo e altezza delle note	72
4.62	Danza in ritmo bulgaro n. 1, B. Bartok, diagramma a barre e diagramma a scatola circolare	73
4.63	Danza in ritmo bulgaro n. 1, B. Bartok, distribuzione delle sequenze distinte	73
4.64	Danza in ritmo bulgaro n. 1, B. Bartok, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati . . .	74
4.65	Risultati delle divergenze calcolate mediante simulazione	77
4.66	Matrice delle distanze calcolate simmetrizzando i risultati della divergenza di Kullback- Leibler	79
4.67	Mappa dei punti caldi calcolata a partire dalla matrice delle distanze, con dendrogramma. La numerazione dei brani segue quella presentata all'interno di questo Capitolo.	80
4.68	Scaling in 2D, soluzione "naive"	81
4.69	Scaling in 3D, soluzione "naive"	82
4.70	Scaling in 2D, a partire dalle divergenze calcolate con la penalizzazione del valore della probabilità	84
4.71	Scaling in 3D, a partire dalle divergenze calcolate con la penalizzazione del valore della probabilità	84
4.72	Proiezione dei risultati dell'algorithm t-SNE in uno spazio bidimensionale	86
4.73	Proiezione dei risultati dell'algorithm t-SNE in uno spazio tridimensionale	87

Introduzione

L'obiettivo di questo lavoro è presentare un metodo per analizzare i brani musicali, cercando di identificare alcuni tratti salienti dello “stile” con cui sono stati composti e di mostrare quanto possano essere simili o diversi tra loro. L'idea nasce dalla lettura di un articolo pubblicato nel 2011 su Significance: “Chopin, mazurkas and Markov, Making music in style with statistics”[4]. In questa pubblicazione viene spiegato come sia stato possibile ricreare delle mazurke, particolari forme musicali che hanno origine nella danza popolare polacca, tentando di imitare lo stile del compositore F. Chopin con l'ausilio di computer e statistica.

Definire lo stile di un'opera d'arte, di qualsiasi tipo, o di un autore, è un'operazione delicata, il concetto stesso di *stile* non esiste a priori, ed è in continua evoluzione. In questo elaborato, così come nella ricerca degli autori, l'attenzione viene posta in modo particolare sul meccanismo con cui ciascun brano musicale si sviluppa ed evolve, cioè sul processo che genera le note.

Per individuare poi una misura di somiglianza nell'andamento dei dati, ciò corrisponde al secondo obiettivo che ci si pone, è logico richiedere che si utilizzino criteri indipendenti da valutazioni e giudizi soggettivi, dalle esperienze, dal gusto e dalla sensibilità di ascoltatori o critici esperti. Inoltre risulta opportuno e necessario prendere in considerazione soltanto aspetti intrinseci ai brani, tralasciando al contrario caratteristiche della musica legate alle esecuzioni, alle scelte degli interpreti o ulteriori informazioni, storiche o biografiche.

In analogia a quanto viene fatto con le tecniche di scoring nell'ambito dell'analisi testuale, si cercherà di misurare le affinità tra i brani del *corpus* e di proiettare questi, individuandone ordinamento e posizione reciproca, in uno spazio bidimensionale o tridimensionale, il tutto in modo coerente, tentando di dare infine un'interpretazione ai risultati ottenuti. Partendo con un'opportuna raccolta e codifica dei dati, in modo da recuperare da questi quanta più informazione possibile, tenendo conto della loro natura temporale e, come verrà spiegato, circolare, si procederà in una prima fase con l'adattamento di Modelli di Markov nascosti (Hidden Markov Models), nati per il riconoscimento e la sintesi vocale e sviluppati per esempio con gli studi di genomica, ed in seguito, con degli strumenti e dei metodi che permettono di confrontare distribuzioni di probabilità (la divergenza

di Kullback-Leibler) e unità statistiche (lo scaling multidimensionale, la tecnica tSNE).

Adottare una prospettiva così freddamente analitica e razionalistica, e raccogliere dei dati di tipo quantitativo per modellare aspetti legati alla creatività e all'arte potrebbe a primo impatto risultare non appropriato, troppo azzardato e al contempo limitante. Tuttavia, poiché lo scopo finale non è valutare la qualità delle opere composte, non è attribuire ad esse un significato, né ricostruire le vere regole che disciplinano la composizione, ma piuttosto cogliere affinità, tendenze e tratti comuni, accostarsi a questo problema adottando un approccio statistico è possibile, e a nostro parere anche molto interessante. Per di più, tra discipline scientifiche e musica, esiste un radicato sistema di interconnessioni. La matematica e la statistica sono state nel passato e sono tutt'oggi largamente utilizzate sia dai compositori nella creazione delle loro opere, che dagli analisti, nello studio di forme e motivi, così come, dall'altro lato, lo studio della musica fornisce spesso spunti per l'evoluzione delle teorie scientifiche. Nel Capitolo 1 ci si occupa di tracciare una breve storia del legame tra questi due campi del sapere, dei metodi quantitativi adottati nel tempo in ambito musicale, delle loro potenzialità e dei loro limiti. Con il Capitolo 2 sono introdotti i modelli di Markov e i modelli di Markov nascosti (HMM), con un'estensione al caso di sequenze di dati multidimensionali, mentre nel Capitolo 3 viene illustrato il metodo che è stato utilizzato per determinare le misure di diversità tra modelli di Markov nascosti e viene fatto un richiamo sullo scaling multidimensionale e sulla tecnica t-SNE, nel Capitolo 4 viene presentato e descritto nel dettaglio l'insieme dei dati, e riportato passo passo come è stata condotta l'analisi. Infine vengono formulate alcune considerazioni in merito alle difficoltà incontrate e suggeriti alcuni possibili sviluppi.

Capitolo 1

Musica e Statistica

1.1 Un mondo ordinato, un approccio quantitativo

L'orecchio dell'uomo classifica i suoni come musica o come rumore a seconda del fatto che identifichi in essi una forma strutturata o meno. La natura fisica di un suono è quella di un'onda sinusoidale di pressione che si propaga mediante un corpo elastico. Pertanto qualità percepite (altezza, intensità e timbro), su cui ci si basa per formulare giudizi di tipo estetico sul suono (colore, pienezza), sono in realtà quantità misurabili e corrispondono ad oscillazioni di breve o lungo periodo, regolari o irregolari, proporzionate o casuali. Sviluppi funzionali misurati e ordinati corrispondono ad effetti sonori chiari e definiti, anche detti "eufonici". Questi principi d'ordine matematici che regolano la produzione sonora e che permettono di distinguere gli eventi "musicali" da ciò che viene detto "rumore" sono propri in natura anche di altri organismi sensibili diversi dall'uomo. Alcune specie di uccelli, ad esempio, come l'usignolo eremita, noto per la sua incredibile musicalità nel canto, emettono principalmente suoni dotati di una frequenza che è un multiplo intero di una frequenza fondamentale, secondo lo stesso principio che regola gli intervalli all'interno delle scale armoniche della musica occidentale, e in parte di quella orientale.

Oggi è noto che mentre si ascolta della musica, il cervello formula in continuazione delle aspettative sugli elementi futuri della successione di suoni e sugli accostamenti possibili proprio in base ai rapporti tra le altezze. La messa a punto della previsione sarà più veloce e fine quanto più un orecchio è abituato al riconoscimento degli elementi che caratterizzano il segnale. Un orecchio allenato riuscirà a catturare bene l'altezza relativa tra suoni (la distanza in numero di semitoni intercorrenti tra essi), sia che siano riprodotti in sequenza e definiscano così un intervallo melodico, sia che si registrino in modo simultaneo e che quindi costituiscano un intervallo armonico. Le anticipazioni e le idee sull'esito

degli istanti successivi dipenderanno tuttavia anche dalla sensibilità, dalla memoria degli eventi passati di ciascuna mente oltre che da altri fattori incontrollabili. Cosicché le congetture di chi ode per la prima volta una melodia (e anche di chi per la prima volta la scrive), su come questa possa evolvere saranno difficilmente uguali tra individui diversi e in diverse circostanze. Il risultato del pensiero è non deterministico.

Ma come nasce l'interesse della scienza, della matematica alla musica? Secondo la tradizione fu Pitagora, filosofo del VI secolo a.C, il primo ad accostarsi allo studio della musica con un approccio scientifico-matematico. Egli, avendo notando che vi è una corrispondenza tra distanza in altezza tra le note (ampiezza degli intervalli musicali) e le dimensioni dei corpi mediante cui il suoni si propagano, riuscì a dimostrare quali fossero le relazioni esatte tra le due grandezze. Mostrò che dividendo una corda tesa in base a numeri interi consecutivi e mettendola in vibrazione, è possibile generare suoni armoniosi e piacevoli, consonanti. Al contrario, se i rapporti sono più complessi, le corrispondenti note risultano invece dissonanti. Per poter notare come anche altri aspetti del linguaggio musicale

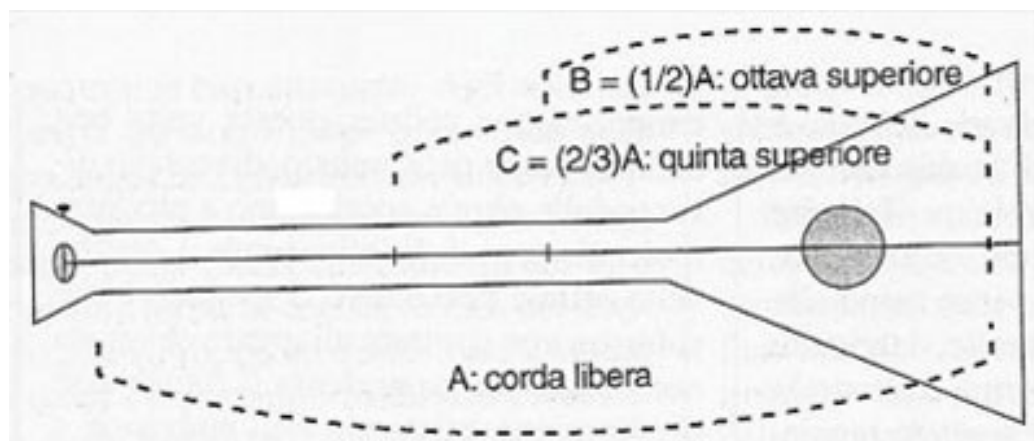


Figura 1.1: Lunghezza della corda in relazione all'altezza dei suoni.

si siano sviluppati in dipendenza da nozioni matematiche basta pensare alla divisione del ritmo e del metro musicale: ogni quantità legata alla durata dei suoni e alla scansione nel tempo è indicata con una frazione. Un'altra delle intuizioni sul parallelismo tra musica e matematica è connessa alla forma della scala musicale. L'esempio geometrico che più sembra adatto a descrivere questo tipo di organizzazione, è la figura della circonferenza: proprio come dopo i 12 semitoni di una scala si torna alla nota di partenza per ripetere la successione, così anche dopo aver percorso il perimetro di un cerchio si torna al punto iniziale. Dallo studio congiunto di musica e topologia derivano invero molti risultati che mettono in analogia alcune tipiche strutture di cui si ha riscontro in musica con le proprietà

di oggetti geometrici complessi. Uno di questi per esempio è il nastro di Moebius, con il quale si può descrivere visivamente come sono legate ciclicamente le progressioni armoniche. Le proprietà di questo oggetto sono state anticipate da alcune composizioni di Bach. È il caso del celebre *Canone Cancrizzante*, brano palindromo e per di più eseguibile contemporaneamente da due strumentisti in direzione uno opposta a quella dell'altro. È noto anche che Bach si fosse interessato alla teoria delle serie numeriche e che ne facesse uso per esempio nella forma della fuga. Altri oggetti della geometria precorsi dai compositori sono i sistemi frattali,



Figura 1.2: Incisione del Canone Cancrizzante sul nastro di Moebius

apparati che si articolano ripetendo la propria struttura su scale diverse. Soprattutto nel periodo del Classicismo a partire dall'applicazione del principio d'ordine che regola lo sviluppo di queste entità, vennero formulate e poi applicate le teorie per la composizione di brani intrinsecamente coerenti ed equilibrati a vari livelli: dalla micro alla macro struttura. Sempre in epoca illuminista, Eulero elaborò la teoria di grafi, e giunse a rappresentare il sistema delle relazioni fra le altezze in musica in due dimensioni, attraverso il *Tonnetz* (Figura 1.3). Le sue teorie vennero riprese nei secoli successivi per studiare i percorsi hamiltoniani nelle progressioni armoniche della musica di Beethoven e sono alla base delle più recenti rappresentazioni, per esempio mediante ipersfere, delle successioni accordali. A partire dalla seconda metà del '700 divennero alla moda diversi sistemi per generare musica combinando dei frammenti precomposti. Uno di questi metodi, la cui invenzione è attribuita a Mozart, prevedeva la composizione di semplici brani a partire da serie di lanci di una coppia di dadi sfruttando le leggi della probabilità. Gli

ta di floride ricerche ed interessanti sviluppi per l'analisi musicale, in parte resa possibile dalle innovazioni tecnologiche e all'impiego del calcolatore. Una delle personalità cardine dell'epoca fu il compositore e architetto greco Iannis Xenakis, che nel 1963 pubblicò *Musiques Formelles* [17], volume nel quale raccoglieva e spiegava le possibilità di utilizzo in questa disciplina di procedure stocastiche raffinate, del calcolo delle probabilità, della teoria dei giochi, introducendo in ambito musicale termini e concetti sino ad allora inediti: masse di eventi sonori, vettori, densità, grado d'ordine, tasso di trasformazione. Xenakis studiò ed utilizzò le proprietà delle catene di Markov per comporre ed analizzare dei frammenti di musica.

1.2 Caratteristiche e limiti dei dati

A questo punto ci si concentra su alcuni aspetti pratici da cui non si può prescindere nell'impostare una risposta alla domanda: "come è possibile confrontare brani musicali diverse, cosa determina uno stile?". Innanzitutto sarà necessario riuscire ad individuare le differenze e le analogie tra i brani sulla base di dati musicali, nel modo più obiettivo possibile, senza considerare aspetti per così dire estrinseci alla composizione, per esempio di tipo interpretativo-esecutivo. In musica i principali aspetti che concorrono a caratterizzare un brano e vanno considerati sono melodia, ritmo, armonia, oltre a dinamica, timbro. Alcuni di questi parametri sono legati all'organico strumentale scelto oppure a disposizione del compositore (altezza dei suoni, ritmo, dinamica, timbro, volume), altri sono molto variabili anche in sezioni diverse dello stesso brano. Il genere della rapsodia, ad esempio, prevede l'accostamento di un insieme di spunti melodici, di carattere improvvisativo e frammentario, che possono essere citazioni di altri autori, variazioni su temi popolari, e non è così immediato individuare gli elementi di unitarietà. Per cercare di ricavare delle informazioni su ciò che vorremmo chiamare 'stile', sarà determinante recuperare in qualche modo gli aspetti e le strutture per così dire latenti, in particolare l'informazione legata al modo di giustapporre e concatenare i suoni, lo sviluppo delle idee del compositore.

I dati su cui è desiderabile lavorare sono dunque le note stesse. Sarà probabilmente più efficace focalizzare l'attenzione solo sulle altezze dei suoni e sul momento in cui si registra il loro impulso. Detto ciò, sono percorribili più strade: si possono considerare le altezze come delle modalità, come dei dati categoriali oppure si può pensare di mantenere l'ordinamento naturale dei dati, che è di tipo circolare. Una terza opzione, presentata da Ian Beran nel suo volume *Statistics in Musicology*[2], consiste nell'esaminare la sequenza degli intervalli, ovvero la serie delle differenze del numero di semitoni. Questo approccio tuttavia non permette di valutare contemporaneamente più linee melodiche in parallelo (poli-

fonia). Un elemento da tenere presente è che le altezze dei suoni sono relative, e che gli intervalli svolgono una diversa funzione in base al contesto in cui si trovano. Due successioni di suoni che presentano gli stessi intervalli nell'ordine, pur se il punto di partenza non è lo stesso, producono lo stesso effetto. Se invece di considerare nello stesso brano due intervalli di uguale ampiezza, a seconda del grado di distanza del pattern dal baricentro della scala di riferimento (la tonica), questi produrranno risultati uditivamente molto diversi. Inoltre per la natura circolare dei suoni, spostandosi in avanti o all'indietro di 12 semitoni si ricade su una nota nominalmente identica a quella di partenza. Sfruttando questa idea, è possibile

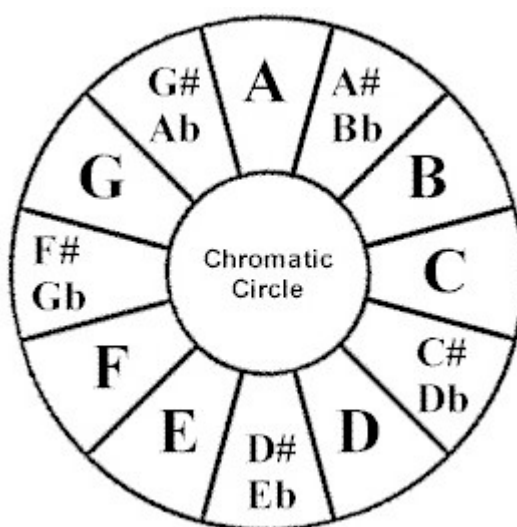


Figura 1.4: Struttura circolare del sistema notazionale

studiare le altezze delle note, anche in modulo 12, mantenendo l'importante loro corrispondenza, al netto dell'ottava in cui si presentano, (cioè quando la frequenza dell'onda sonora viene raddoppiata o dimezzata). Questo aspetto è di fondamentale importanza quando si vuole valutare il processo di formazione di un brano. Ogni suono viene perciò pensato in relazione alla sua "funzione", cioè al suo *grado*: più che l'altezza in termini numerici, si considerano come delle modalità. Il modo con cui vengono accostati i 12 semitoni diventa l'elemento chiave della caratterizzazione stilistica. Nei capitoli seguenti, si adotterà lo schema appena indicato per trattare i dati musicali.

Capitolo 2

Modelli di Markov e modelli di Markov nascosti

2.1 Catene di Markov

Nel seguente paragrafo sono riportati alcune notazioni e alcuni risultati basilari che verranno richiamati in seguito.

2.1.1 Definizioni

Processo stocastico discreto

Siano V uno spazio finito o numerabile, Ω lo spazio degli eventi elementari, Φ una misura di probabilità definita su Ω , $t \in \mathbb{N}$ e $\{C_t\}$ una successione di variabili aleatorie. $\{C_t\}$ è un processo stocastico a tempo discreto se

$$C_t : \Omega \rightarrow V, \quad t = 1, 2, 3, \dots \quad (2.1)$$

Processo markoviano del primo ordine

Si dice che $\{C_t\}$ è un processo (o catena) di Markov del primo ordine a tempo discreto e a valori finiti in V , con $|V| = M$ stati, se per ogni $t \in \mathbb{N}$ vale

$$Pr(C_{t+1} | C_t, \dots, C_1) = Pr(C_{t+1} | C_t), \quad (2.2)$$

ovvero se la probabilità al t -esimo passo condizionata alla storia dell'intero processo è equivalente alla probabilità condizionata al passo $t - 1$. In un tale processo il futuro dipende dal passato solo mediante il presente. La (2.2) è anche detta *proprietà di Markov*.

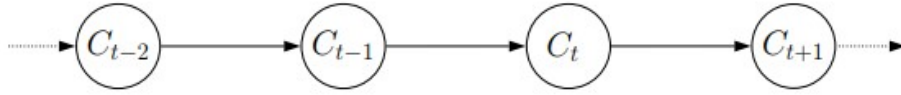


Figura 2.1: Rappresentazione di una catena di Markov mediante un grafo.

In una catena markoviana si definiscono *probabilità di transizione* le probabilità condizionate:

$$\gamma_{ij} = \Pr(C_{t+s} = j | C_t = i) \quad i, j = 1 \dots M \quad (2.3)$$

Se queste misure non dipendono da t , la catena si dice *omogenea*.

Ad ogni catena markoviana è associata una sola matrice di transizione Γ , quadrata, di dimensione $M \times M$ che ha come elementi le probabilità di transizione γ_{ij} associate alle modalità i, j .

$$\Gamma = \begin{pmatrix} \gamma_{11} & \dots & \gamma_{1M} \\ \dots & \gamma_{ij} & \dots \\ \gamma_{M1} & \dots & \gamma_{MM} \end{pmatrix}$$

La matrice Γ soddisfa la seguente proprietà:

$$\sum_{j=1}^M \gamma_{ij} = 1, \forall i \in \{1, \dots, M\} \quad (2.4)$$

Un'altra proprietà fondamentale delle catene markoviane a stati finiti omogenee è che esse soddisfano l'equazione di Chapman–Kolmogorov:

$$\Gamma(t + u) = \Gamma(t)\Gamma(u) \quad (2.5)$$

Stazionarietà

Sia δ un vettore riga con elementi non negativi, se $\delta\Gamma = \delta$ e $\delta\mathbf{1} = 1$, allora δ è associato a una distribuzione stazionaria del processo con matrice di transizione Γ . Una catena markoviana omogenea, a tempo discreto, con spazio S degli stati finito, ammette un'unica distribuzione stazionaria strettamente positiva, cioè, dati $\delta > 0$ e U vettore della distribuzione d'inizio processo, allora

$$\delta \text{ è stazionario} \Leftrightarrow \delta (I_M - \Gamma + U) = 1$$

In un processo markoviano stazionario, la probabilità di realizzazione di ogni stato è costante nel tempo e uguale alla probabilità al tempo 1.

2.1.2 Inferenza

Data una realizzazione di una catena di Markov, un modo per stimare la matrice di transizione è calcolare il valore degli elementi γ_{ij} come rapporto tra numero di transizioni dallo stato i allo stato j osservate nel processo, che denotiamo con f_{ij} , e numero di stati i osservati, f_{i+} . Di seguito viene mostrato che questa operazione coincide con la stima di massima verosimiglianza di Γ , condizionatamente alla prima osservazione.

Stima di Massima Verosimiglianza

Supponiamo di voler stimare il valore degli $M(M-1)$ parametri della matrice di transizione (poiché la somma per riga degli elementi della matrice è uguale a 1, imponiamo M vincoli), di una catena di Markov a M stati, a partire da una sua realizzazione C_1, C_2, \dots, C_T . La verosimiglianza condizionata alla prima osservazione è pari a

$$\mathcal{L} = \prod_{i=1}^M \prod_{j=1}^M \gamma_{ij}^{f_{ij}}, \quad (2.6)$$

da cui, applicando il logaritmo si ottiene la log-verosimiglianza nella forma

$$\ell = \sum_{i=1}^M \left(\sum_{j=1}^M f_{ij} \log \gamma_{ij} \right) = \sum_{i=1}^M l_i \quad \forall i \in \{1 \dots M\}. \quad (2.7)$$

È possibile massimizzare la log-verosimiglianza (ℓ) massimizzando separatamente ciascun contributo alla sommatoria (l_i). Derivando infine ciascun membro rispetto a γ_{ij} per $j \in \{1, \dots, M\}$, e uguagliando il risultato a zero avendo riscritto $\gamma_{ii} = 1 - \sum_{h \neq i} \gamma_{ih}$,

$$\gamma = \sum_{i=1}^M \left(\sum_{j=1}^M f_{ij} \log \gamma_{ij} \right) = \sum_{i=1}^M l_i \quad \forall i \in \{1 \dots M\} \quad (2.8)$$

si ottiene

$$0 = \frac{-f_{ii}}{1 - \sum_{h \neq i} \gamma_{ih}} + \frac{f_{ij}}{\gamma_{ij}} = -\frac{f_{ii}}{\gamma_{ii}} + \frac{f_{ij}}{\gamma_{ij}} \quad (2.9)$$

Perciò, a patto che f_{i+} sia diverso da zero, si ottiene un massimo per la funzione di verosimiglianza in corrispondenza di $\gamma_{ij} = f_{ij}/f_{i+}$ (per $i, j \in \{1, \dots, M\}$) e $\gamma_{ii} = \sum_j 1 - \gamma_{ij}$ (per $i, j \in \{1, \dots, M\}$ e $i \neq j$), che corrispondono esattamente alle stime di transizione empiriche. Si noti che non è stata usata l'ipotesi di stazionarietà per derivare questo risultato. Assumendo la stazionarietà, usare la verosimiglianza condizionata equivale ad usare la verosimiglianza esatta.

2.1.3 Processi Markoviani di ordini superiore

Talvolta anche condizionatamente all'evento pregresso, non è soddisfatta l'assunzione di indipendenza tra una variabile C_t e il passato del processo. È possibile in questi casi estendere la (2.2) definendo un processo di Markov di ordine l , con $l \geq 2$, che soddisfa la seguente proprietà:

$$Pr(C_t|C_{t-1}, C_{t-2}, \dots) = Pr(C_t|C_{t-1}, \dots, C_{t-l}). \quad (2.10)$$

Per esempio, un processo markoviano del second'ordine, se stazionario, è caratterizzato dalle probabilità di transizione

$$\gamma_{ijk} = Pr(C_t = k|C_{t-1} = i, C_{t-2} = j). \quad (2.11)$$

La matrice di transizione viene sostituita da un tensore di transizione di dimensioni M^3

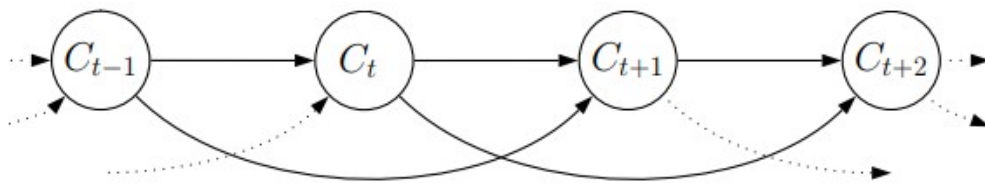


Figura 2.2: Grafo di un processo markoviano di secondo ordine

2.2 Modelli markoviani nascosti

Un modello di Markov nascosto (HMM) è un particolare tipo di mistura: si compone di un processo $\{C_t\}$ discreto, che evolve secondo una legge di Markov, i cui stati non sono osservabili direttamente, e di una sequenza (o più d'una), $\{X_t\}$, che è una funzione probabilistica degli stati nascosti dalla catena.

2.2.1 Notazione

Si consideri il caso in cui la sequenza univariata $\{X_t\}$ ha supporto discreto e finito. Un HMM discreto del primo ordine, è caratterizzato dai seguenti elementi:

- la sequenza di stati osservati $\{X_t\}$, detta anche sequenza dipendente, con generico stato osservato $s \in \{1, \dots, S\}$.

- la sequenza di stati nascosti o latenti $\{C_t\}$, con generico elemento $m \in \{1, \dots, M\}$.
- il vettore delle probabilità iniziali δ di lunghezza M , con generico elemento δ_m che corrisponde alla probabilità di osservare al tempo 1 lo stato m esimo del processo latente:

$$\delta_m = Pr(C_1 = m), m \in \{1, \dots, M\}.$$

- la matrice di transizione $\Gamma = \{\gamma_{ij}\}$ di dimensioni M^2 , in cui l'elemento γ_{ij} rappresenta la probabilità di passare dall' i esimo o stato nascosto al tempo $t - 1$ al j esimo stato nascosto al tempo t :

$$\gamma_{ij} = Pr(C_t = i | C_{t-1} = j), i, j \in \{1, \dots, M\}.$$

- la matrice di emissione $\Lambda = \{\lambda_{is}\}$ di dimensioni $M \times S$, in cui l'elemento λ_{is} rappresenta la probabilità associata allo stato nascosto i esimo di generare lo stato osservato s -esimo

$$\lambda_{is} = Pr(X_t = s | C_t = i), i \in \{1, \dots, M\}, s \in \{1, \dots, S\}.$$

Pertanto il modello si può indicare anche in notazione compatta

$$\Psi = \{\Theta, C_t, X_t\},$$

dove

$$\Theta = \{\delta, \Gamma, \Lambda\}.$$

Si considerino d'ora in avanti solo modelli di Markov nascosti omogenei, in cui le probabilità di transizione rimangono invariate nel tempo. Si assume che la sequenza C_t segua un processo markoviano del primo ordine e che le osservazioni X_t dipendano unicamente dallo stato nascosto al medesimo istante. Denotando con $C^{(t-1)}$, l'insieme informativo relativo all'intera storia del processo nascosto, e con $X^{(t-1)}$ quello relativo alla storia del processo osservato, il modello si può scrivere come:

$$Pr(C_t | C^{(t-1)}) = Pr(C_t | C_{t-1}), t \in \{2, 3, \dots, T\} \quad (2.12)$$

$$Pr(X_t | X^{(t-1)}, C^{(t)}) = Pr(X_t | C_t), t \in \{1, \dots, T\} \quad (2.13)$$

Questa struttura può essere rappresentata anche come nel grafo riportato di seguito. Se la catena markoviana latente $\{C_t\}$ ha M stati, allora si dice che $\Psi = \{\Theta, C_t, X_t\}$, è un HMM a M stati.

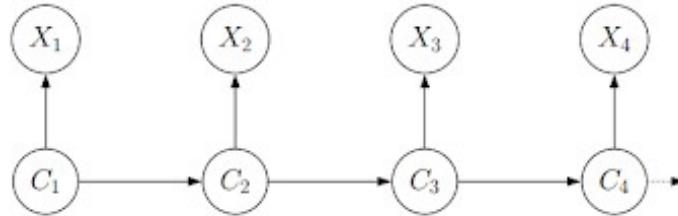


Figura 2.3: Grafo di un HMM

2.3 Problemi e inferenza

Data una realizzazione di un HMM, le domande a cui si può voler trovare risposta sono:

- identificare i parametri $\Theta = \{\delta, \Gamma, \Lambda\}$ per cui è massima la probabilità di aver generato la serie osservata (problema di stima).
- trovare la sequenza di stati nascosti che massimizza la probabilità di aver dato origine ai dati, quando è noto il modello (problema di decodifica).
- dato un modello $\Psi = \{\Theta, C_t, X_t\}$, calcolare la probabilità che si verifichi la sequenza data (problema di valutazione).

Nelle pagine a seguire ci si occuperà più nel dettaglio solo di alcuni strumenti per risolvere il problema di stima.

Esistono più metodi e algoritmi per stimare i parametri del modello. Uno di questi è la stima di massima verosimiglianza, o massimizzazione diretta numerica.

2.3.1 Stima di massima verosimiglianza

Supponendo che il modello generatore dei dati sia omogeneo e del primo ordine, è possibile scrivere la distribuzione delle osservazioni come

$$\sum_{i=1}^M \Pr(C_t = i) \Pr(X_t = s | C_t = i) = \sum_{i=1}^M u_i(t) \lambda_{is} \quad (2.14)$$

In cui $u_i(t)$ è il vettore riga contenente il valore della probabilità associata allo stato nascosto i esimo al tempo t . La scrittura della distribuzione è possibile

anche per processi di ordine più elevato, purché omogenei. Inoltre può essere equivalentemente riscritta in forma matriciale

$$\begin{aligned} \Pr(X_t = s) &= \mathbf{u}(1)\Lambda\mathbf{1}' \begin{pmatrix} \lambda_{11} & \dots & \lambda_{S1} \\ \vdots & \lambda_{is} & \vdots \\ \lambda_{M1} & \dots & \lambda_{MS} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \\ &= \mathbf{u}\Lambda\mathbf{1}' \end{aligned} \quad (2.15)$$

Si può verificare che

$$\mathbf{u}(t) = \mathbf{u}(1)\Gamma^{t-1}. \quad (2.16)$$

Perciò, se una catena markoviana è omogenea, vale che

$$\mathbf{u}(t) = \mathbf{u}(1)\Gamma^{t-1}\Lambda\mathbf{1}'. \quad (2.17)$$

In condizioni di stazionarietà, con δ vettore dei valori di probabilità associato alla distribuzione stazionaria, allora il risultato si semplifica in $\delta\Gamma^{t-1} = \delta$, per ogni valore di t , e perciò

$$\Pr(X_t = s) = \delta\Lambda\mathbf{1}'. \quad (2.18)$$

La verosimiglianza dell'intera sequenza è data da

$$L_T = \Pr(X^{(T)} = x^{(t)}) = \sum_{c_1, c_2, \dots, c_{T-1}}^M \Pr(X^{(T)} = x^{(T)}, C^{(T)} = c^{(T)}),$$

dove ciascuno degli addendi del secondo termine è calcolato come

$$\Pr(X^{(T)}, C^{(T)}) = \Pr(C_1) \prod_{t=1}^T \Pr(C_t | C_{t-1}) \prod_{t=1}^T \Pr(X_t | C_t). \quad (2.19)$$

Segue che

$$L_T = \sum_{i=1}^M \left(\delta \lambda_{1i} \prod_{t=2}^T \gamma_{t-1i} \lambda_{ti} \right)$$

con λ_{ti} uguale alla probabilità che lo stato nascosto i -esimo generi (al tempo t) lo stato osservato.

$$L_T = \delta\Lambda(x_1)\Gamma\Lambda(x_2)\Gamma\Lambda(x_3) \cdots \Gamma\Lambda(x_T)\mathbf{1}' \quad (2.20)$$

Nella procedura di massimizzazione diretta della verosimiglianza emergono alcune complicazioni, i problemi coinvolti sono :

1. underflow;
2. vincoli sui parametri;
3. esistenza di molti massimi locali per la funzione di verosimiglianza;
4. presenza di dati mancanti.

Underflow

Uno dei principali problemi di natura computazionale nelle procedure di stima è dato dal fatto che la verosimiglianza, nel caso di distribuzioni dipendenti discrete, in quanto prodotto di probabilità, e quindi di fattori inferiori a 1, tende progressivamente a zero con l'aumentare del numero di osservazioni della sequenza. Una via percorribile è data dall'applicazione di opportune riparametrizzazioni delle grandezze in gioco, per permettere agli algoritmi di arrivare ad una soluzione.

Vincoli

Vi sono due tipologie di vincoli da considerare:

- i parametri della matrice Γ e della matrice Λ devono essere non negativi.
- la somma per riga dei parametri della matrice di transizione deve essere pari a 1.

Per soddisfare le due condizioni, anche in questo caso viene applicata una serie di riparametrizzazioni.

Per Λ si consideri una matrice $H = \{\eta_{is}\}$, con $\eta_{is} = \log \lambda_{is}$, da cui $\lambda_{is} = e^{\eta_{is}}$.

Per Γ si considerino:

- una matrice $T = \{\tau_{ij}\}$, $\tau_{ij} \in \mathbb{R}$, $i, j \in \{1, \dots, M\}$;
- una funzione g strettamente crescente e positiva, $g(x) = \exp x$;
- una matrice $P = \{\rho_{ij}\}$, $\rho_{ij} = \begin{cases} g(\tau_{ij}) & i \neq j \\ 1 & i = j \end{cases}$.
- la relazione $\gamma_{ij} = \rho_{ij} / \sum_k \rho_{ik}$, da cui $\tau_{ij} = \log(\gamma_{ij} / (1 - \sum_j \gamma_{ij}))$ dove η_{is} e τ_{ij} sono detti *parametri di lavoro*, mentre λ_{is} e γ_{ij} sono detti *parametri naturali*.

Per la proprietà di invarianza della funzione di verosimiglianza, è possibile massimizzare L_T rispetto a η_{is} e τ_{ij} , che variano nello spazio \mathbb{R}^2 e ricondursi in seguito alla parametrizzazione originale in λ_{is} e γ_{ij} .

Massimi locali

Naturalmente, si richiede di fornire all'algoritmo di ottimizzazione numerica dei buoni valori iniziali per i parametri da stimare. Questo permette di ottenere la soluzione ottima in un tempo ragionevole. Inoltre è consigliabile eseguire l'algoritmo con diversi valori di partenza, per poter escludere che le stime trovate rappresentino solo dei massimi locali per la verosimiglianza.

Dati mancanti

In ambito di serie temporali, è facile imbattersi in sequenze ove siano presenti anche molti dati mancanti. Con modelli markoviani, tuttavia, gli aggiustamenti da apportare alla funzione di verosimiglianza in questi casi non sono affatto complessi. All'interno dell'espressione, in corrispondenza delle osservazioni mancanti vengono imputate in vece di Γ , delle matrici I_M , e analogamente degli 1 per ciascuna delle probabilità di emissione della serie dipendente.

La soluzione delle equazioni di stima va ottenuta per via numerica. Per ulteriori approfondimenti sugli algoritmi di ottimizzazione si vedano Altman e Petkau (2005) [12] e Turner (2008)[14].

2.3.2 Scelta del modello

A volte, quando il numero di stati nascosti non è dato a priori, ci si trova nella situazione di voler confrontare anche modelli con numero di parametri diverso. Al crescere di M , cioè del numero di stati nascosti, corrisponde in genere un miglior adattamento del modello in termini di verosimiglianza. Ma accanto a un numero maggiore di stati non osservabili si ha anche una crescita non lineare, ma quadratica del numero di parametri. Occorre un criterio di preferenza che pesi opportunamente la bontà dell'adattamento con la complessità del modello. L'idea alla base è quella di utilizzare una misura di discrepanza tra il modello *vero*, Ψ , e il modello stimato, Ξ , che viene indicata con $\Delta(\Psi, \Xi)$, e che quantifica la perdita in termini di *efficacia predittiva*. Di fatto però, il modello Ψ è ignoto, cosicché la scelta si basa sulla minimizzazione della devianza atteso $E_{\Psi}(\Delta(\Psi, \Xi))$. Secondo questo presupposto, un modello sovrapparametrizzato sarà meno efficace rispetto ad un modello parsimonioso, purché correttamente specificato e sotto condizioni di regolarità, minimizzare $E_{\Psi}(\Delta(\Psi, \Xi))$ equivale a minimizzare il Criterio di Akaike (AIC) $-2 \log L + 2p$, in cui $\log L$ è la log-verosimiglianza del modello adattato e p ne indica il numero di parametri. Il primo termine misura l'adattamento e decresce con l'introduzione di nuovi parametri (stati), mentre il secondo funge da penalità e aumenta in corrispondenza di modelli più complessi. Poiché si dimostra che, secondo questo criterio, la probabilità di escludere un modello sovrapparametrizzato non si annulla al divergere di n , in alternativa, la selezione può essere più convenientemente condotta cercando di minimizzare il criterio di informazione Bayesiano (BIC), diverso dal criterio AIC nella misura della penalità e che è invece consistente: $-2 \log L + p \log T$, in cui T è sempre il numero di osservazioni nel processo.

2.4 HMM per sequenze multivariate

La definizione di HMM può essere estesa anche al caso multidimensionale, in cui ad ogni passo del processo viene generato un vettore di osservazioni anziché un singolo elemento.

Nel caso di sequenze multivariate, ogni osservazione è indicata nella forma x_{kt} , $k=1, \dots, K$, $t=1, \dots, T$.

L'indice t fa riferimento agli istanti della successione, mentre l'indice n segue la posizione degli elementi nel vettore generato ad ogni istante, ovvero le sequenze univariate simultanee.

Per questi modelli le assunzioni che vengono fatte sono sia di indipendenza condizionata longitudinale (tra gli elementi della stessa sequenza osservata), dato lo stato nascosto, che indipendenza condizionata parallela (tra gli elementi delle distinte sequenze). In più si considera che vi sia la stessa struttura latente per ciascuna delle sequenze parallele, cosicché il modello ha una sola matrice di transizione, ma diverse matrici di emissione (K).

$$\Pr(X_{1t}, \dots, X_{Kt}) = \Pr(X_{1t}|C_t) \dots \Pr(X_{Kt}|C_t) \quad (2.21)$$

$$\Pr(X_{kt}, X_{ks}) = \Pr(X_{kt}|C_t) \Pr(X_{ks}|C_t) \quad (2.22)$$

Indicando con Θ l'insieme dei parametri del modello, con X l'insieme delle osservazioni, ovvero $\Theta=(\delta, \Gamma, \Lambda_1, \dots, \Lambda_K)$, $X=(x_{11}, \dots, x_{tk}, \dots, x_{TK})$, la verosimiglianza può essere scritta come:

$$\begin{aligned} L_T(\Theta|X) &= \sum_{i=1}^M \Pr(X|C_i, \Theta) \Pr(C_i|\Theta) \\ &= \sum_{i=1}^M \left(\Pr(C_1|\Theta) \Pr(X_1|C_1, \Theta) \prod_{t=2}^T \Pr(C_t|C_{t-1}, \Theta) \Pr(X_t|C_t, \Theta) \right) \quad (2.23) \\ &= \sum_{i=1}^M \left(\delta \lambda_{1i1} \dots \lambda_{1iK} \prod_{t=2}^T \gamma_{t-1i} \lambda_{ti1} \dots \lambda_{tiK} \right), \end{aligned}$$

con λ_{tik} probabilità che lo stato nascosto i -esimo generi (al tempo t) lo stato osservato all'interno della sequenza n -esima. In alcuni casi l'assunzione di indipendenza non è del tutto plausibile. Se lo scopo dell'analisi è di fare delle previsioni o simulare nuove sequenze è necessario verificare con cautela l'assunzione; se tuttavia si desidera descrivere delle strutture e ricavare informazione "sintetica", l'indipendenza viene accettata anche se non è del tutto ragionevole in teoria. Con modelli multivariati si considerano in genere un numero M di stati nascosti ridotto, per ottenere modelli più parsimoniosi e garantire perciò una maggiore efficienza

computazionale nella stima accanto ad una procedura inferenziale più semplice. L'inferenza sugli stati nascosti attraverso un approccio multivariato come quello descritto (rispetto per esempio alla formulazione di un modello con diverse catene nascoste) risulta particolarmente utile se alcune delle sequenze parallele sono solo parzialmente osservate, ovvero in presenza di molti dati mancanti.

2.5 Software

Il pacchetto di R `seqHMM`, nasce per trattare agevolmente modelli di Markov nascosti e misture di HMM per dati categoriali, anche nel caso di sequenze multidimensionali. Sono implementate delle funzioni efficienti, che in alcuni casi riescono a sfruttare le potenzialità del calcolo parallelo. Sono implementati diversi metodi per ottenere le stime dei parametri del modello (la massima verosimiglianza, ma anche il metodo E-M, qui non trattato, e la massimizzazione locale [9]).

18CAPITOLO 2. MODELLI DI MARKOV E MODELLI DI MARKOV NASCOSTI

Capitolo 3

Confrontare gli HMM

Dati due o più HMM, possiamo essere interessati a volerli mettere a confronto: non solo per decidere quale sia il migliore sullo stesso insieme di dati, in termini di bontà di adattamento, come accennato alla fine del Capitolo 2, ma anche per valutarne il grado di somiglianza, partendo da più insiemi di dati. In questo Capitolo si discute un modo per calcolare una misura di similarità per HMM.

3.1 Il caso multivariato

Nel caso di HMM ottenuti a partire da diversi insiemi di dati, e da diversi processi generatori, ci si trova tipicamente con un numero di stati nascosto non costante, poiché per ogni campione si è cercato di ottenere la stima dei parametri del modello teorico per la quale è massima la probabilità di aver generato le osservazioni. Inoltre, nel caso di dati multivariati, anche in condizioni di uguaglianza nel numero di stati nascosti e a parità di lunghezza della serie, si avranno un numero variabile di osservazioni, multiplo del numero di sequenze parallele. Ciò implica che per ogni modello avremo un numero di parametri diverso: matrici di transizione di dimensione variabile ($M \times M$, con M intero generico) e matrici di emissione diverse nel numero di righe $M \times S$ e in numero non costante (K , dove K corrisponde alla dimensione trasversale della serie, ovvero a quante sequenze parallele abbiamo osservato). Per questo motivo non è possibile calcolare una distanza mettendo in relazione diretta le matrici di transizione o quelle di emissione.

Si noti che sussiste un ulteriore problema, legato alla natura dei dati relativi all'altezza delle note. Infatti, è stato già anticipato che quasi tutti i brani musicali sono costruiti a partire da una specifica tonalità (o tono) e le altezze dei suoni hanno una funzione “relativa”, che dipende dalla loro posizione nel sistema di riferimento i cui sono inserite. D'altra parte la scelta della tonalità è legata alla tessitura e alle potenzialità degli strumenti utilizzati, può avere una valenza sim-

bolica, può essere casuale, o ancora può essere il risultato di un'operazione di trasposizione di un brano pensato originariamente in altro tono d'impianto: qualunque sia la motivazione che spinge un compositore ad adottare un particolare punto di partenza per il suo pezzo, questa non ha in genere a che vedere con il processo secondo cui il brano potrà evolvere. La tonalità perciò è un elemento che ai fini di quest'analisi ha valenza di confondente, sarà necessario "depurare" i dati rispetto a questa variabile per rendere i brani confrontabili. Poiché il sistema tonale è circolare, questa operazione corrisponde dal punto di vista geometrico, all'applicazione di una rotazione alle matrici di emissione, le matrici di transizione, invece rimarrebbero invariate. Equivalentemente, per eliminare in partenza, prima di aver stimato i modelli, l'effetto di confondimento è possibile trasportare alla medesima tonalità e quindi ruotare i dati relativi alle altezze codificate da 0 a 11, oppure operare sulle altezze assolute una semplice traslazione. Un altro

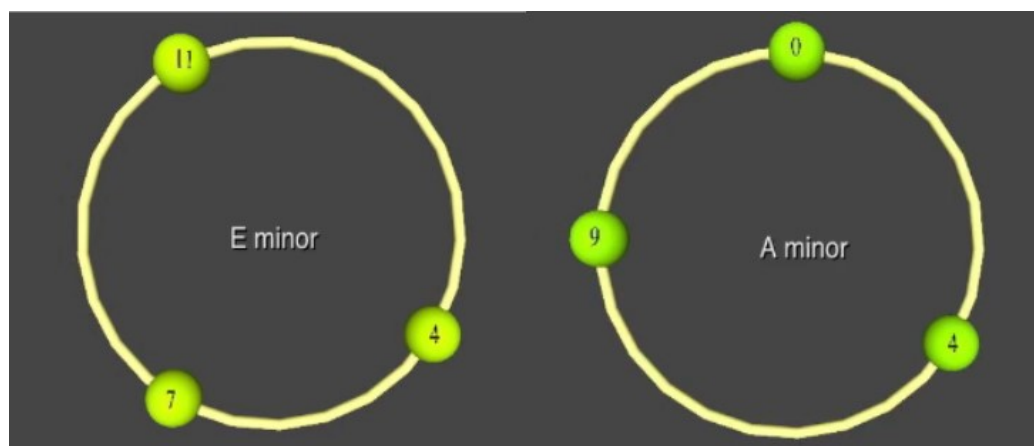


Figura 3.1: Rotazione di $5/6\pi$ (5 semitoni) e passaggio dall'accordo di *mi minore* all'accordo di *la minore*

modo di pensare alla similarità è calcolare la probabilità di osservare una stessa sequenza per due modelli dati. La sequenza deve essere riconducibile agli stati osservati e non a quelli nascosti, altrimenti la probabilità associata ad una sequenza con M possibili stati, per un modello con un numero s di stati stimato inferiore a M sarebbe sempre uguale a zero. Andrebbe quindi stabilito quale sequenza utilizzare o a quale modello ricorrere per generarla e anche in questo caso l'operazione diventerebbe complicata se le sequenze osservate fossero multivariate e i modelli da confrontare avessero dimensioni diverse.

3.2 Divergenza di Kullback-Leibler

Partendo dal presupposto di utilizzare una distanza in termini probabilistici, B.H. Juang e L. R. Rabiner in un lavoro del 1984,[10], presentarono una tecnica per il calcolo della similarità che consiste nella ristima di HMM e nel calcolo della quantità

$$\Delta(\psi, \xi) = K(\psi, \xi) - K(\psi, \xi) \quad (3.1)$$

$$= \lim_{T \rightarrow \infty} \frac{1}{T} [\log \Pr(X_{\psi'}^{(T)} | \xi) - \log \Pr(X_{\psi'}^{(T)} | \psi)],$$

dove $X_{\psi'}^{(T)}$ indica l'insieme delle possibili realizzazioni del processo di lunghezza T dal vero modello ψ . Questa quantità è legata, a meno di un fattore moltiplicativo positivo, all'entropia relativa, o divergenza di Kullback-Leibler, che per distribuzioni discrete ψ e ξ è definita come

$$D_{KL}(\psi, \xi) = \sum \Pr(X_{\psi'}^{(T)} | \psi) (\log \Pr(X_{\psi'}^{(T)} | \psi) - \log \Pr(X_{\psi'}^{(T)} | \xi)), \quad (3.2)$$

Semplificando la scrittura di $\Pr(X_{\psi'}^{(T)} | \psi)$ e $\log \Pr(X_{\psi'}^{(T)} | \xi)$ in ψ e ξ , la (3.2) gode delle seguenti proprietà:

1. $D(\psi, \xi) \geq 0$
2. $D(\psi, \xi) = 0 \leftrightarrow \xi = \psi$
3. $D(\psi, \xi) \neq D(\xi, \psi)$, ovvero assenza di simmetria

Dimostrazione della proprietà 1:

$$\begin{aligned} -D_{KL}(\psi, \xi) &= - \sum \psi (\log(\psi) - \log(\xi)) \\ &= \sum \psi (\log(\xi) - \log(\psi)) \end{aligned}$$

Per la disuguaglianza di Jensen

$$\begin{aligned} &= \sum \psi (\log(\xi) - \log(\psi)) \leq \log(\sum \psi (\xi - \psi)) \\ &= \log(\sum \xi) \leq \log 1 = 0. \end{aligned}$$

Perciò

$$-D_{KL}(\psi, \xi) \leq 0$$

e

$$D_{KL}(\psi, \xi) \geq 0$$

. Inoltre, la (3.2) si può riscrivere nella forma

$$\begin{aligned} \sum p(\log(p) - \log(q)) &= \sum \psi(\log(\psi)) - \sum \psi(\log(\xi)) \\ &= -H(\psi, \xi) - E_\psi H(\xi) \end{aligned}$$

, dove $H(\psi)$ sta ad indicare l'entropia, mentre $-H(\psi, \xi)$ è la cross-entropia. Pertanto la divergenza rappresenta l'incremento atteso in entropia risultante dall'uso di una approssimazione della distribuzione di ψ con ξ . Poiché la divergenza, come la (3.2) è non simmetrica, ma sempre non negativa, Juang e L. R. Rabiner, proposero nello stesso lavoro dell'84 la forma

$$\Delta(\psi, \xi)^* = \frac{1}{2}(\Delta(\psi, \xi) + \Delta(\xi, \psi)),$$

che può essere interpretata come una misura di difficoltà (o facilità) nel discriminare il modello ψ dal modello ξ o equivalentemente ξ da ψ . Nello studio degli HMM per sequenze multivariate, utilizzando la procedura proposta dagli autori, che prevede la ristima sotto la distribuzione ξ dei modelli generati da ψ , la varietà delle successioni non è la stessa, mentre la stima di $\Pr(X_{\psi'}^{(T)}|\xi)$ è possibile solo in presenza di un numero omogeneo di matrici di emissione. Le soluzioni possibili sono o generare i nuovi dati con il numero minimo di sequenze parallele dei modelli ψ e ξ , oppure utilizzare tutte le matrici di emissione a disposizione per generare le nuove sequenze e ristimare le $\Pr(X_{\psi'}^{(T)}|\xi)$ utilizzando a rotazione le matrici del modello ξ . Nel Capitolo 4 è stata utilizzata quest'ultima modalità di procedere. Per approfondire si veda inoltre [7].

3.3 Mappare i dati

3.3.1 Scaling Multidimensionale

Scopo dello scaling multidimensionale è riuscire a collocare un insieme di N unità in uno spazio di poche dimensioni (D), tentando di mantenere il più possibile invariata la struttura delle relazioni tra i soggetti. Le misure di distanza sono in genere simmetriche, soddisfano cioè la proprietà

$$d(X_1, X_2) = d(X_2, X_1).$$

e possono essere sia distanze metriche (di tipo spaziale, differenze quantitative relative a caratteristiche fisiche), che non metriche, (giudizi, opinioni e punteggi). I valori sono riassunti in una matrice (simmetrica) di dimensioni $N \times N$. A partire da questa, la collocazione delle unità non può essere univocamente determinata ma per lo stesso insieme di misure osservate si hanno infinite soluzioni che sono traslazioni e rotazioni della configurazione di partenza.

Scaling metrico

Per ogni coppia di unità statistiche si osserva una misura di dissimilarità o di distanza. Questa quantità è supposta essere legata proprio alla distanza tra due corrispondenti punti in uno spazio euclideo: lo scaling metrico consente di individuare le coordinate. Si ipotizza che lo spazio originale, da cui derivano le quantità note, sia uno spazio D -dimensionale. Sia $\mathbf{X}=N \times D$ l'insieme delle osservazioni, $x_i=1 \times D$ il vettore delle coordinate per l' i esima unità statistica. Si consideri matrice $\mathbf{B}=N \times N$, tale che

$$\mathbf{B} = \mathbf{X}\mathbf{X}^\top \quad (3.3)$$

il cui generico elemento si ottiene da

$$b_{ij} = \sum_{d=1}^D x_{id}x_{jd}. \quad (3.4)$$

Si consideri inoltre la matrice delle distanze $\mathbf{D}=N \times N$, con generico elemento d_{ij} ; si verifica facilmente che

$$d_{ij}^2 = (x_i - x_j)^2 = b_{ii} + b_{jj} - 2b_{ij}. \quad (3.5)$$

Pertanto, scrivendo la matrice \mathbf{D} , nelle tre componenti per cui vale l'espressione (3.5), fattorizzando poi \mathbf{B} nel prodotto in (3.3), e ponendo dei vincoli per l'identificabilità, è possibile risalire agli elementi di \mathbf{X} . Viene posto \bar{x} all'origine, $\sum_{i=1}^N x_{id}=0$, per $m \in \{1, \dots, D\}$. Segue che la somma dei termini di ogni riga (e di ogni colonna) di \mathbf{B} è uguale a 0. Sommando in i e in j gli elementi in (3.5), si ottengono quindi

$$\sum_{i=1}^N d_{ij}^2 = T + Nb_{jj} \quad (3.6)$$

$$\sum_{j=1}^N d_{ij}^2 = T + Nb_{ii} \quad (3.7)$$

$$\sum_{i=1}^N \sum_{j=1}^N d_{ij}^2 = 2NT \quad (3.8)$$

dove $T = \sum_{i=1}^N b_{ii}$ è la traccia della matrice \mathbf{B} . In base a questo risultato si ha che

$$b_{ij} = -\frac{1}{2}(d_{ij}^2 - d_{i.}^2 - d_{.j}^2 + d_{..}^2) \quad (3.9)$$

con

$$d_{i.}^2 = \frac{1}{N} \sum_{j=1}^N d_{ij}^2,$$

$$d_{.j}^2 = \frac{1}{N} \sum_{i=1}^N d_{ij}^2,$$

$$d_{..}^2 = \frac{1}{N^2} \sum_{j=1}^N d_{ij}^2.$$

Pertanto, individuati gli elementi b_{ij} come d_{ij}^2 , non resta che fattorizzare \mathbf{B} . Usando la scomposizione spettrale, si ha

$$\mathbf{B} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top.$$

$\mathbf{\Lambda}$ è la matrice diagonale degli autovalori di \mathbf{B} , mentre \mathbf{V} è la matrice che contiene nelle colonne gli autovettori di \mathbf{B} . Supponendo pari ad M la dimensione dello spazio delle colonne di \mathbf{X} , gli autovalori $\lambda_{D+1}, \dots, \lambda_N$ vengono fissati pari a zero. Il modello di rappresentazione D -dimensionale può essere considerato adeguato se il valore GOF (Goodness Of Fit), così definito

$$GOF = \frac{\sum_{i=1}^D \lambda_i}{\sum_{i=1}^N \lambda_i} \quad (3.10)$$

risulta maggiore o uguale a 0.8 [6].

3.3.2 L'algoritmo t-SNE

T-SNE (t- distributed Stochastic Neighbor Embedding) è una tecnica che consente di produrre una rappresentazione in poche dimensioni (D) della struttura di un insieme di dati, per ognuno dei quali si ha un gran numero (L) di record. I contesti in cui risulta vincente e più viene utilizzata, sono legati alla visualizzazione delle immagini. Le unità statistiche nello spazio originale sono indicate con x_i , $i=1 \dots N$, $x_i=(x_{i1}, x_{i2}, \dots, x_{iL})$, mentre nello spazio di arrivo sono indicate con y_i , $i=1 \dots N$, $y_i=(y_{i1}, \dots, y_{iD})$, $D \ll L$. Per ciascuna unità statistica viene stimata una distribuzione normale L variata, dopodiché, per ogni coppia di osservazioni x_i, x_j vengono calcolate le quantità

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}, \quad (3.11)$$

dove $p_{i|j}$ è la densità delle distribuzione centrata in x_j calcolata in x_i , e rappresentano ciascuna la probabilità che, dato x_j , il punto ad esso più vicino sia x_i .

$$p_{i|j} = \frac{\exp(-\|x_i - x_j\|_2)}{\sum_{h \neq i} \exp(-\|x_i - x_h\|_2)}$$

e viceversa per $p_{j|i}$.

Le quantità analoghe nello spazio di arrivo sono calcolate per ciascuna coppia y_i, y_j direttamente con la densità della distribuzione congiunta, supposta essere una t di Student M-variata, con un grado di libertà:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|_2)^{-1}}{\sum_{h \neq i} (1 + \|y_h - y_i\|_2)^{-1}}. \quad (3.12)$$

Le coordinate delle osservazioni y_i $i \in \{1 \dots N\}$ nello spazio D -dimensionale vengono trovate minimizzando la funzione costo

$$C^* = \sum_i KL(P_i \parallel Q_i) = \sum_i \sum_j p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right)$$

ovvero minimizzando la somma totale delle Divergenze di Kullback-Leibler tra le distribuzioni L e D -dimensionali calcolate per ogni coppia. La divisione per $2N$ nella formula 3.11 implica che ciascun punto abbia un peso significativo all'interno della funzione di costo ($\sum_j p_{ij} > \frac{1}{N}, \forall i$), mentre l'assunzione sulla forma della distribuzione permette, a pari valore per la densità congiunta delle osservazioni i e j , rispetto ad una Gaussiana, di ottenere delle unità più distanziate, più sgranate, sul supporto D -dimensionale. Per i dettagli sul funzionamento dell'algoritmo di minimizzazione della funzione C^* si veda Van Der Maaten (2008) [15].

Capitolo 4

Esempi di applicazione

4.1 Dati audio digitali: il formato MIDI

È stato scelto di considerare i dati audio in formato MIDI ("Music Instrument Digital Interface"). Il vantaggio rispetto a segnali da formati mp3 o wav sta nel fatto che i file con estensione MIDI memorizzano istruzioni per la realizzazione di suoni e non i suoni stessi. Questo ha tre implicazioni:

- i file sono più compatti (un minuto di musica MIDI occupa all'incirca lo stesso spazio di memoria di un secondo di musica in formato wav);
- è più facile estrarre dell'informazione da essi, per come sono strutturati i dati;
- non contengono effetti speciali, ma solo gli elementi essenziali per la riproduzione delle note.

Per una trattazione dettagliata degli aspetti informativi si veda C. Weihs, D. Jan-nach, I. Vatulkin, G. Rudolph (2016) [16]. In questi file, le altezze dei suoni sono codificate con numeri interi ordinati, da 0 a 127, dove ogni unità corrisponde ad un semitono diverso. Ricordando che ogni volta che si raddoppia la frequenza (f) di un suono, si ottiene lo stesso suono ad un intervallo di ottava, si ricava facilmente che il numero n di semitoni che intercorrono tra una nota e quella corrispondente alla frequenza di riferimento (f_{rif}), che è convenzionalmente il *la* del corista (A4, 440 Hz), è dato dalla relazione

$$n = 12 \log_2(f/f_{rif}).$$

Considerando come nota più bassa e quindi come valore minimo la frequenza 8.17 e assegnandole il valore 0, si ha che alla frequenza 440 corrisponde il numero 69, e il valore di ciascuna nota sarà dato da $69 + 12 \log_2(f/f_{rif})$. Per il prospetto

completo della codifica delle frequenze in valori interi secondo questo schema si veda la figura 4.1

MIDI Note	Frequency	MIDI Note	Frequency	MIDI Note	Frequency
C1 0	8.1757989156	12	16.3515978313	24	32.7031956626
Db 1	8.6619572180	13	17.3239144361	25	34.6478288721
D 2	9.1770239974	14	18.3540479948	26	36.7080959897
Eb 3	9.7227182413	15	19.4454364826	27	38.8908729653
E 4	10.3008611535	16	20.6017223071	28	41.2034446141
F 5	10.9133822323	17	21.8267644646	29	43.6535289291
Gb 6	11.5623257097	18	23.1246514195	30	46.2493028390
G 7	12.2498573744	19	24.4997147489	31	48.9994294977
Ab 8	12.9782717994	20	25.9565435987	32	51.9130871975
A 9	13.7500000000	21	27.5000000000	33	55.0000000000
Bb 10	14.5676175474	22	29.1352350949	34	58.2704701898
B 11	15.4338531643	23	30.8677063285	35	61.7354126570
C4 36	65.4063913251	48	130.8127826503	60	261.6255653006
Db 37	69.2956577442	49	138.5913154884	61	277.1826309769
D 38	73.4161919794	50	146.8323839587	62	293.6647679174
Eb 39	77.7817459305	51	155.5634918610	63	311.1269837221
E 40	82.4068892282	52	164.8137784564	64	329.6275569129
F 41	87.3070578583	53	174.6141157165	65	349.2282314330
Gb 42	92.4986056779	54	184.9972113558	66	369.9944227116
G 43	97.9988589954	55	195.9977179909	67	391.9954359817
Ab 44	103.8261743950	56	207.6523487900	68	415.3046975799
A 45	110.0000000000	57	220.0000000000	69	440.0000000000
Bb 46	116.5409403795	58	233.0818807590	70	466.1637615181
B 47	123.4708253140	59	246.9416506281	71	493.8833012561
C7 72	523.2511306012	84	1046.5022612024	96	2093.0045224048
Db 73	554.3652619537	85	1108.7305239075	97	2217.4610478150
D 74	587.3295358348	86	1174.6590716696	98	2349.3181433393
Eb 75	622.2539674442	87	1244.5079348883	99	2489.0158697766
E 76	659.2551138257	88	1318.5102276515	100	2637.0204553030
F 77	698.4564628660	89	1396.9129257320	101	2793.8258514640
Gb 78	739.9888454233	90	1479.9776908465	102	2959.9553816931
G 79	783.9908719635	91	1567.9817439270	103	3135.9634878540
Ab 80	830.6093951599	92	1661.2187903198	104	3322.4375806396
A 81	880.0000000000	93	1760.0000000000	105	3520.0000000000
Bb 82	932.3275230362	94	1864.6550460724	106	3729.3100921447
B 83	987.7666025122	95	1975.5332050245	107	3951.0664100490
C10 108	4186.0090448096	120	8372.0180896192		
Db 109	4434.9220956300	121	8869.8441912599		
D 110	4698.6362866785	122	9397.2725733570		
Eb 111	4978.0317395533	123	9956.0634791066		
E 112	5274.0409106059	124	10548.0818212118		
F 113	5587.6517029281	125	11175.3034058561		
Gb 114	5919.9107633862	126	11839.8215267723		
G 115	6271.9269757080	127	12543.8539514160		
Ab 116	6644.8751612791				
A 117	7040.0000000000				
Bb 118	7458.6201842894				
B 119	7902.1328200980				

Figura 4.1: Relazione tra frequenza delle onde sonore (Hz) e codifica .MIDI

4.2 Il pacchetto tuneR

All'interno dell'ambiente **R** [11], risulta utile, se si vogliono analizzare dei file audio, il pacchetto `tuneR`: permette di aprire, riprodurre, convertire, modificare e creare tracce musicali codificate in diversi formati (mp3, MIDI, wave, wav). Di interesse per estrapolare informazione dai nostri file MIDI sono le funzioni `readMidi` e `getMidiNotes`. Un esempio di dataset che è possibile ottenere con questi comandi è riportato in Figura 4.2. La prima colonna mostra l'indice identificativo dell'osservazione, la singola nota, la variabile `time` indica il momento in cui si registra l'impulso sonoro, `length` ne indica la durata, `track` e `channel` indicano, con numerazioni diverse in brani polifonici, quale strumento esegue la nota, `note` indica numericamente l'altezza della nota nella codifica MIDI, `notename` indica il nome della nota e `velocity` indica l'intensità.

	time	length	track	channel	note	notename	velocity
1	49152	0	2	0	63	d#'	96
2	49152	0	2	0	63	d#'	96
3	52224	0	2	0	67	g'	96
4	52224	0	2	0	67	g'	96
5	55296	0	2	0	70	a#'	96
6	55296	0	2	0	70	a#'	96
7	58368	0	2	0	70	a#'	96
8	58368	0	2	0	70	a#'	96
9	61440	0	2	0	70	a#'	96
10	61440	0	2	0	70	a#'	96
11	64512	0	2	0	70	a#'	96
12	64512	0	2	0	70	a#'	96

Figura 4.2: Dataset ottenuto con le funzioni del pacchetto `tuneR`

4.3 I brani considerati

I brani analizzati sono 15, di 9 diversi autori, scritti per diverse formazioni strumentali:

- 3 brani ottenuti con il “Dice game” di Mozart, citato nel Capitolo 1, che indicheremo con *Pseudo-Mozart 1*, *Pseudo-Mozart 2* e *Pseudo-Mozart 3*.
- *I Corale della Cantata BWV 140* di J. S. Bach
- *Preludio e Fuga BWV 884* di J. S. Bach
- *Improvviso op. 90 n. 2* di F. Schubert
- *Quartetto per archi n.12, I movimento* di F. Schubert

- *Arabesque op. 18* di R. Schumann
- *Notturmo op. 15 n. 2* di F. Chopin
- *Notturmo op. 9 n. 2* di F. Chopin
- *Danza Russa da “Lo schiaccianoci”* di P. I. Tchaikovsky
- *“The snow is dancing”* di C. Debussy
- *“La fille au cheveux de lin”* di C. Debussy
- *Danza in ritmo bulgaro n. 1* di B. Bartok
- *Tre pezzi per piano op. 11 n. 1* di A. Schoenberg

È possibile scaricare i file utilizzati in questo lavoro in formato MIDI dal sito tokunstderfuge.com. In Figura 4.3, i brani sono ordinati in base all'anno in cui sono stati composti.

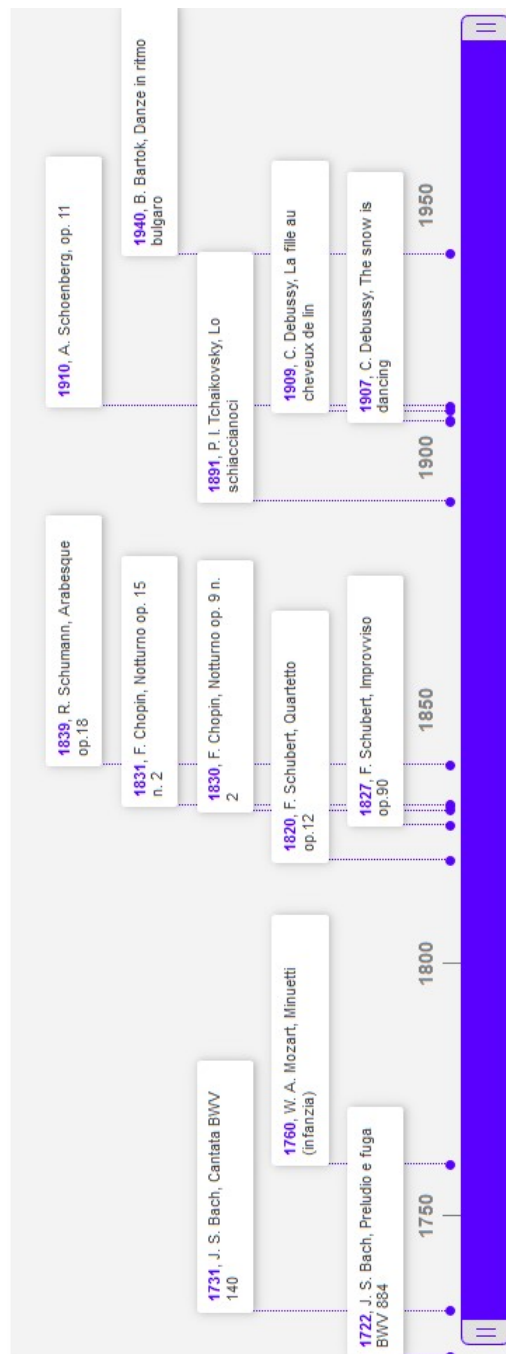


Figura 4.3: Linea del tempo e anno di composizione dei brani analizzati

1. J. S. Bach, Cantata BWV 140, Corale I in mi bemolle maggiore

Il primo brano preso in esame è il primo corale dalla Cantata BWV 140 di J. S. Bach, “*Wachet auf, ruft uns die Stimme*”. In questo esempio abbiamo 8 distinte linee ciascuna associata ad un diverso esecutore. In Figura 4.4, dove è rappresentato un diagramma di dispersione delle variabili tempo e altezza, vengono utilizzati 8 colori diversi per rappresentarle. Il dataset contiene 6.279 osservazioni, relative

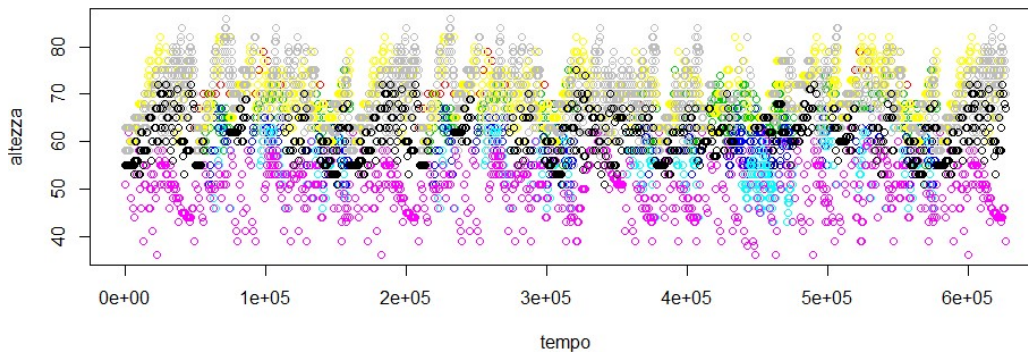


Figura 4.4: J. S. Bach, Cantata BWV 140, Corale I, tempo e altezza delle note

a 2.041 diversi istanti di tempo. Il passo successivo per poter analizzare il brano consiste nel creare un dataset in cui le osservazioni di ciascuna sequenza vengano allineate rispetto al tempo: una prima colonna conterrà i singoli valori della variabile tempo, mentre nelle successive, in numero pari a quante sono le sequenze parallele, verranno inseriti i valori che fanno riferimento all'altezza delle note. In Figura 4.5 sono mostrate le prime sei righe del nuovo dataset. Si osservi che ci sono molti valori mancanti, corrispondono agli istanti in cui lo strumento corrispondente osserva una pausa o tiene la nota precedente; come spiegato nel Capitolo 2, la presenza di valori mancanti non rappresenta un ostacolo per il successivo adattamento di un HMM. Utilizzando l'altezza delle note in modulo 12, sfruttando la

tempo	v1	v2	v3	v4	v5	v6	v7	v8
49152	63	NA	NA	NA	39	75	75	67
52224	67	NA	NA	NA	48	NA	63	55
55296	70	NA	NA	NA	43	67	NA	58
58368	70	63	NA	NA	51	NA	67	NA
61440	70	65	58	NA	50	65	NA	NA
64512	70	NA	67	51	51	NA	79	NA

Figura 4.5: J. S. Bach, Cantata BWV 140, dataset relativo alle sequenze simultanee

natura ciclica dei dati, è possibile studiare la distribuzione dei suoni condensando l'informazione relativa a note molto acute e molto gravi. Come ci si può aspettare si osservano 7 modalità molto frequenti (3,5,7,8,10,0,2): corrispondono alle note della scala relative alla tonalità di impianto del brano, in questo caso $M\flat$, che corrisponde alla cifra 3. Si osservino il grafico a barre e il diagramma a scatola e baffi circolare (Figura 4.6). In figura 4.7 invece sono rappresentate le distribuzioni condizionate per ognuna delle sequenze simultanee.¹ Mediante la funzione

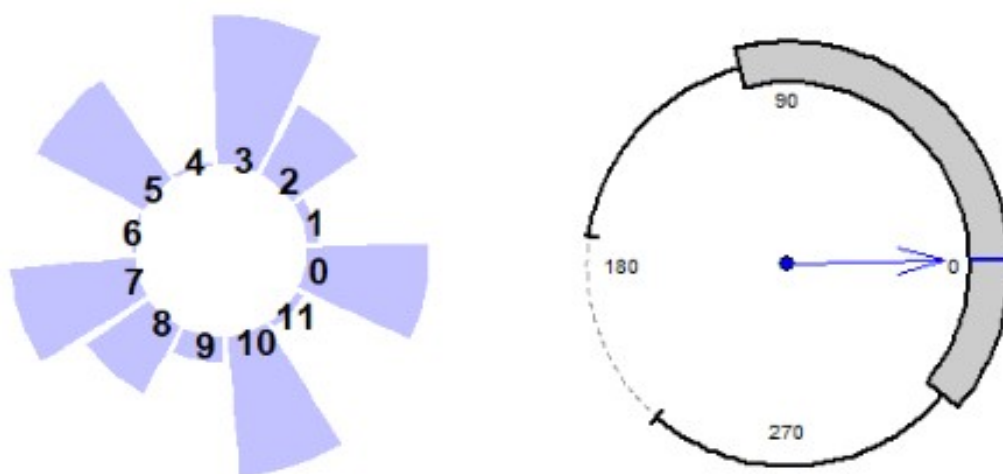


Figura 4.6: J. S. Bach, Cantata BWV 140, Corale I, diagramma a barre e diagramma a scatola circolare

`build_hmm` del pacchetto `seqHMM` [8], che costruisce un oggetto `hmm` contenente le sequenze osservate e l'insieme degli elementi da stimare $\Theta = \{\delta, \Gamma, \Lambda\}$, e la funzione `fit_model` si è provato ad adattare dei modelli utilizzando un numero variabile M di stati nascosti. Per ciascun modello ottenuto si è calcolato il valore dell'AIC, come $-2(\log L) + 2P$, in questo caso $P = M^2 + (MS \times 8)$, e S è una media pesata del numero di stati osservati nelle sequenze, perciò $S \times 8$ rappresenta il numero totale di colonne delle matrici di emissione. Nel grafico a sinistra di Figura 4.8'è rappresentata sull'asse delle ordinate la log-verosimiglianza, l'adattamento sembra migliorare notevolmente passando da 1 a 6 stati nascosti, dopodiché la curva sembra raggiungere un plateau. Nel grafico a destra si osserva

¹ Il diagramma a barre circolare consente una immediata visualizzazione della distribuzione dei dati. La freccia indica la direzione della media circolare, definita come il punto per cui è minima la somma pesata delle distanze tra gli archi: in pratica mostra qual è il baricentro della distribuzione. L'ordinamento di posizione (il rango) viene assegnato a partire dall'antimedia circolare (nella direzione opposta a quella della media). I margini della scatola rappresentano i quartili della distribuzione. Si veda D. Buttarazzi, G. Pandolfo, G. C. Porzio (2018) [5].

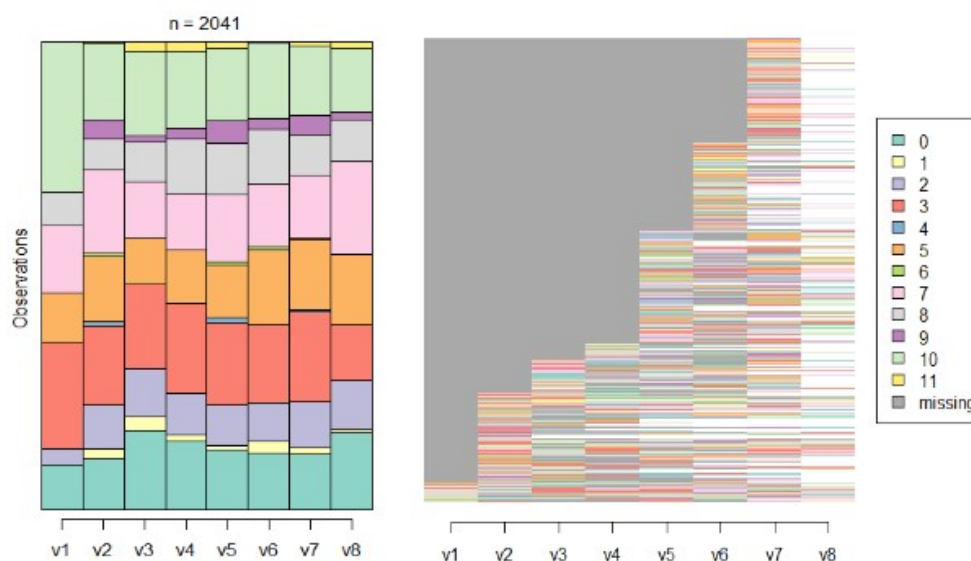


Figura 4.7: J. S. Bach, Cantata BWV 140. A sinistra, distribuzione delle sequenze distinte, a destra, rappresentazione delle successioni ordinate e frequenza di missing value

il valore dell'AIC, si ha la conferma che il numero di stati che identifica il modello corretto sia 6. Si nota un'anomalia nella stima del modello con 5 stati nascosti, in quanto ci si aspetterebbe che il valore della log-log-verosimiglianza massimizzata sia maggiore rispetto quello calcolato con quattro stati. L'algoritmo utilizzato dal pacchetto sembra essersi arrestato in corrispondenza di un massimo locale, ripetendo la procedura e portando il numero massimo di iterazioni da 10.000 a 10.000.000.000, il problema non si è eliminato.

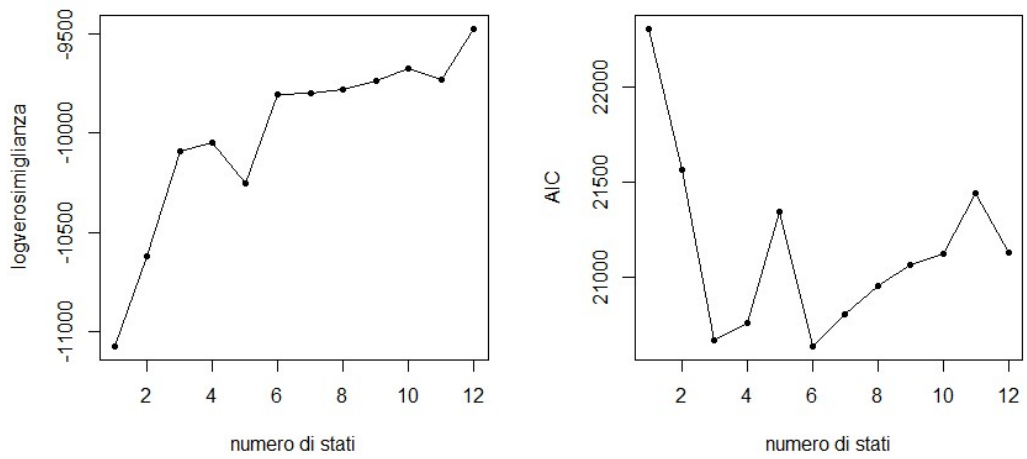


Figura 4.8: J. S. Bach, Cantata BWV 140, log-verosimiglianza massimizzata e criterio di Akaike al variare del numero di stati nascosti stimati

2. Pseudo-Mozart 1

In rete, digitando “Musikalische Würfelspiele” o “Mozart dice game”, si riescono a trovare diversi software in cui è implementato il “gioco dei dadi di Mozart” : a partire da dei valori iniziali forniti dall’utente, viene generata una sequenza di numeri casuali e con questi creato un minuetto in stile pianistico, giustappo- nendo dei frammenti di melodie mozartiane (si veda il Capitolo 1). Poiché i brani che si ottengono sono polifonici, ed è di interesse valutare come si articola lun- go ciascuna sequenza la successione delle note, le diverse linee melodiche sono state separate con una tecnica di clustering, utilizzando K-Means su una trasfor- mazione delle variabili altezza e tempo. In questo caso sono state usate le variabili $time^{3/10}$ e $note^{6/5}$; sono stati creati tre gruppi. La scelta di elevare a potenze diverse i valori del tempo e dell’altezza risponde all’esigenza di creare gruppi di forma longitudinale. Senza applicare questa trasformazione, poiché i valori che assume la variabile tempo sono molto elevati rispetto ai valori dell’, la separazione avviene per “macro-periodi”. Sono stati fatti alcuni tentativi con una procedura di clustering gerarchico, ma con scarsi risultati. Nei diagrammi di dispersione tempo-altezza riportati in Figura 4.9, le note dei diversi gruppi, che cioè fanno ca- po a diverse linee melodiche sono rappresentate con diversi colori. Vengono poi riportati i grafici relativi alla frequenza delle note (sempre in modulo 12) e il dia- gramma a scatola circolare. Si ha poi il grafico delle frequenze condizionate per le tre linee individuate. (Figure 4.10 e 4.11). Anche per questo brano sono stati adattati dei modelli prendendo in considerazione diverse possibilità per il nume- ro di stati nascosti. Il valore della log-verosimiglianza sembra aumentare fino in corrispondenza 6 stati, dopodiché il miglioramento sembra essere esiguo. Il crite- rio di Akaike suggerisce invece un numero di stati nascosti decisamente inferiore. Questo è dovuto anche al fatto che si hanno poche osservazioni, 513 , relative a 480 valori del tempo distinti, pertanto la log-verosimiglianza, in cui compare il prodotto di tante matrici di transizione quanti sono gli istanti temporali distinti, assume dei valori non molto diversi al variare del numero di stati nascosti, e la penalità dell’AIC ha un impatto forte (Figura 4.12).

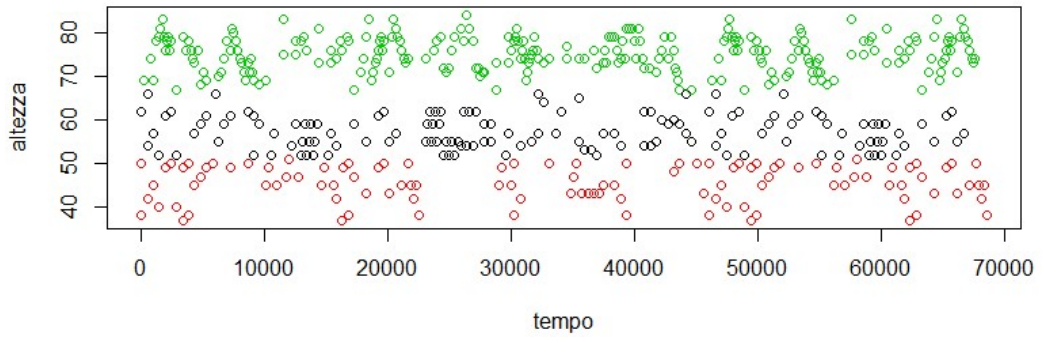


Figura 4.9: Pseudo-Mozart 1, tempo e altezza delle note

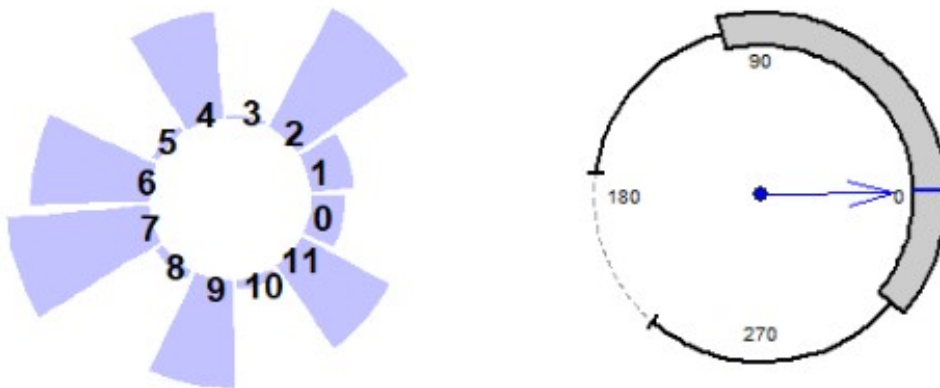


Figura 4.10: Pseudo-Mozart 1, diagramma a barre e diagramma a scatola circolare

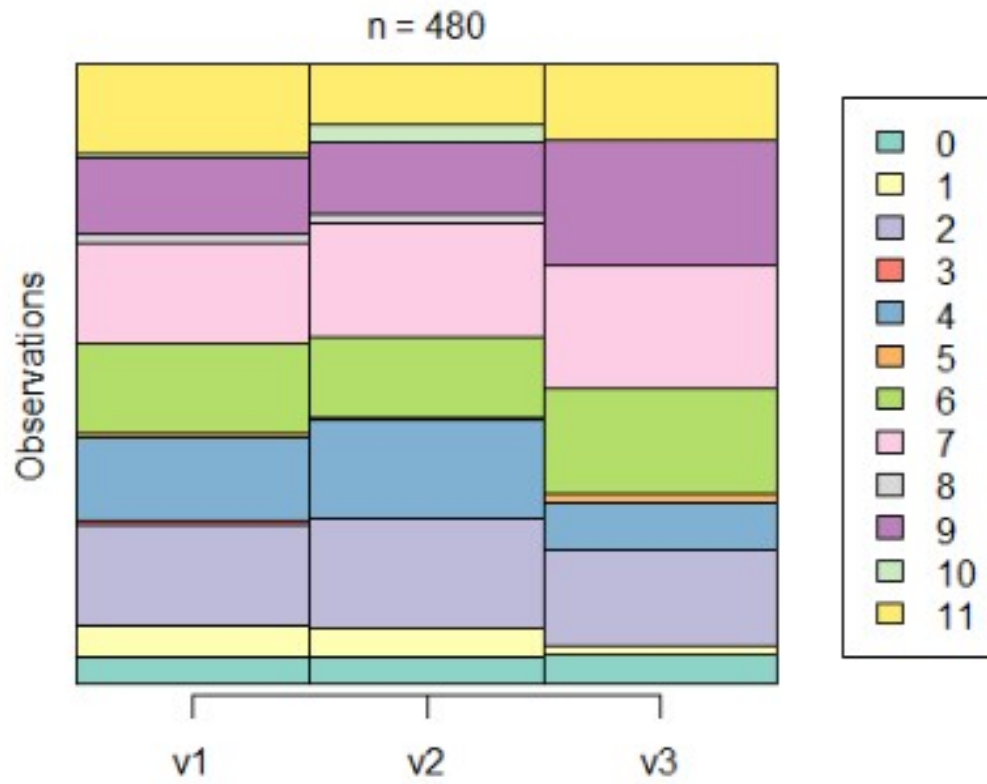


Figura 4.11: Pseudo-Mozart 1, distribuzione delle sequenze distinte

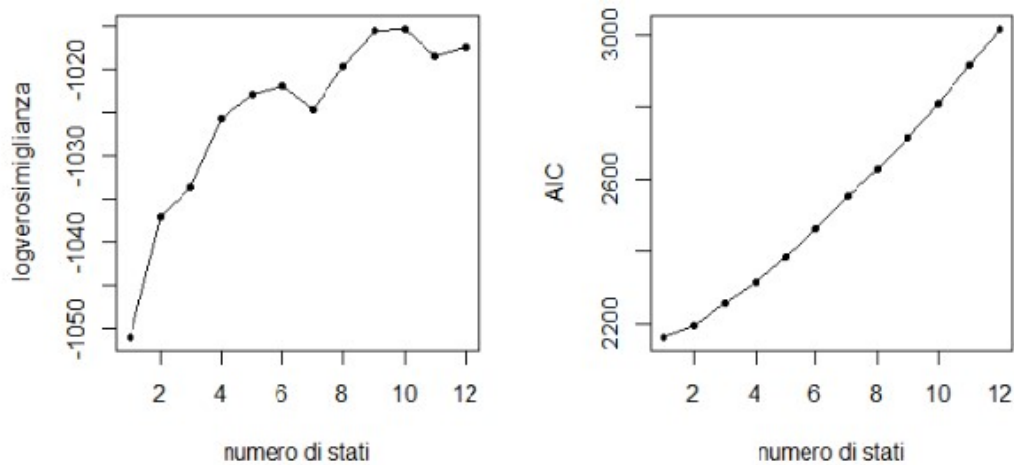


Figura 4.12: Pseudo-Mozart 1, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

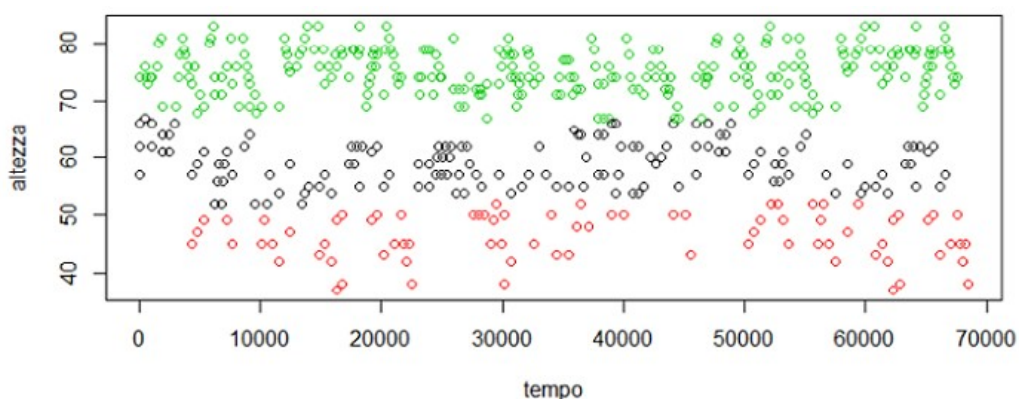


Figura 4.13: Pseudo-Mozart 2, tempo e altezza delle note

3. Pseudo-Mozart 2

Per ottenere i tre gruppi nel secondo brano aleatorio in stile mozartiano si sono usate le trasformazioni $\log(\text{time}^{0.3})$ e $\log(\text{note}^{1.1})$. L'elevamento a potenze diverse serve a dare, anche in questo caso, maggiore importanza all'altezza delle note, tuttavia la trasformazione da sola fa sì che le osservazioni relative a note molto acute vengano separate in modo molto netto dalle restanti, anche qualora facciano parte di una medesima linea melodica. Pertanto è risultato utile applicare anche il logaritmo, per uniformare le grandezze e ottenere una suddivisione delle sequenze che approssimasse quella reale. Ascoltando il brano e scorrendo visivamente i punti nel diagramma di dispersione (Figura 4.13), si possono abbastanza facilmente riconoscere le linee melodiche sovrapposte. La distribuzione delle note in ciascuna di queste, sembra essere simile (Figura 4.15). Il secondo brano presenta una distribuzione delle note abbastanza dissimile dal primo, (si veda la Figura 4.14). Infatti la tonalità di partenza è diversa. Ciononostante, essendo presenti nell'insieme di dati soltanto 449 osservazioni per 490 valori del tempo, il comportamento della funzione di log-verosimiglianza, così come il valore dell'AIC (Figura 4.16), è analogo al caso precedente.

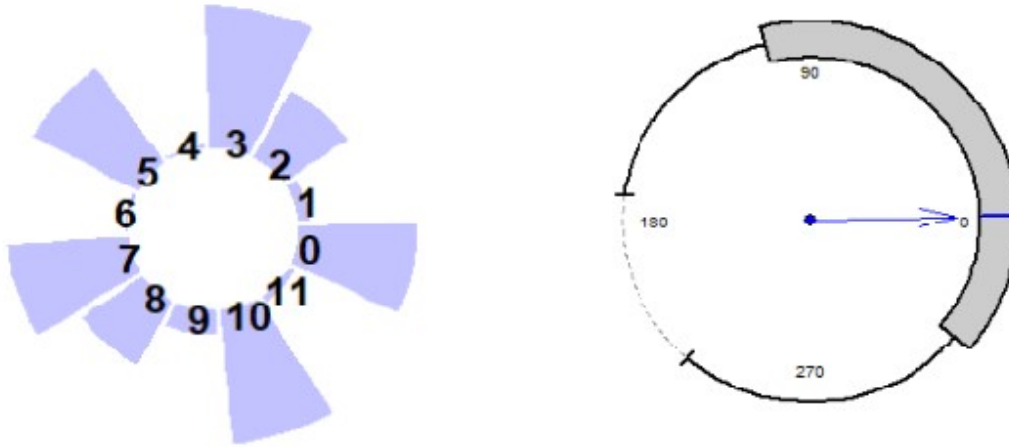


Figura 4.14: Pseudo-Mozart 2, diagramma a barre e diagramma a scatola circolare

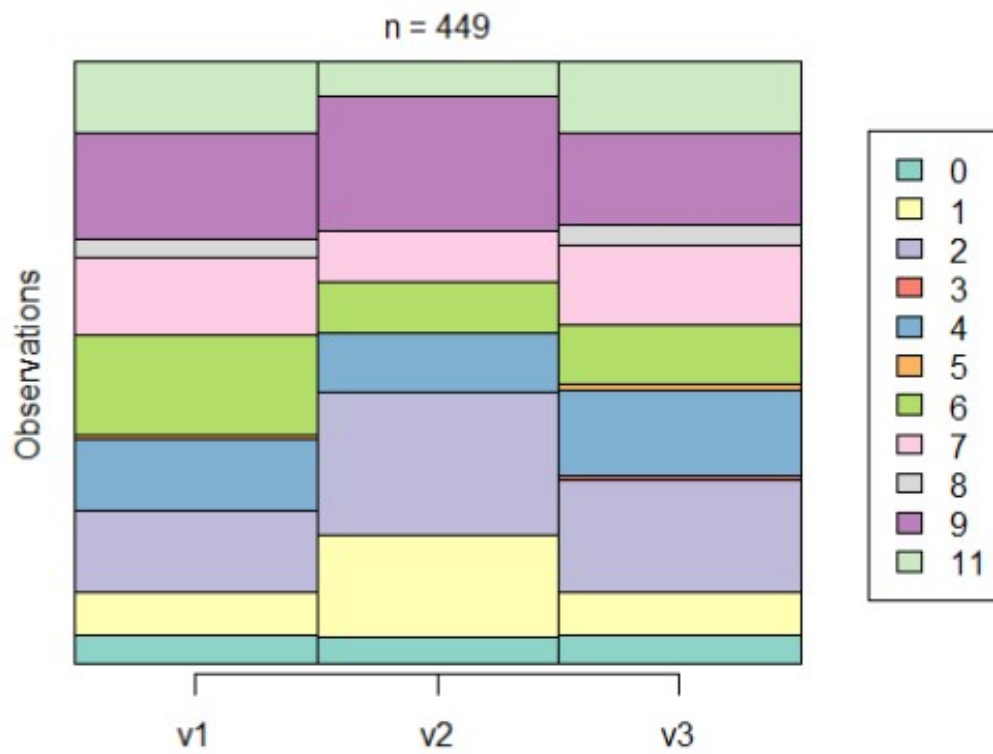


Figura 4.15: Pseudo-Mozart 2, distribuzione delle sequenze distinte

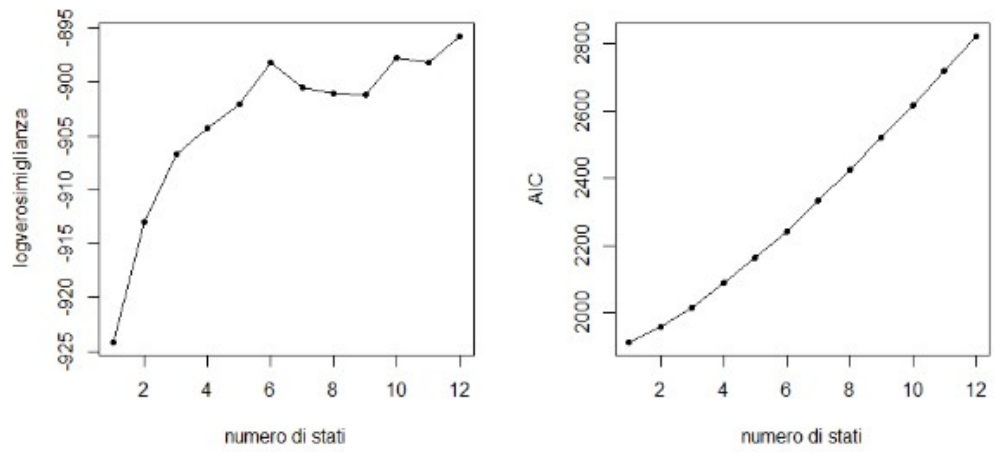


Figura 4.16: Pseudo-Mozart 2, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

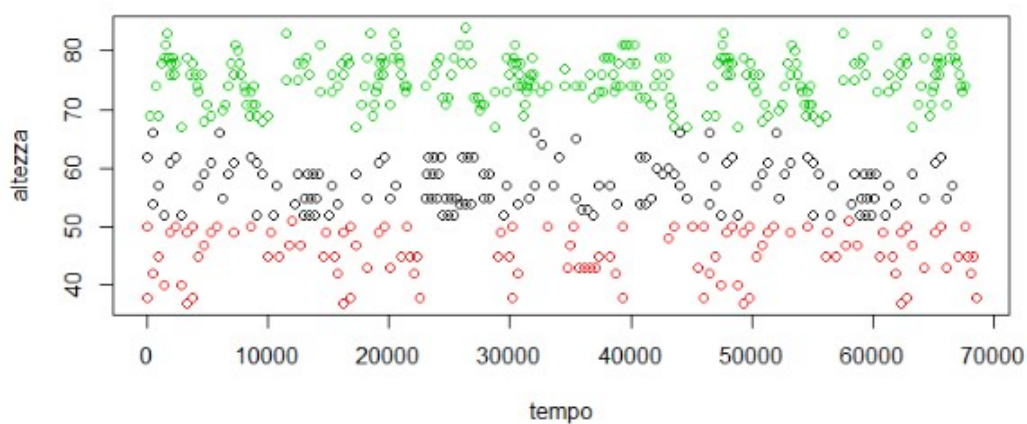


Figura 4.17: Pseudo-Mozart 3, tempo e altezza delle note

4. Pseudo-Mozart 3

Il numero di note distinte è 495, fanno riferimento a 460 diversi istanti temporali. La trasformazione utilizzata per definire i tre gruppi con K-Means è $\log(\text{time}^{3/10})$ e $\log(\text{note}^{11/10})$. Questa sembra aver prodotto dei buoni raggruppamenti (si veda la Figura 4.17). La distribuzione delle note è quasi identica a quella relativa al primo dei brani pseudo-mozartiani, (Figura 4.18), lo stesso vale per la distribuzione per ognuna delle linee (Figura 4.19). Si osservi che il numero di stati “ideale” secondo il criterio di Akaike rimane 1 (Figura 4.20).

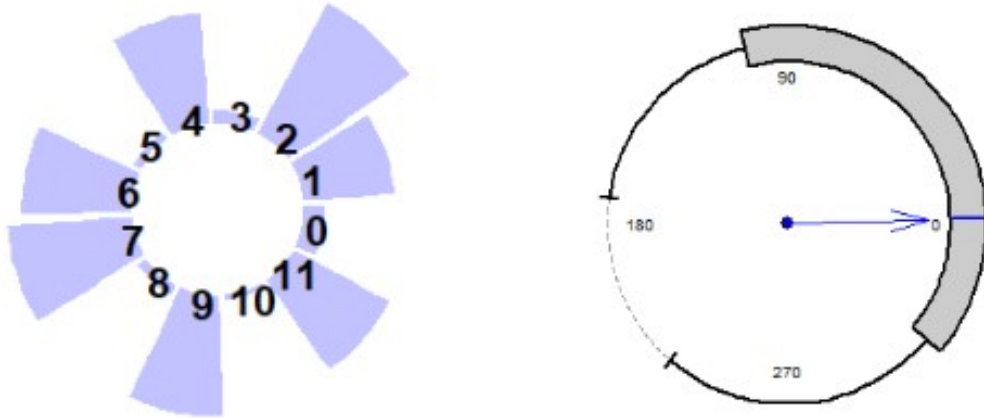


Figura 4.18: Pseudo-Mozart 3, diagramma a barre e diagramma a scatola circolare

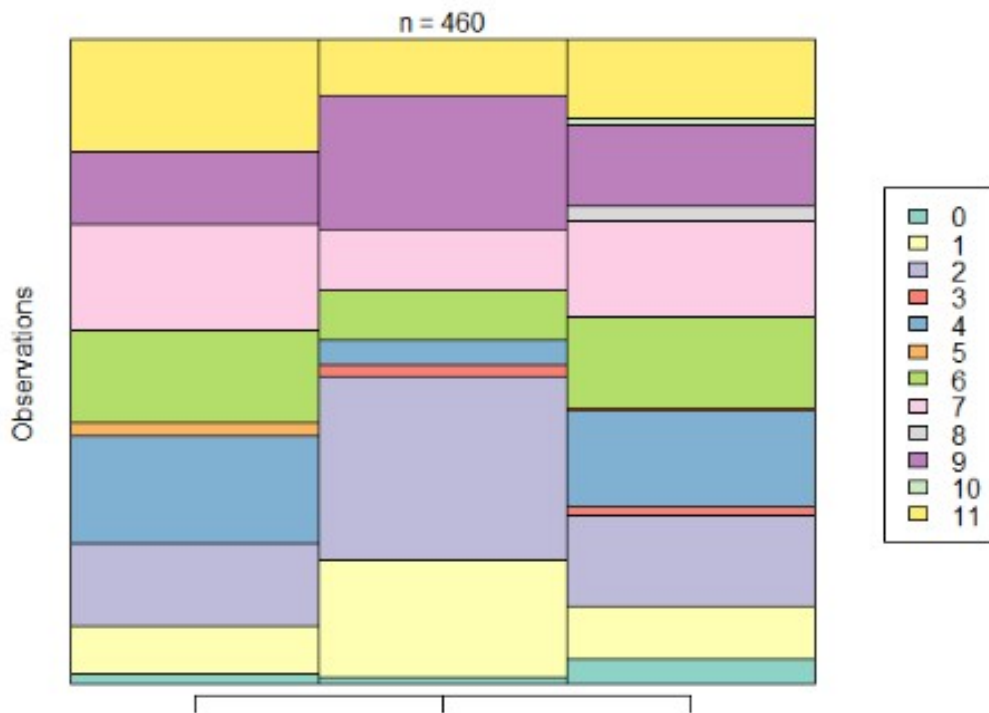


Figura 4.19: Pseudo-Mozart 3, distribuzione delle sequenze distinte

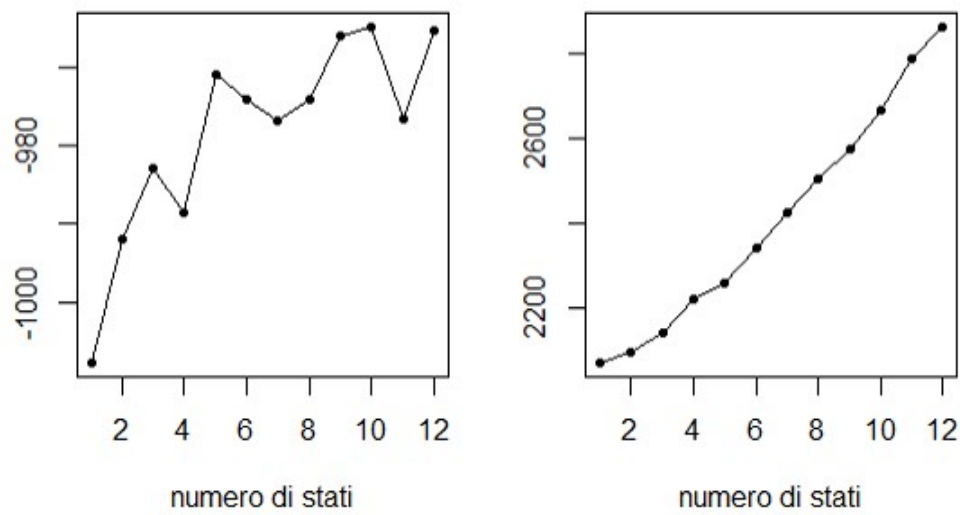


Figura 4.20: Pseudo-Mozart 3, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

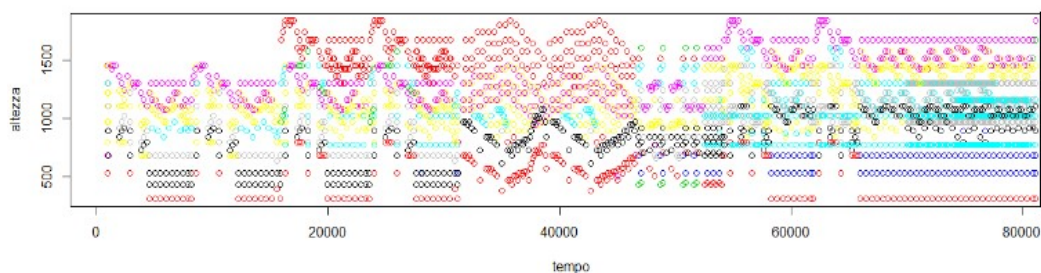


Figura 4.21: Danza Russa, tempo e altezza delle note

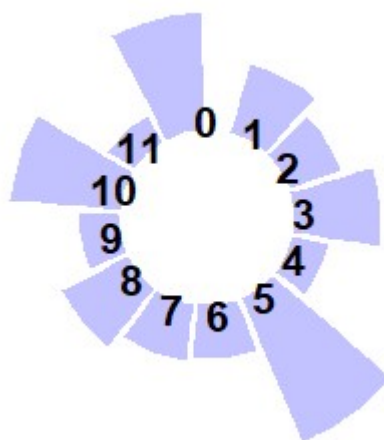


Figura 4.22: Danza Russa, barplot

5. Danza Russa, da “Lo schiaccianoci”, P. I. Tchaikovsky

Il brano seguente è una composizione per orchestra: si distinguono ben 17 sequenze parallele (si veda la Figura 4.21). La distribuzione delle note è diversa all'interno delle varie sequenze (Figura 4.23), quella complessiva è rappresentata nel diagramma a barre in Figura 4.22. Alcune di queste presentano solo due modalità, (è il caso delle percussioni). È un pezzo abbastanza breve, si hanno 810 tempi distinti ma poiché si osservano molte sequenze simultanee, le osservazioni sono 5.448. Per questo brano non è stato possibile costruire il diagramma a scatola circolare, la direzione dell'anti-media non viene identificata. Il numero di stati che possiamo ritenere ottimo per il criterio di Akaike è 6 (Figura 4.24).

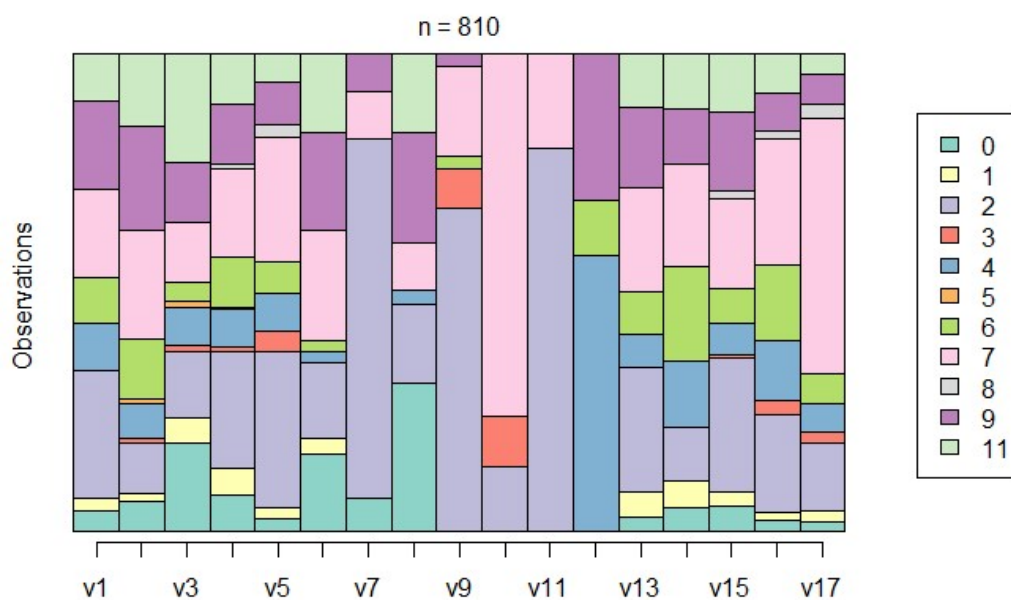


Figura 4.23: Danza Russa, distribuzione delle sequenze distinte

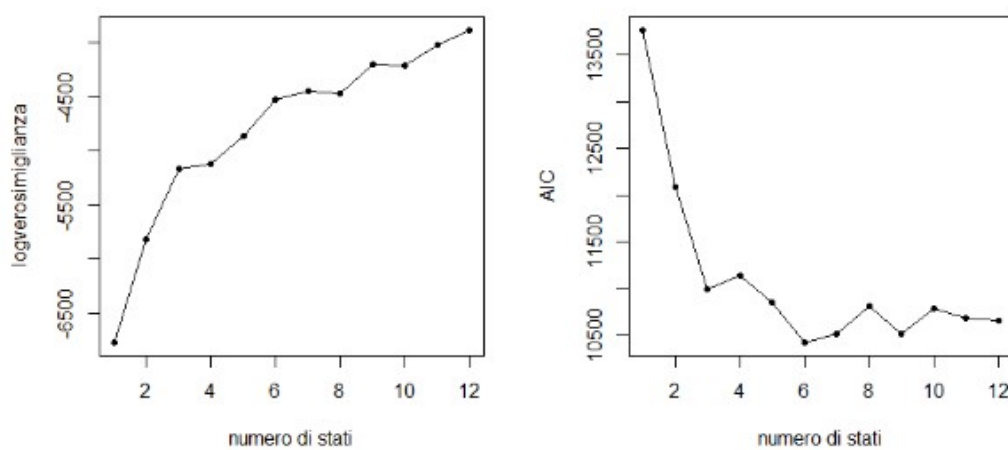


Figura 4.24: Danza Russa, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

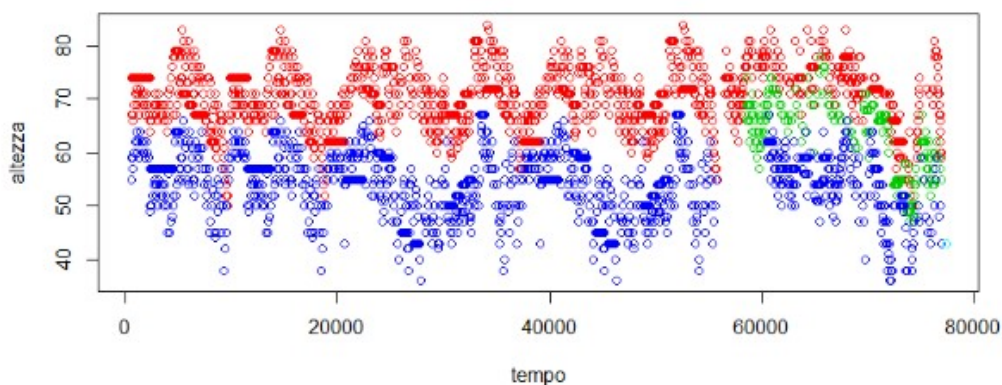


Figura 4.25: Preludio e Fuga BWV 884, J. S. Bach, tempo e altezza delle note

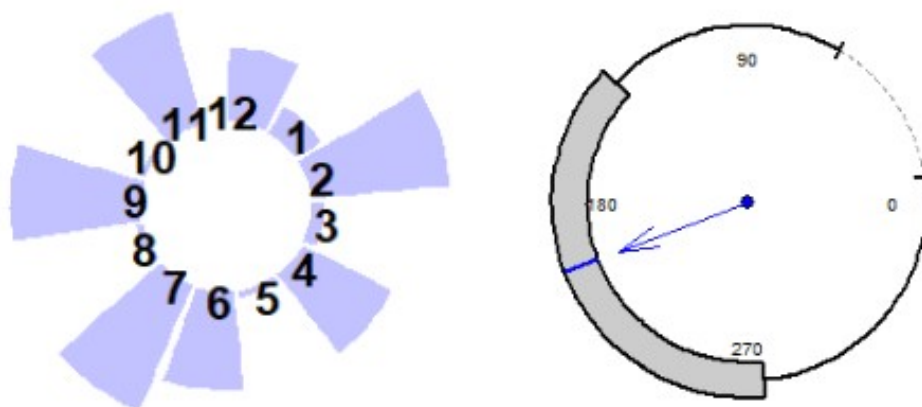


Figura 4.26: Preludio e Fuga BWV 884, J. S. Bach, diagramma a barre e diagramma a scatola circolare

6. Preludio e Fuga BWV 884, J. S. Bach

Si tratta in realtà di una coppia di brani per strumento a tastiera; sono stati analizzati come fossero un unico poiché la forma del *Preludio e fuga* prevede una forte connessione tra le due parti, prendono le mosse da simili spunti, hanno lo stesso carattere. Le osservazioni totali sono 2.545, e sono collocate in 1.637 istanti; nel file le sequenze sono già suddivise, sono 3 (Figura 4.25), le distribuzioni delle note sono rappresentate nei grafici in Figura 4.27. La tonalità del brano è “sol maggiore” (la nota *sol* corrisponde al numero 7), la direzione della media circolare è coerente con questo dato (Figura 4.26). Il numero di stati nascosti ottimale risulta 6, in modo analogo al primo brano di Bach analizzato (Figura 4.28).

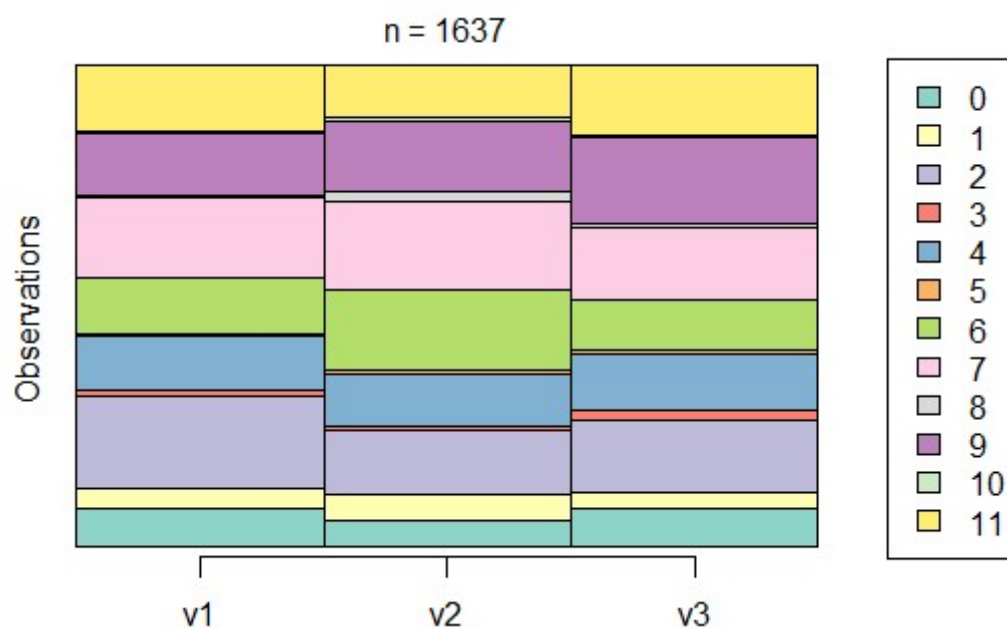


Figura 4.27: Preludio e Fuga BWV 884, J. S. Bach, distribuzione delle sequenze distinte

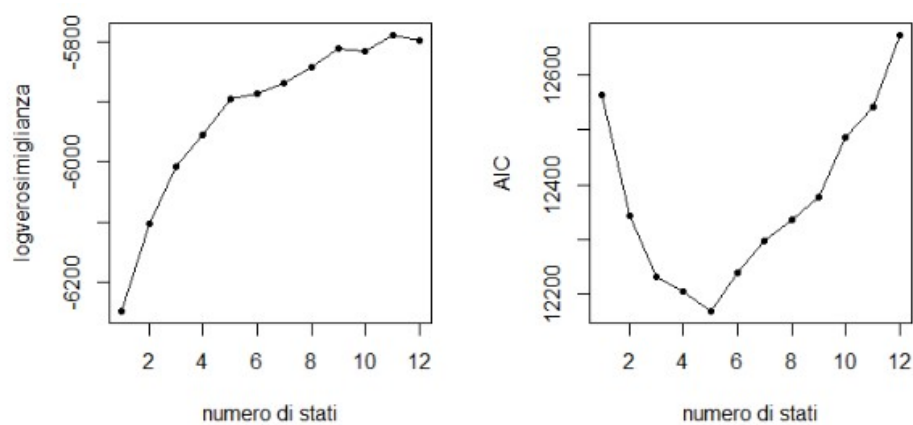


Figura 4.28: Preludio e Fuga BWV 884, J. S. Bach, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

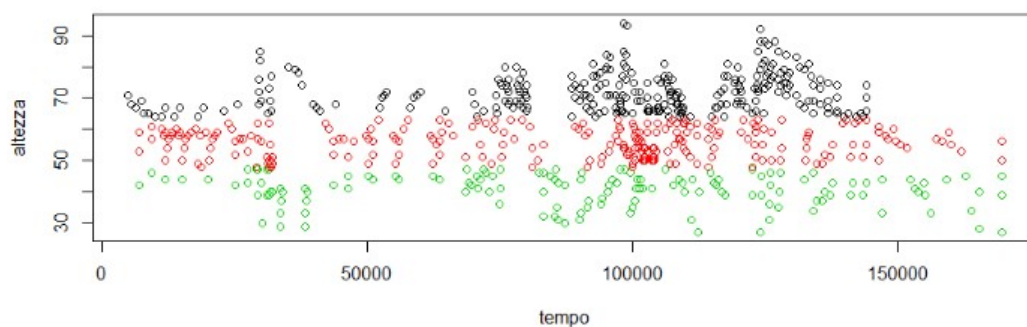


Figura 4.29: Tre pezzi per piano, op. 11 n 1, A. Schoenberg, tempo e altezza delle note



Figura 4.30: Tre pezzi per piano, op. 11 n 1, A. Schoenberg, diagramma a barre e diagramma a scatola circolare

7. Tre pezzi per piano, op. 11 n. 1, A. Schoenberg

Il seguente è un pezzo dal carattere aforistico, molto frammentato. Si differenzia dagli altri perché non gli si può attribuire una tonalità. Le 3 sequenze sono state separate utilizzando K-means e la trasformazione $\log(\text{time}^{0.12})$, $\log(\text{note}^{0.98})$; le linee sono rappresentate con diversi colori nel grafico di Figura 4.29. La distribuzione delle note è uniforme (si vedano le Figure 4.30 e 4.31), ciò corrisponde ad una prerogativa della corrente della musica seriale, nella quale il brano si colloca: i principi destrutturalisti di “non ripetizione” e “variazione costante” garantiscono la medesima importanza alle note. Le osservazioni totali sono 672, distribuite in 565 valori del tempo; il criterio di Akaike indica che il numero ottimo di stati nascosti è (Figura 4.32).

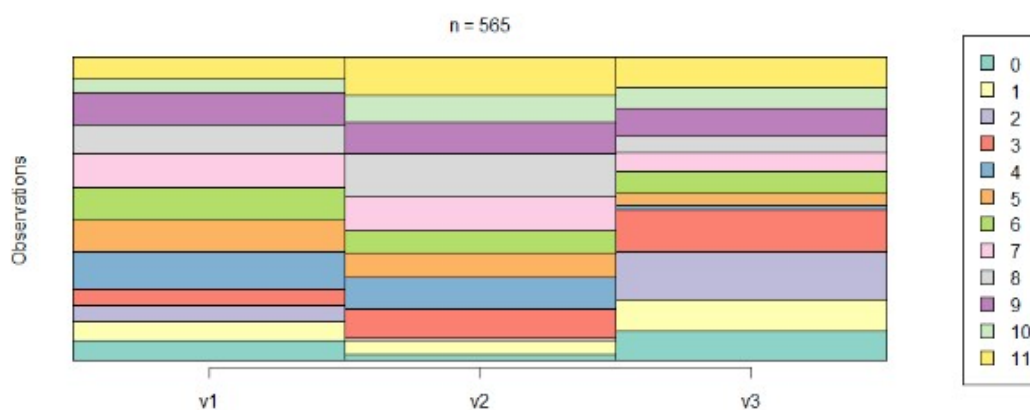


Figura 4.31: Tre pezzi per piano, op. 11 n 1, A. Schoenberg, distribuzione delle sequenze distinte

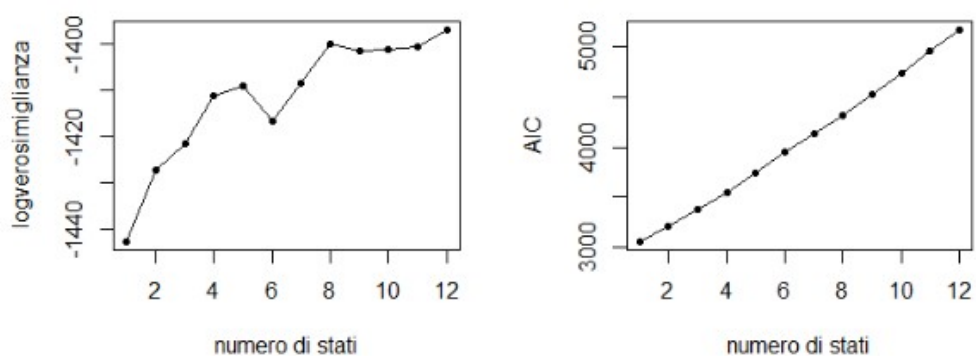


Figura 4.32: Tre pezzi per piano, op. 11 n 1, A. Schoenberg, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

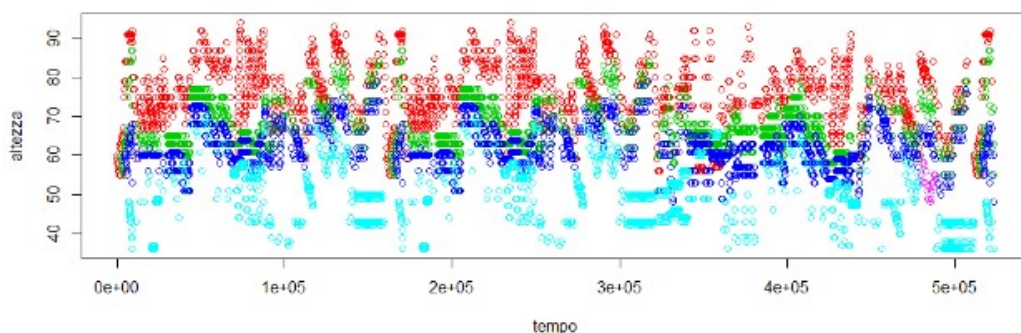


Figura 4.33: Quartetto per archi n. 12, F. Schubert, tempo e altezza delle note

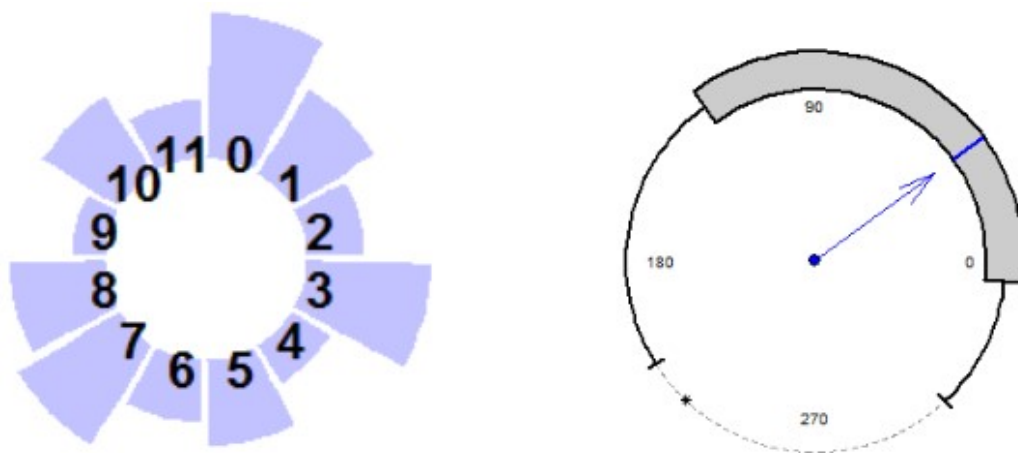


Figura 4.34: Quartetto per archi n. 12, F. Schubert, diagramma a barre e diagramma a scatola circolare

8. Quartetto per archi n. 12, F. Schubert

Il Quartetto per archi n.12 si presenta nella tonalità di do minore. La forma del quartetto è tipica in musica del periodo Classico. La media angolare sembra approssimativamente riconoscere la direzione dello 0, che corrisponde proprio al *do* (Figura 4.34). Si possono distinguere 5 sequenze; uno dei quattro strumenti esegue nell'ultima parte una doppia linea (si veda il diagramma di dispersione in Figura 4.33). Il brano è piuttosto lungo e denso di note, le osservazioni totali sono 7.771, relative a 3.026 istanti temporali (Figura 4.35); il vero modello generatore dei dati, secondo l'AIC, è basato su 9 stati nascosti (si veda il grafico in Figura 4.36).

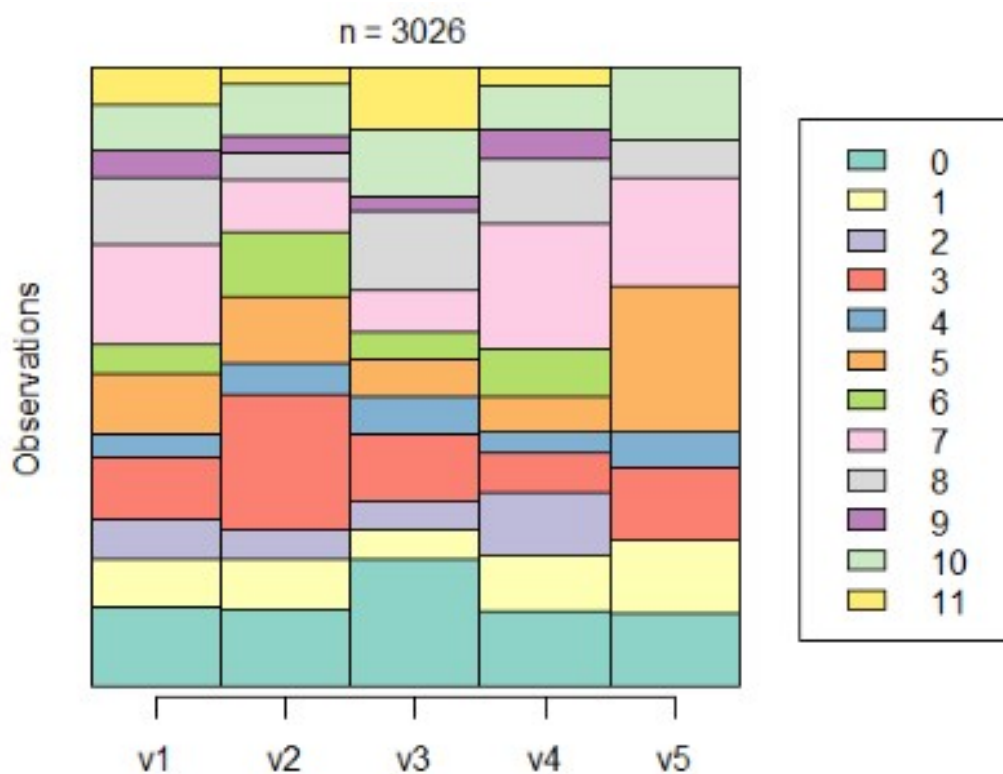


Figura 4.35: Quartetto per archi n. 12, F. Schubert, distribuzione delle sequenze distinte

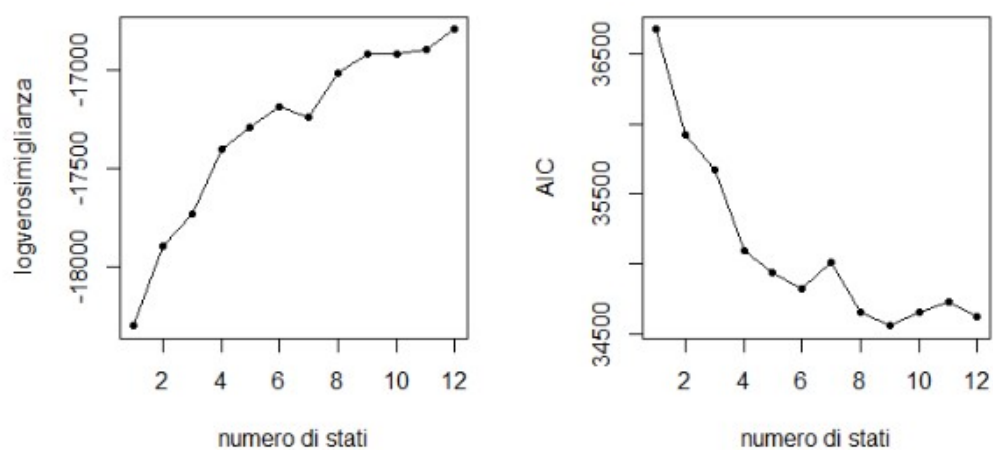


Figura 4.36: Quartetto per archi n. 12, F. Schubert, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

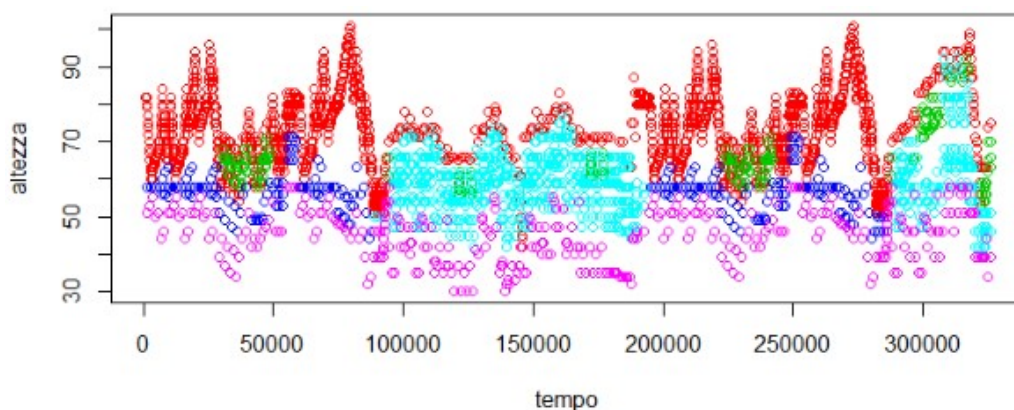


Figura 4.37: Imrovviso op. 90 n. 2, F. Schubert, tempo e altezza delle note

9. Imrovviso op. 90 n. 2, F. Schubert

Questa è una composizione per pianoforte, nella tonalità di mi bemolle maggiore, la media angolare risulta opposta alla direzione che ci si aspetta, ovvero quella del 3 (Figura 4.38). Possiamo spiegare questo fenomeno con una considerazione di tipo storico-musicale. Infatti il brano è riconducibile alla corrente del Romanticismo, e come la quasi totalità dei pezzi scritti nello spaccato temporale corrispondente è caratterizzato da una forte tensione emotiva. In musica la tensione si ottiene polarizzando i suoni intorno al quinto grado della scala, in questo caso il *sol*, la cui cifra corrispondente è il 10. Anche in questo caso si possono distinguere 5 sequenze, (5 sono i colori utilizzati in Figura 4.37), e per queste la distribuzione delle note è osservabile in Figura 4.39. Il brano è abbastanza breve ma le note sono molto fitte. Il numero totale di istanti temporali è 2.092, mentre le osservazioni sono 3.144; l'adattamento migliore, secondo il criterio di Akaike prevede un HMM con 11 stati nascosti (si veda la Figura 4.40).

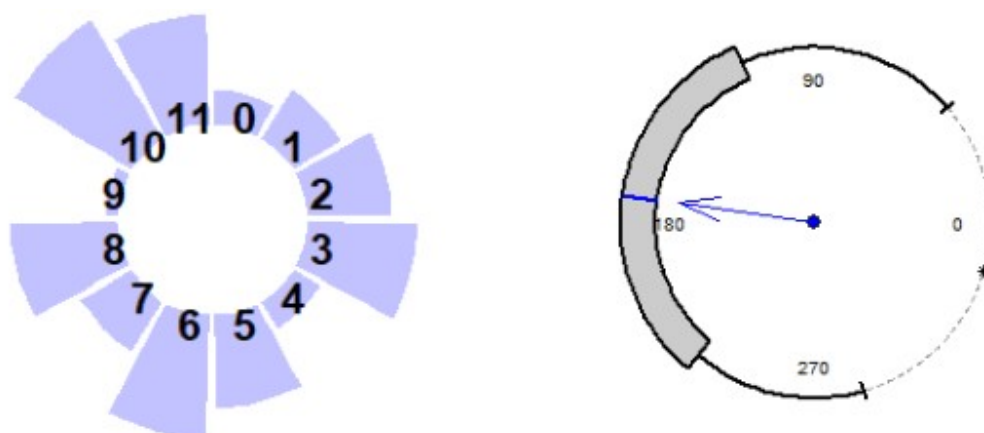


Figura 4.38: Imrovviso op. 90 n. 2, F. Schubert, diagramma a barre e diagramma a scatola circolare

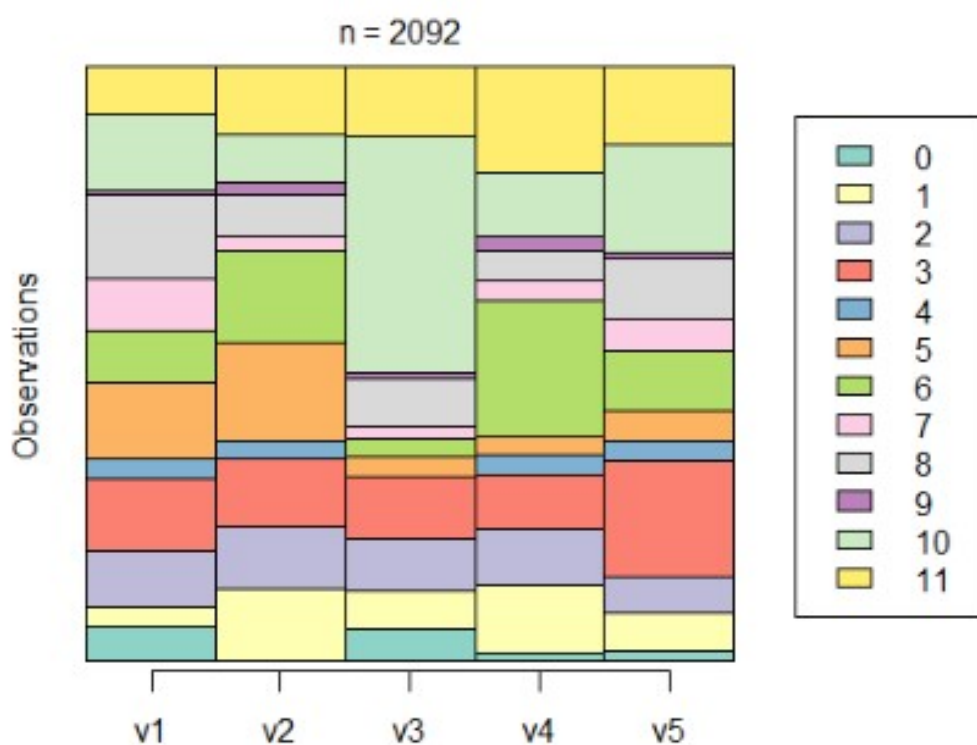


Figura 4.39: Imrovviso op. 90 n. 2, F. Schubert, distribuzione delle sequenze distinte

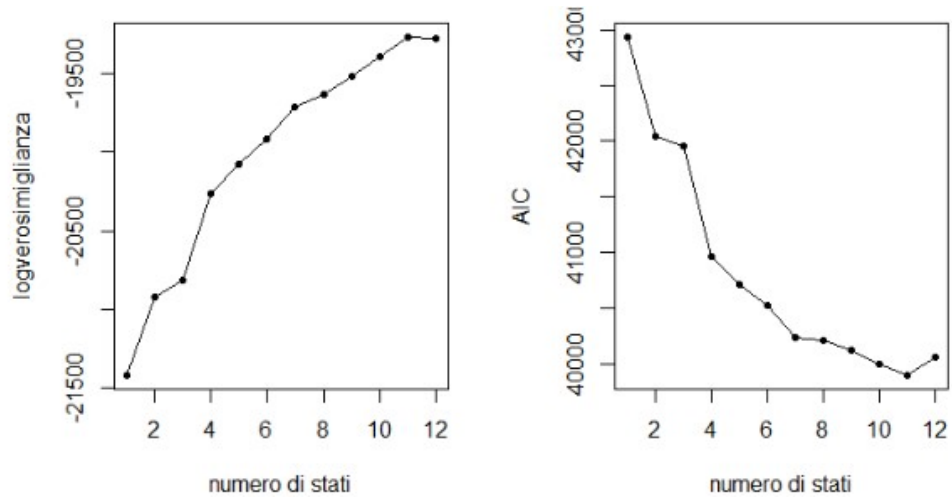


Figura 4.40: Imrovviso op. 90 n. 2, F. Schubert, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

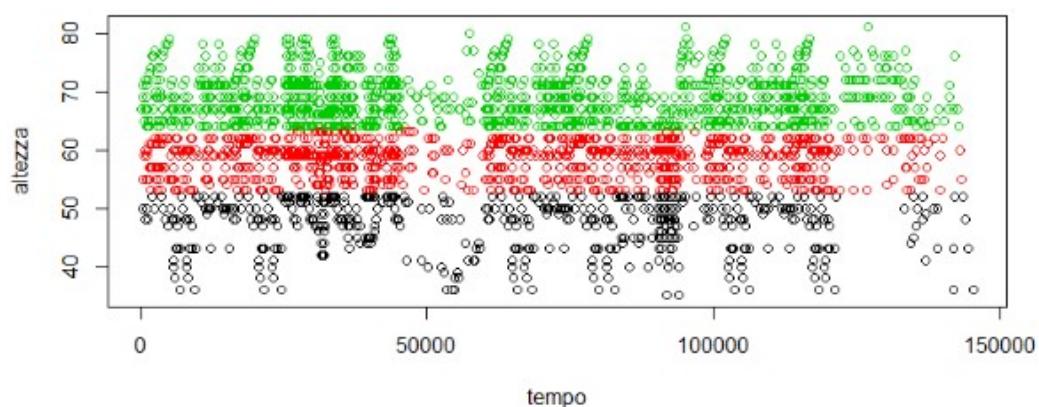


Figura 4.41: Arabesque op. 18 in do maggiore, R. Schumann, tempo e altezza delle note

10. Arabesque op. 18 in do maggiore, R. Schumann

Il brano che segue è anch'esso per pianoforte, di stampo romantico. La tonalità di impianto è do maggiore; la direzione della media angolare è opposta alla direzione in cui si trova lo 0, il *do*, la cifra puntata dalla freccia è il 7 (Figura 4.42). Sono state ricavate 3 distinte sequenze utilizzando K-means e la trasformazione $\log(\text{time}^{0.12})$, $(\log \text{note}^{0.58})$, i risultati della procedura di clustering sono visibili nel diagramma di dispersione in Figura 4.41, mentre in Figura 4.43 si possono vedere le distribuzioni di frequenza delle note per ciascuna linea identificata. Il numero di osservazioni è elevato, se ne hanno 2.554, in 2.275 istanti. Il numero di stati nascosti che garantisce un adattamento migliore, secondo il criterio di Akaike è 3 (Figura 4.44).

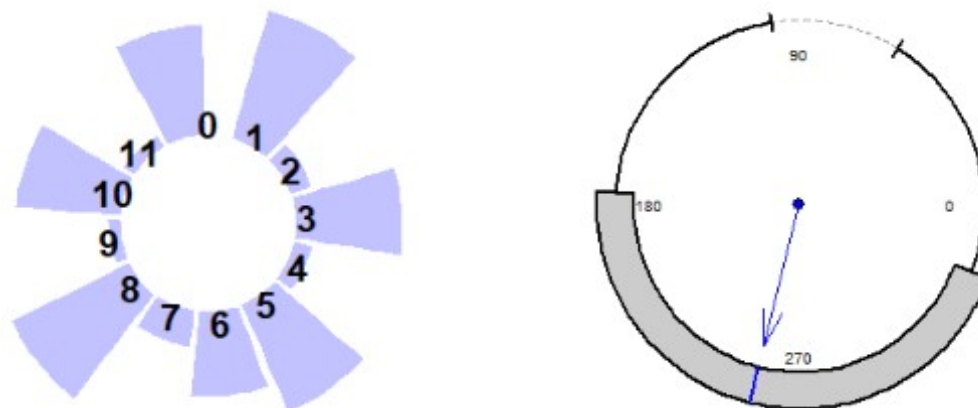


Figura 4.42: Arabesque op. 18 in do maggiore, R. Schumann, diagramma a barre e diagramma a scatola circolare

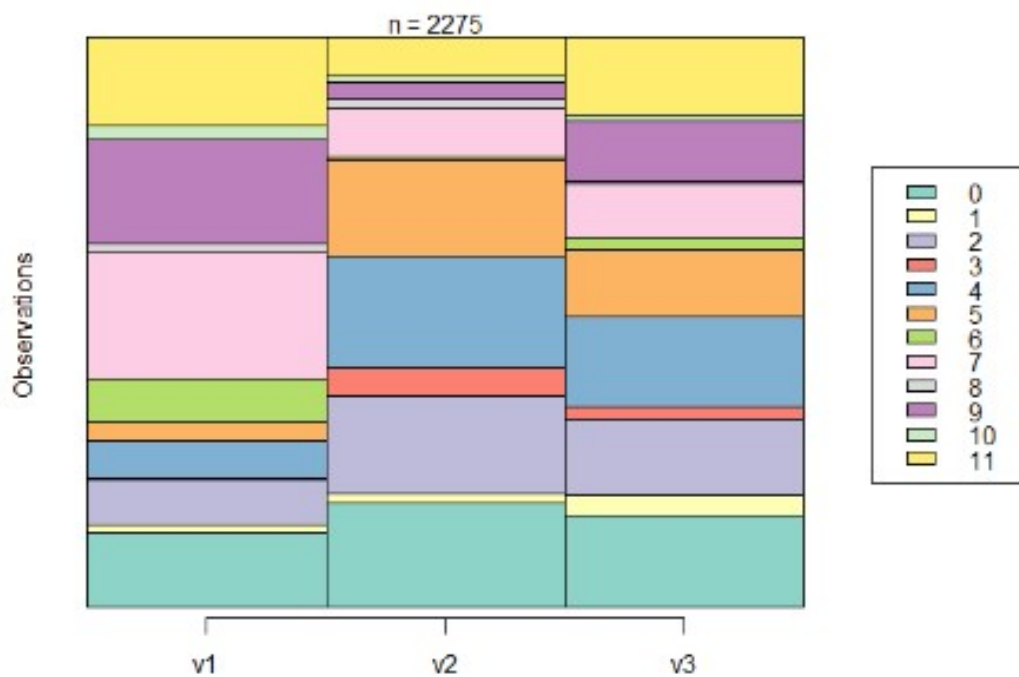


Figura 4.43: Arabesque op. 18 in do maggiore, R. Schumann, distribuzione delle sequenze distinte

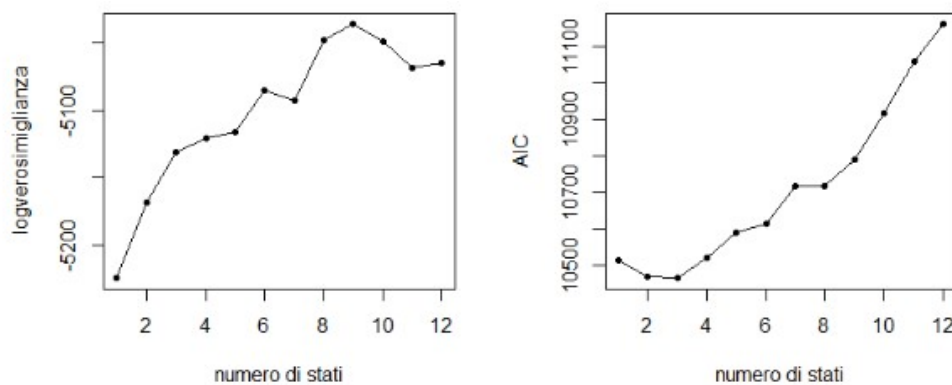


Figura 4.44: Arabesque op.18 in do maggiore, R. Schumann, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

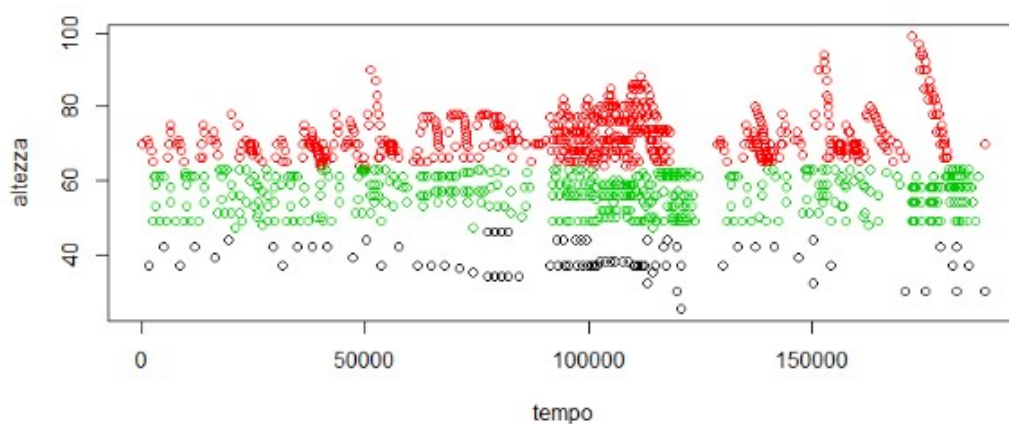


Figura 4.45: Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, tempo e altezza delle note

11. Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin

Con il Notturmo op. 15 n. 2 in fa diesis minore si ricade ancora nella categoria dei brani per pianoforte, scritti nel periodo del Romanticismo. Anche in questo caso la direzione della media angolare è opposta alla direzione in cui si trova la cifra 6, il *fa diesis* (Figura 4.46). La separazione delle linee melodiche è avvenuta mediante K-means e la trasformazione delle variabili tempo e altezza in $\text{time}^{0.2}$, $\log(\text{note}^{2.5})$; sono stati ricavate tre sequenze sovrapposte, per ciascuna delle quali la distribuzione delle note è riportata in Figura 4.47. Le osservazioni presenti sono 1.168, relative a 1.124 diversi istanti. Il numero di stati nascosti che garantisce l'adattamento migliore, secondo il criterio di Akaike è 1 (si veda il grafico in Figura 4.48). Questa caratteristica sembra stonare con gli altri brani del periodo. Si osservi però che l'andamento del brano, rispetto ai tre precedenti è molto più statico. Si veda il diagramma di dispersione tempo-altezza in Figura 4.45

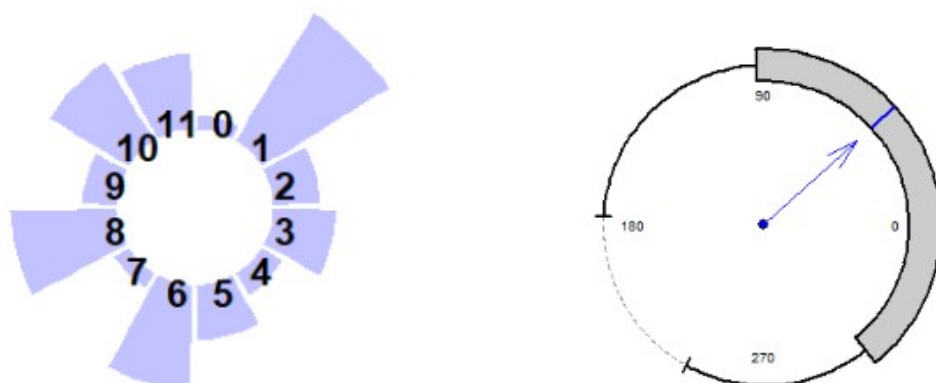


Figura 4.46: Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, diagramma a barre e diagramma a scatola circolare

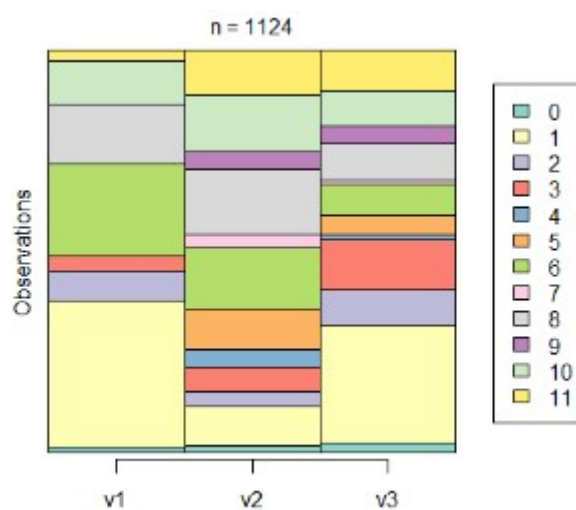


Figura 4.47: Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, distribuzione delle sequenze distinte

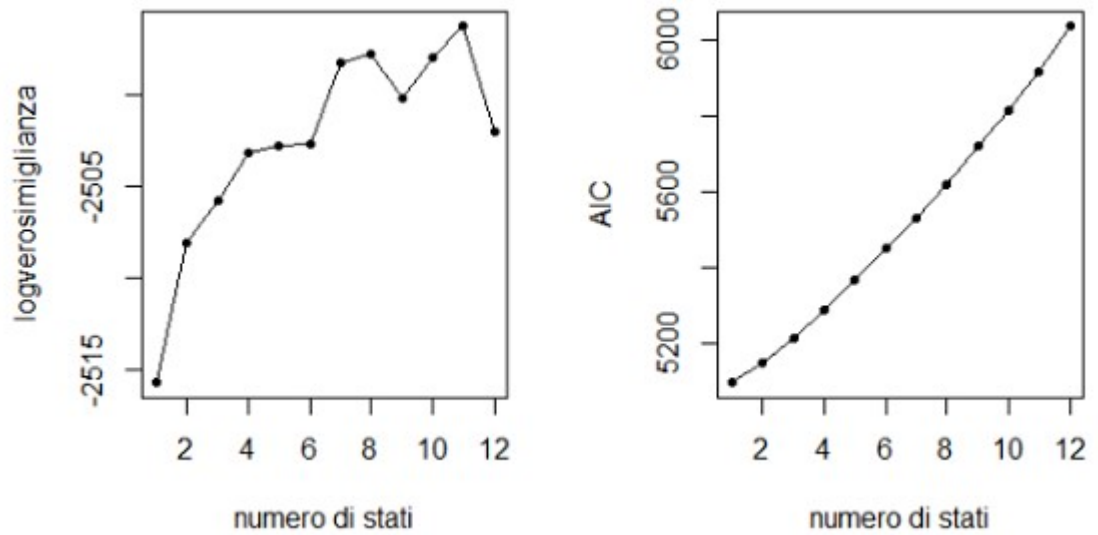


Figura 4.48: Notturmo op. 15 n. 2 in fa diesis minore, F. Chopin, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

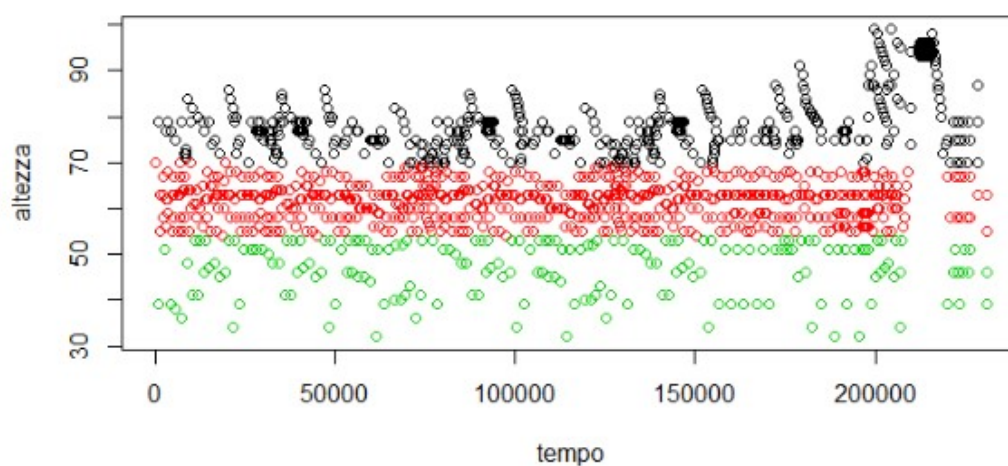


Figura 4.49: Notturmo op. 9 n. 2, F. Chopin, tempo e altezza delle note

12. Notturmo op. 9 n. 2 in mi bemolle maggiore, F. Chopin

Il brano che segue è ancora un Notturmo, questa volta nella tonalità di *mi bemolle maggiore*. La freccia che nel diagramma a barre circolare indica direzione della media angolare punta approssimativamente al numero 10, opposto al centro rispetto al 3 (Figura 4.50). Per il raggruppamento delle sequenze, ancora una volta è stato utilizzato K-means con le trasformazioni di tempo e altezza in $\log(\text{time}^{0.1})$, e $\log(\text{note}^{0.5})$. I risultati sono visibili in Figura 4.49. Le osservazioni presenti sono in numero simile a quelle del brano precedente, 1.299, distribuite in 1.255 istanti temporali, si veda la distribuzione delle note in Figura 4.51. Il numero di stati nel modello migliore secondo il criterio di Akaike è 1 (Figura 4.52).

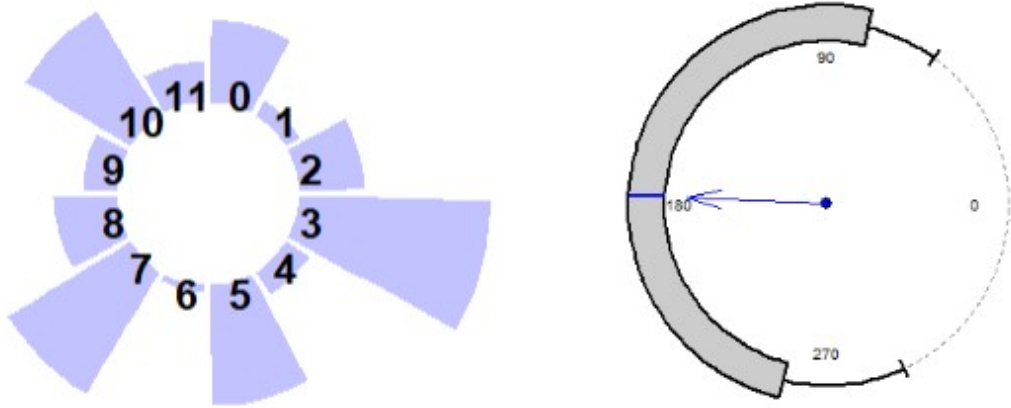


Figura 4.50: Notturmo op. 9 n. 2, F. Chopin, diagramma a barre e diagramma a scatola circolare

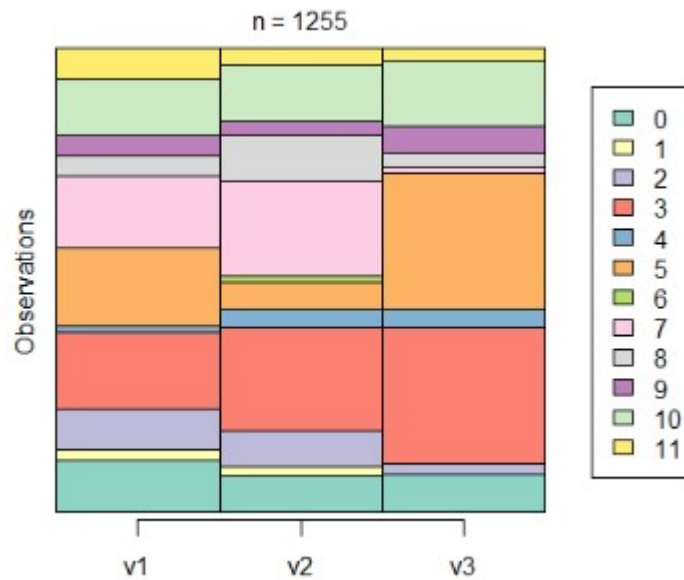


Figura 4.51: Notturmo op. 9 n. 2, F. Chopin, distribuzione delle sequenze distinte

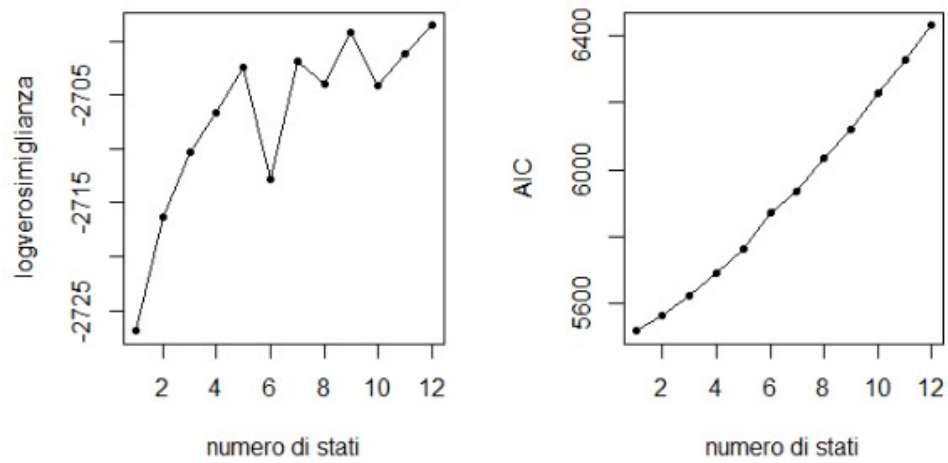


Figura 4.52: Notturmo op. 9 n. 2 , F. Chopin, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

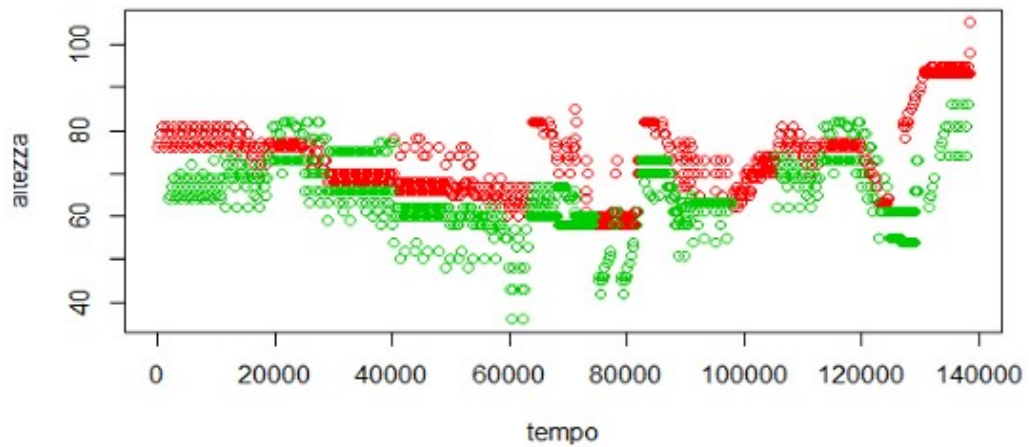


Figura 4.53: The snow is dancing, C. Debussy, tempo e altezza delle note

13. C. Debussy, The snow is dancing

La seguente è un'opera di Debussy: si tratta di uno dei primi pezzi minimalisti nella storia della musica. Le sequenze simultanee sono due, già distinte nel file originale (si veda la rappresentazione in Figura 4.53). Il numero di osservazioni è 1.389 per 1.181 valori della variabile tempo. La tonalità di riferimento è *re minore*, anche se in realtà, la musica di questo autore viene definita politonale; la distribuzione delle note si può osservare in Figura 4.54 e 4.55. Il criterio di Akaike indica che il modello per cui l'adattamento risulta migliore ha 2 stati latenti (Figura 4.56).

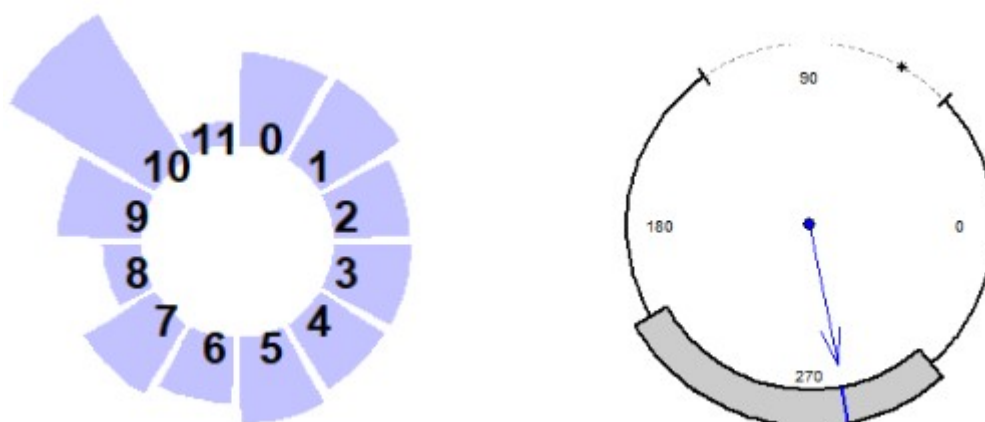


Figura 4.54: The snow is dancing, C. Debussy, diagramma a barre e diagramma a scatola circolare

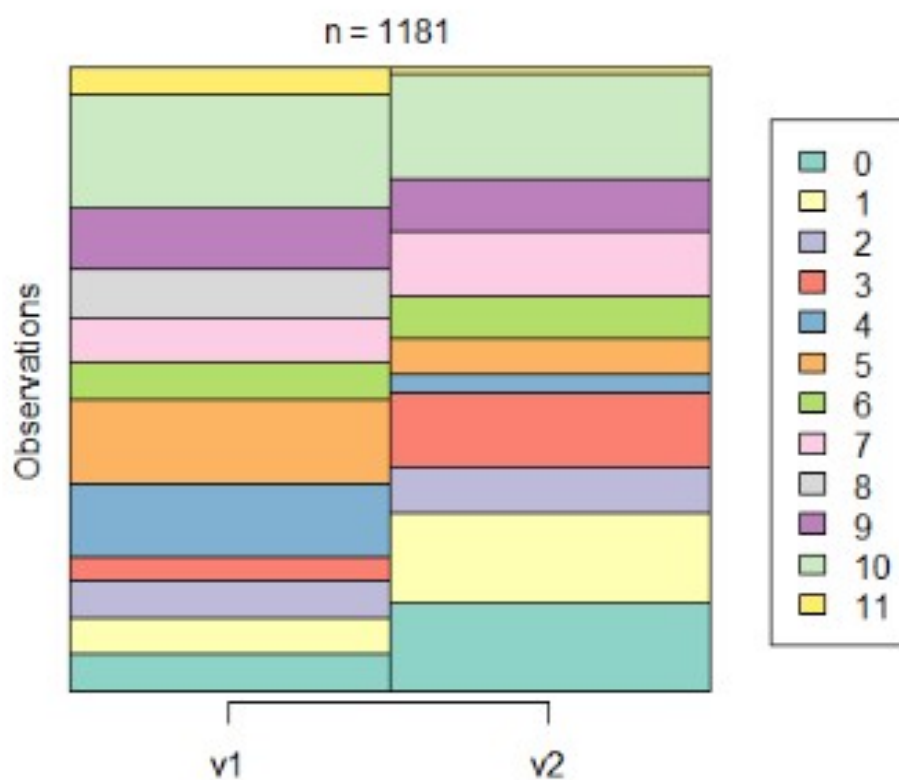


Figura 4.55: The snow is dancing, C. Debussy, distribuzione delle sequenze distinte

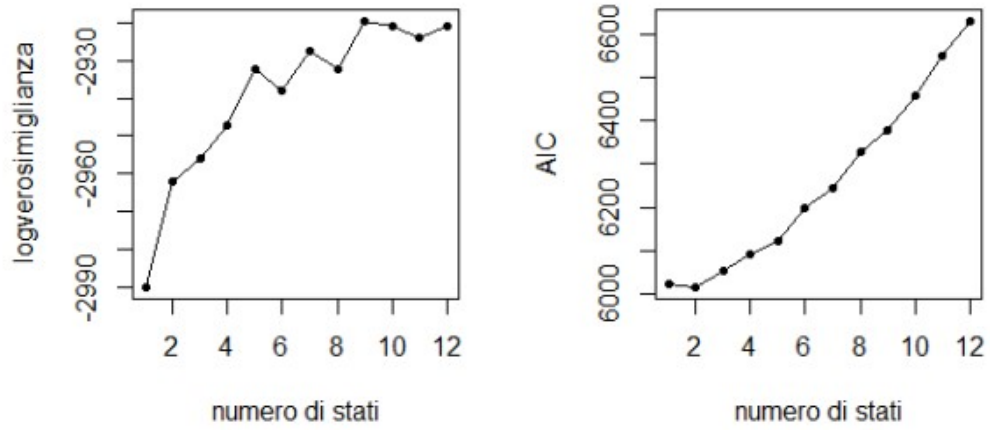


Figura 4.56: The snow is dancing, C. Debussy, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

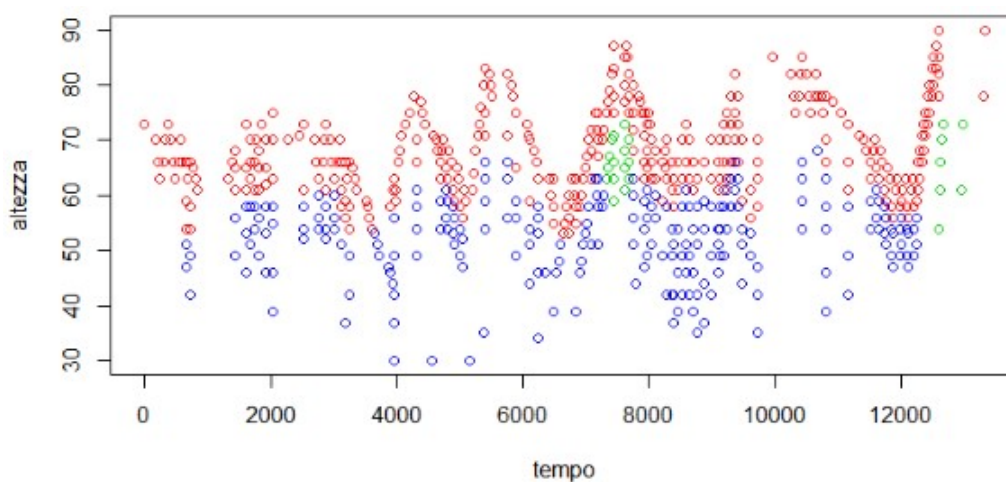


Figura 4.57: La fille au cheveux de lin, C. Debussy, tempo e altezza delle note

14. La fille au cheveux de lin, C. Debussy

Il secondo brano di C. Debussy preso come esempio è “La fille au cheveux de lin”. La tonalità a cui fa capo è *sol bemolle maggiore*; in questo caso il diagramma a scatola circolare non fornisce un’informazione aderente a questo dato (Figura 4.58). Si hanno due sequenze parallele (si vedano i colori utilizzati nel grafico 4.57 e le distribuzioni di frequenza per le linee distinte in Figura 4.59), per un totale di 582 osservazioni. È il brano più breve fra quelli analizzati, consta di soli 252 valori distinti per la variabile tempo. Come in quello precedente, il modello che, secondo il criterio di Akaike fornisce un migliore adattamento, prevede due stati nascosti (Figura 4.60).

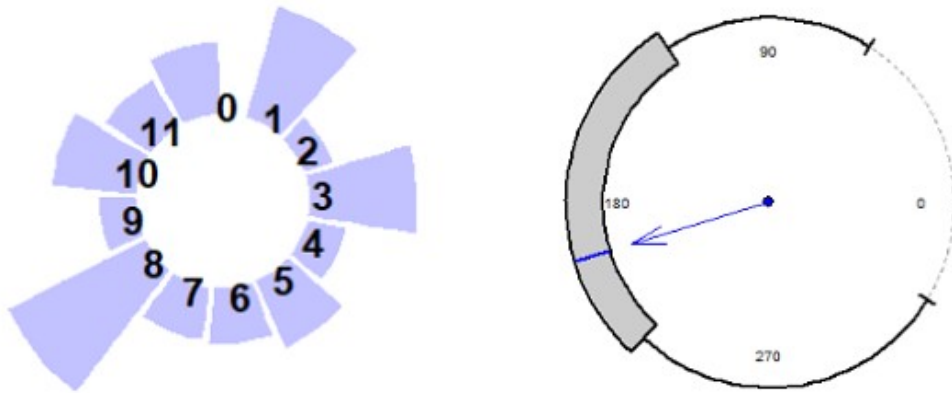


Figura 4.58: La fille au cheveux de lin, C. Debussy diagramma a barre e diagramma a scatola circolare

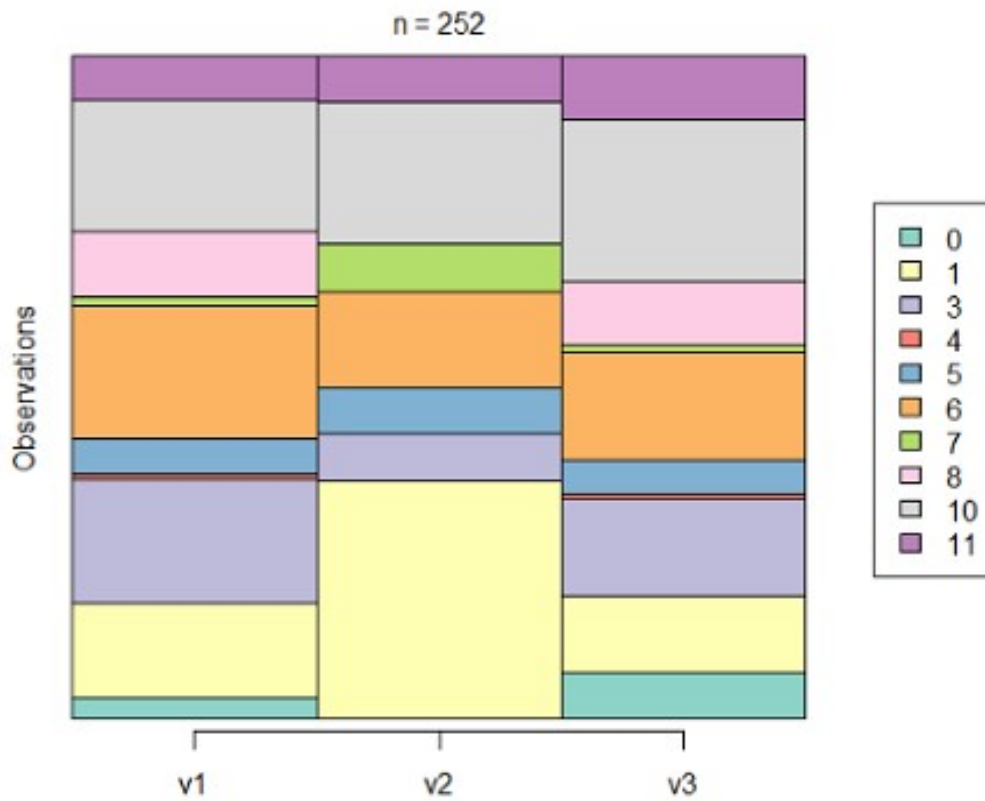


Figura 4.59: La fille au cheveux de lin, C. Debussy, distribuzione delle sequenze distinte

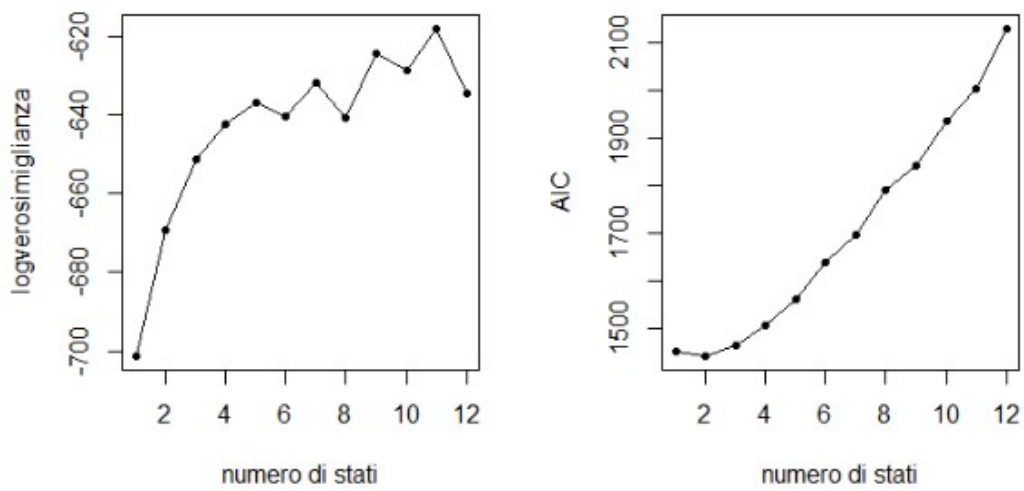


Figura 4.60: La fille au cheveux de lin, C. Debussy, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

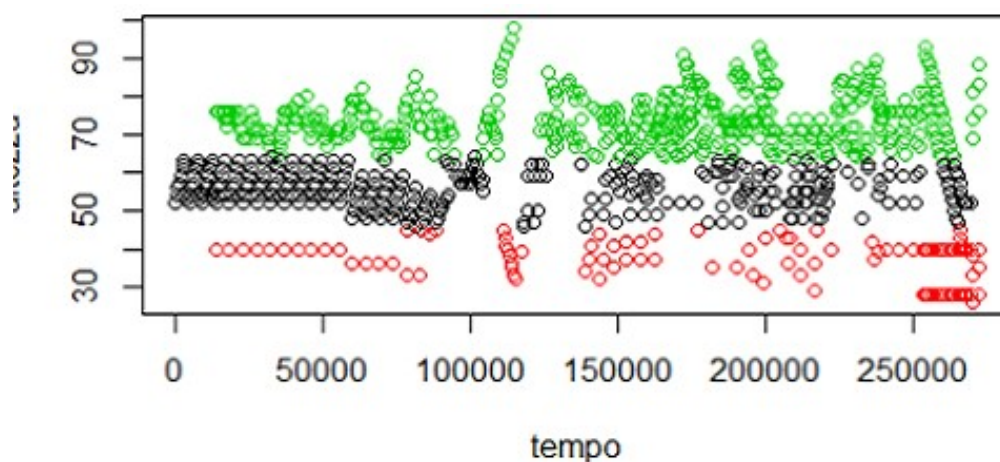


Figura 4.61: Danza in ritmo bulgaro n. 1, B. Bartok, tempo e altezza delle note

Danza in ritmo bulgaro n. 1, B. Bartok

L'ultima composizione analizzata, che è anche la più recente (risale agli anni '40 del Novecento), è una danza di Bartok per pianoforte. La melodia è di ispirazione popolare. Anche per questo breve brano parlare di tonalità risulta non del tutto appropriato. Bartok infatti fu uno dei principali esponenti della musica neo-modale; il "modo" di questo pezzo è il *mi frigio*. La freccia del diagramma a scatola circolare punta verso la direzione opposta, come nei brani precedenti, in cui è presente un forte senso di tensione, (Figura 4.62). In questo caso le osservazioni sono 940 e corrispondono a 520 diversi istanti temporali. Per individuare e separare le 3 sequenze parallele, è stato utilizzato l'algoritmo delle K-medie e le trasformazioni $\log(\text{time}^{0.4})$, e $\log(\text{note}^{1.135})$. Il risultato della suddivisione è soddisfacente, si osservino i colori del grafico di dispersione riportato in Figura 4.61. La distribuzione di frequenza delle note per ciascuna delle linee sovrapposte è rappresentata graficamente in Figura 4.63. Il miglior adattamento si ottiene, per il criterio di Akaike, con un solo stato latente (Figura 4.64).

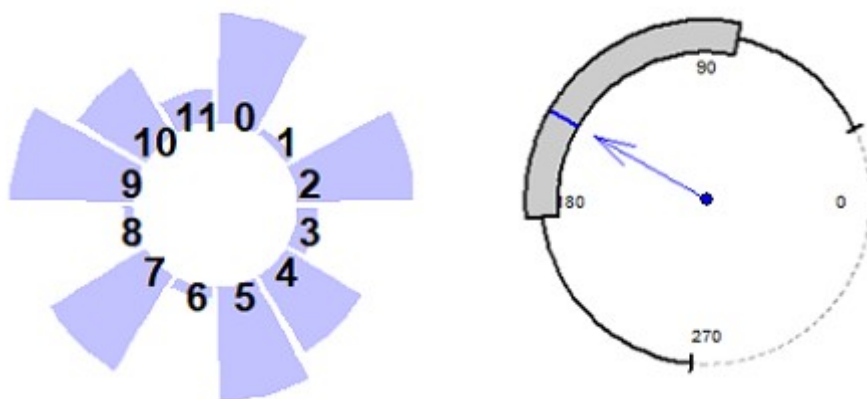


Figura 4.62: Danza in ritmo bulgaro n. 1, B. Bartok, diagramma a barre e diagramma a scatola circolare

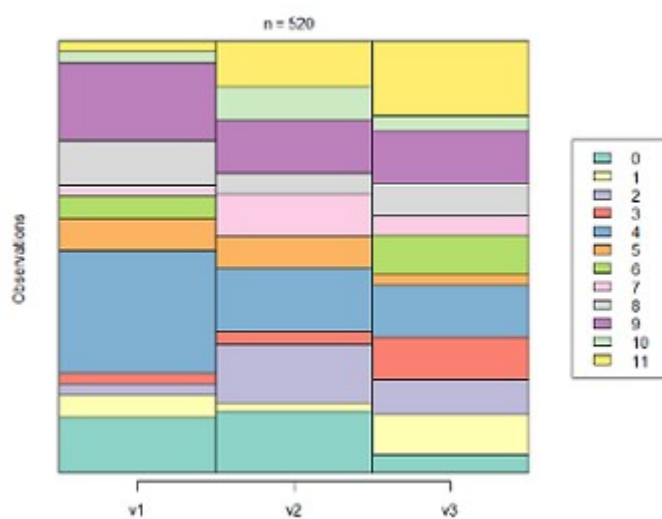


Figura 4.63: Danza in ritmo bulgaro n. 1, B. Bartok, distribuzione delle sequenze distinte

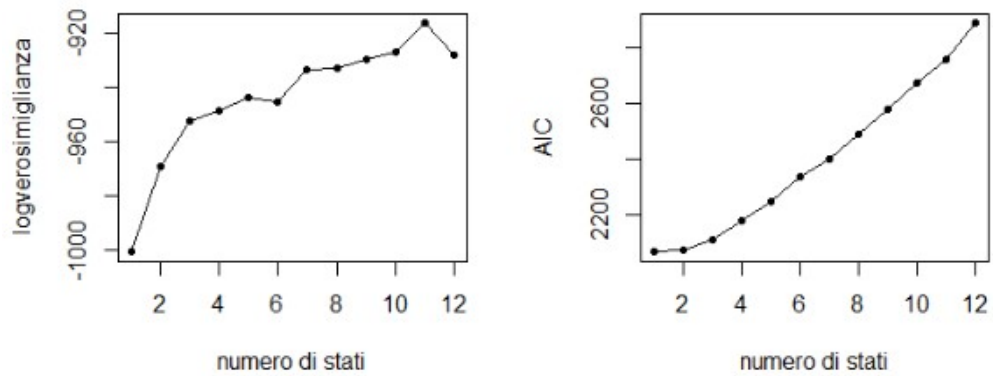


Figura 4.64: Danza in ritmo bulgaro n. 1, B. Bartok, log-verosimiglianza massimizzata e AIC al variare del numero di stati nascosti stimati

4.4 Calcolo delle similarità

In tabella 4.4 sono sintetizzati i risultati relativi all'adattamento dei modelli di Markov nascosti ottenuti nella prima fase dell'analisi. A margine, si può notare per esempio, che per i brani Romantici il numero di stati latenti è mediamente superiore, mentre i brani aleatori nello stile di Mozart, come gli altri di durata inferiore, sembrano avere una struttura latente più semplice. Inoltre in corrispondenza degli stessi compositori, si hanno lo stesso numero di stati. (Si vedano i due brani di Debussy, i brani "pseudomozartiani", e i brani di Chopin). L' successivo è il confronto tra brani. Come spiegato nel Capitolo 3, si è scelto di misurare le diversità tra i modelli basandosi sulla divergenza di Kullback Leibler. Innanzitutto, ciascuno dei brani è stato riportato ad una tonalità comune, eccezione fatta per il pezzo atonale di Schoenberg. Questa operazione ha come risultato il fatto che una volta ri-stimati i modelli, le nuove matrici di emissione risultano ruotate, mentre le matrici di transizione rimangono invariate. In linea con la proprietà di invarianza rispetto a riparametizzazioni della funzione di verosimiglianza, i valori della log-verosimiglianza nei modelli originali e in quelli relativi ai brani trasportati, è, al netto di qualche lieve approssimazione, la medesima. È stato effettuato questo controllo per avere la garanzia che la stima ottenuta in seguito alla trasformazione non fosse associata a delle soluzioni di massimo locale per la funzione di verosimiglianza. Si veda la Tabella 4.4.

Brano	Tonalità	n.stati (AIC)	l dati originali	l dati trasportati
Pseudo-Mozart 1	Re maggiore	1	-1051.1138	-1051.1137
Pseudo-Mozart 2	Re \sharp maggiore	1	-935.7443	-935.7443
Pseudo-Mozart 3	Re maggiore	1	-991.4167	-991.4167
Cantata 140, J. S. Bach	Mib maggiore	6	-9835.6108	-9843.8166
Preludio e Fuga BWV 884, J. S. Bach	Sol maggiore	5	-5919.8184	-5906.3895
Quartetto per archi op.12, F. Schubert	Do maggiore	9	-19567.2429	-19406.3441
Improvviso op. 90 n.2, F. Schubert	Mib maggiore	11	-6911.8485	-6914.6649
Arabesque op.18, R. Schumann	Do maggiore	3	-5142.7292	-5151.2386
Notturmo op.15 n.2, F. Chopin	Fa \sharp minore	1	-2515.6794	-2515.6793
Notturmo op.9 n.2, F. Chopin	Mib maggiore	1	-2726.9282	-2726.9282
The snow is dancing, C. Debussy	Re minore	2	-2961.1967	-2968.5272
La fille au cheveux de lin, C. Debussy	Solbmaggiore	2	-669.3505	-668.6208
Tre pezzi per piano, n.1, A. Schoenberg	?	1	-1442.6678	-1442.6678
Danza in ritmo bulgaro n.1, B. Bartok	Mi	1	-1000.5389	-1000.5389

Tabella 4.4. Modelli adattati e valore della log-verosimiglianza

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]	[,15]
[1,]	0.000	0.022	1.885	2.472	2.450	0.072	2.399	-0.237	-0.241	0.564	2.160	1.966	1.002	0.913	1.874
[2,]	-0.076	0.000	2.401	2.383	2.387	0.210	2.568	-0.244	-0.369	0.591	2.353	2.299	1.047	1.035	2.295
[3,]	-1.520	-1.581	0.000	0.213	0.645	-1.308	0.131	-1.640	-1.688	-1.243	0.054	0.185	-0.808	-0.687	0.137
[4,]	-1.450	-1.414	0.118	0.000	0.355	-1.522	0.328	-1.781	-1.762	-1.186	0.009	0.119	-0.701	-0.817	0.158
[5,]	-1.727	-1.801	-0.065	0.022	0.000	-1.716	0.091	-1.844	-1.824	-1.201	-0.014	-0.085	-0.892	-0.698	0.071
[6,]	-0.165	-0.004	1.722	1.592	1.859	0.000	2.039	-0.232	-0.250	0.322	1.769	2.372	0.779	0.623	1.742
[7,]	-1.484	-1.461	0.179	0.036	0.047	-1.371	0.000	-1.604	-1.606	0.690	0.242	-0.027	-0.670	-0.843	0.128
[8,]	0.127	0.149	2.150	2.095	2.192	0.293	2.102	0.000	-0.056	0.643	2.211	2.160	1.021	1.059	2.133
[9,]	0.197	0.233	0.233	2.025	2.036	0.343	2.112	0.070	0.000	0.732	2.117	2.257	1.087	1.046	2.124
[10,]	-0.384	-0.355	1.429	1.611	1.447	-0.315	-0.660	-0.550	-0.652	0.000	1.586	1.385	0.394	0.677	1.639
[11,]	-1.475	-1.536	0.422	0.189	0.339	-1.338	0.208	-1.606	-1.685	-1.066	0.000	0.098	-0.767	-0.578	0.398
[12,]	-1.440	-1.278	0.316	0.339	0.168	-1.428	0.455	-1.533	-1.605	-1.033	0.327	0.000	-0.680	-0.626	-0.826
[13,]	-0.642	-0.779	0.944	0.900	0.909	-0.698	0.965	-0.733	-0.804	-0.285	1.332	1.033	0.000	0.044	1.215
[14,]	-1.118	-1.355	2.594	2.335	2.501	-1.174	2.654	-1.199	-2.062	-0.583	1.147	2.056	0.250	0.000	1.794
[15,]	-1.508	-1.428	0.271	-0.050	0.032	-1.252	0.290	-1.590	-1.563	-1.563	-1.038	0.105	1.093	-0.711	0.000

Figura 4.65: Risultati delle divergenze calcolate mediante simulazione

La divergenza di Kulback-Leibler è stata calcolata con un procedura di simulazione. Definiamo ψ e ξ due generici modelli stimati, R il numero di replicazioni, X_t la sequenza multivariata generata dal modello ψ , n la lunghezza totale della serie. Il risultato finale per ogni confronto deriva dal calcolo della quantità

$$\Delta(\psi, \xi) = 1/R(1/n \Pr(X_t, \psi)(\log \Pr(X_t, \psi) - \log \Pr(X_t, \xi))) \quad (4.1)$$

Fissando lo stesso seme, per ogni modello ψ sono state generate un numero di sequenze univariate pari al numero di sequenze simultanee del campione originale utilizzando di volta in volta una matrice di emissione diversa, questo mediante la funzione `simulate_hmm`. La lunghezza delle sequenze è stata fissata ponendo pari a 400 il numero di istanti temporali. Per calcolare le quantità $\Pr(X_t, \psi)$ e $\Pr(X_t, \xi)$ sono stati costruiti degli oggetti “hmm” utilizzando le sequenze simulate e le matrici dei modelli ψ e ξ stimati in origine. Nello specifico, quando il numero di matrici di emissione del modello ξ di confronto non era lo stesso sono state utilizzate solo le prime matrici, se il numero era superiore, mentre se il numero era inferiore, sono state usate tutte, e ripetute in modo ciclico. La procedura è stata reiterata 100 volte. I risultati sono riportati in Figura 4.65.

4.4.1 Soluzione “naive” e soluzione alternativa

Si noti che in corrispondenza di modelli che presentano un numero diverso di stati, la quantità $\log \Pr(X_t, \psi) - \log \Pr(X_t, \xi)$ è negativa. Il numero di stati nei modelli originali, infatti è stato scelto mediante il criterio di Akaike, utilizzando una penalizzazione per modelli più complicati. Nella stima della divergenza di Kullback-Leibler il termine della penalità non compare, e pertanto modelli con un numero di stati superiore tendono comunque ad avere un adattamento migliore. Non si ha una “perdita di informazione” ma un “guadagno”. Sono stati considerati perciò i valori assoluti delle quantità ottenute ed è stata calcolata per ogni coppia di unità una media tra $|\Delta(\psi, \xi)|$ e $|\Delta(\xi, \psi)|$. La matrice delle distanze ricavata in questo modo è riportata in Figura 4.66; questa è stata il punto di partenza per

applicare la tecnica dello Scaling Multidimensionale. Si veda anche la Figura 4.67, una mappa dei punti caldi, nella quale il grado di somiglianza per ogni coppia di unità viene indicato con dei colori, scelti su una scala continua che va dal bianco al rosso intenso. A margine della figura è stato costruito un dendrogramma con il metodo del legame completo, che connette le unità più simili, come ad esempio la 13 e la 14 (i due brani di Chopin), e permette di individuare alcuni gruppi. Il calcolo della divergenza è stato eseguito una seconda volta, per provare a superare il problema della negatività dei valori ottenuti, aggiungendo una penalità per il numero di parametri, analoga a quella del criterio di Akaike nella formula 4.4, cioè

$$M \times M + M \times S. \quad (4.2)$$

4.5 Risultati ottenuti

Si è provato a proiettare l'insieme dei dati sia in uno spazio bidimensionale che tridimensionale. Si vedano i grafici in Figura 4.68 e in Figura 4.69 per i risultati relativi alla prima soluzione. La scala dei colori è in relazione con il periodo storico in cui i brani sono stati composti, e va dal blu per i brani più antichi, fino al giallo, per quelli più recenti. Ricostruendo l'aggregato dei dati nel primo caso si raggiunge un valore di GOF (Goodness of fit) pari a 0.82, mentre aggiungendo anche la terza dimensione il risultato sale a 0.89. Osservando le mappe costruite si possono notare alcuni curiosi aspetti: per prima cosa brani dello stesso compositore tendono ad essere vicini. Inoltre le differenze osservate con la modellazione possono trovare una spiegazione legata alla geografia e alla storia: i brani di epoca romantica, dei compositori di area tedesca si collocano nella parte sinistra del grafico, mentre le opere di Bartok e Chopin, insieme a quelle di Mozart e Schoenberg, vissuti nell'area di influenza dell'Impero Austro-Ungarico si collocano a destra. Ancora, osservando il grafico in due dimensioni, si può notare che il "baricentro" per i brani di Debussy è vicino a quello dei brani di Chopin: i due compositori non si conobbero ma vissero entrambi a Parigi. Questa interpretazione sembra avere un senso, anche alla luce dei risultati di un lavoro di analisi simile a questo, in cui sono state messe a confronto delle melodie popolari (in quel caso serie univariate) provenienti da diverse aree d'Europa, sempre utilizzando gli HMM [3]: i modelli riuscivano a far emergere le diversità tra i pezzi di diversa origine geografica. Tagliando lungo la bisettrice il piano xy è poi possibile osservare da una parte i brani veloci, dall'altra quelli con un carattere più pacato. I risultati relativi al calcolo delle divergenze con la penalizzazione sono riportati nelle figure 4.70 e 4.71. Con questa seconda soluzione, facendo riferimento alla rappresentazione in due dimensioni, si nota che i due brani più "avanguardisti", quelli ricchi di dissonanze, di Bartok Schoenberg sono molto ravvicinati. Lo stesso vale per i brani

[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]	[1,9]	[1,10]	[1,11]	[1,12]	[1,13]	[1,14]	[1,15]
0.000	0.049	1.703	1.961	2.089	0.119	1.941	0.182	0.219	0.474	1.818	1.703	0.822	1.015	1.691
0.049	0.000	1.991	1.899	2.094	0.107	2.015	0.196	0.301	0.473	1.944	1.789	0.913	1.195	1.862
1.703	1.991	0.000	0.166	0.355	1.515	0.155	1.895	0.960	1.336	0.238	0.250	0.876	1.641	0.204
1.961	1.899	0.166	0.000	0.188	1.557	0.182	1.938	1.894	1.398	0.099	0.229	0.800	1.576	0.104
2.089	2.094	0.355	0.188	0.000	1.788	0.069	2.018	1.930	1.324	0.177	0.126	0.900	1.600	0.052
0.119	0.107	1.515	1.557	1.788	0.000	1.705	0.262	0.297	0.318	1.553	1.900	0.739	0.898	1.497
1.941	2.015	0.155	0.182	0.069	1.705	0.000	1.853	1.859	0.675	0.225	0.241	0.818	1.749	0.209
0.182	0.196	1.895	1.938	2.018	0.262	1.853	0.000	0.063	0.597	1.909	1.847	0.877	1.129	1.862
0.219	0.301	0.960	1.894	1.930	0.297	1.859	0.063	0.000	0.692	1.901	1.931	0.945	1.554	1.844
0.474	0.473	1.336	1.398	1.324	0.318	0.675	0.597	0.692	0.000	1.326	1.209	0.339	0.630	1.601
1.818	1.944	0.238	0.099	0.177	1.553	0.225	1.909	1.901	1.326	0.000	0.213	1.049	0.862	0.718
1.703	1.789	0.250	0.229	0.126	1.900	0.241	1.847	1.931	1.209	0.213	0.000	0.857	1.341	0.466
0.822	0.913	0.876	0.800	0.900	0.739	0.818	0.877	0.945	0.339	1.049	0.857	0.000	0.147	1.154
1.015	1.195	1.641	1.576	1.600	0.898	1.749	1.129	1.554	0.630	0.862	1.341	0.147	0.000	1.253
1.691	1.862	0.204	0.104	0.052	1.497	0.209	1.862	1.844	1.601	0.718	0.466	1.154	1.253	0.000

Figura 4.66: Matrice delle distanze calcolate simmettizzando i risultati della divergenza di Kullback- Leibler

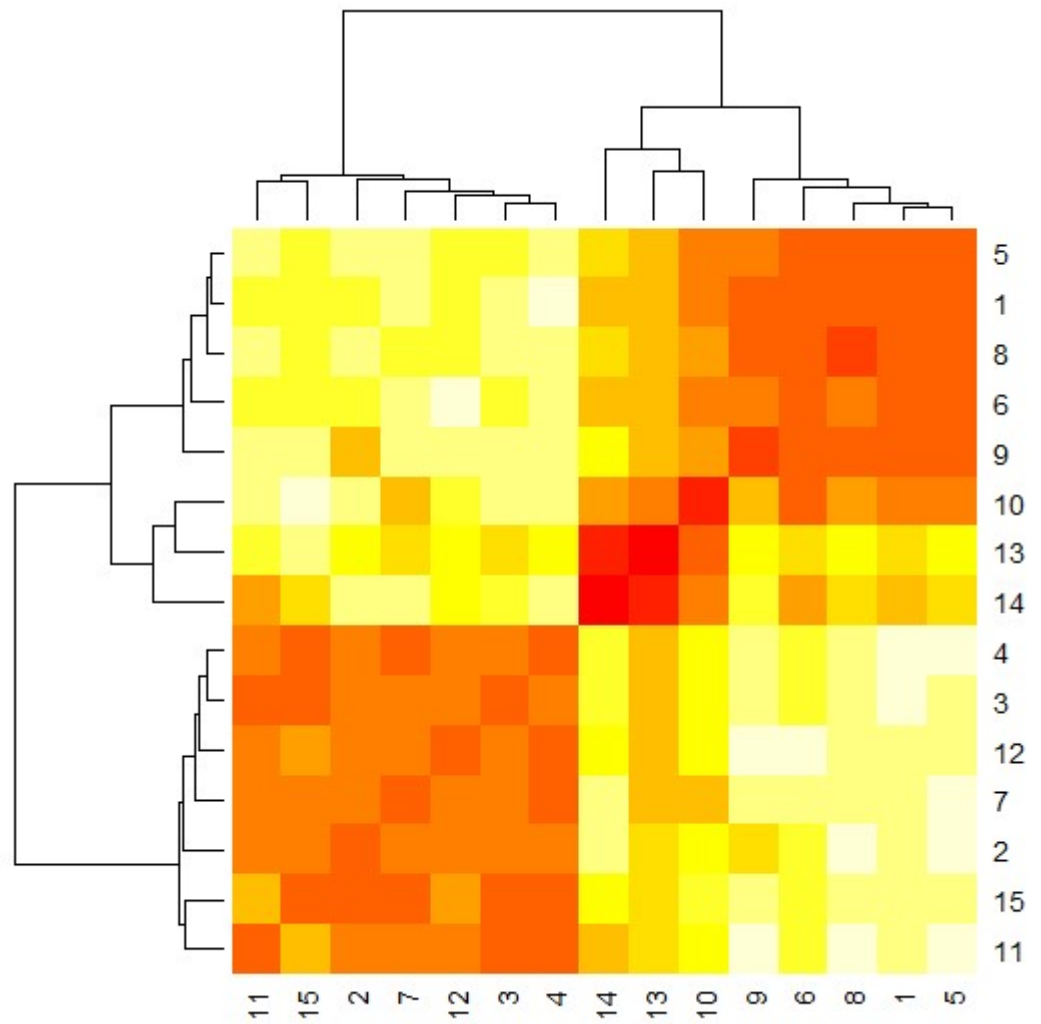


Figura 4.67: Mappa dei punti caldi calcolata a partire dalla matrice delle distanze, con dendrogramma. La numerazione dei brani segue quella presentata all'interno di questo Capitolo.

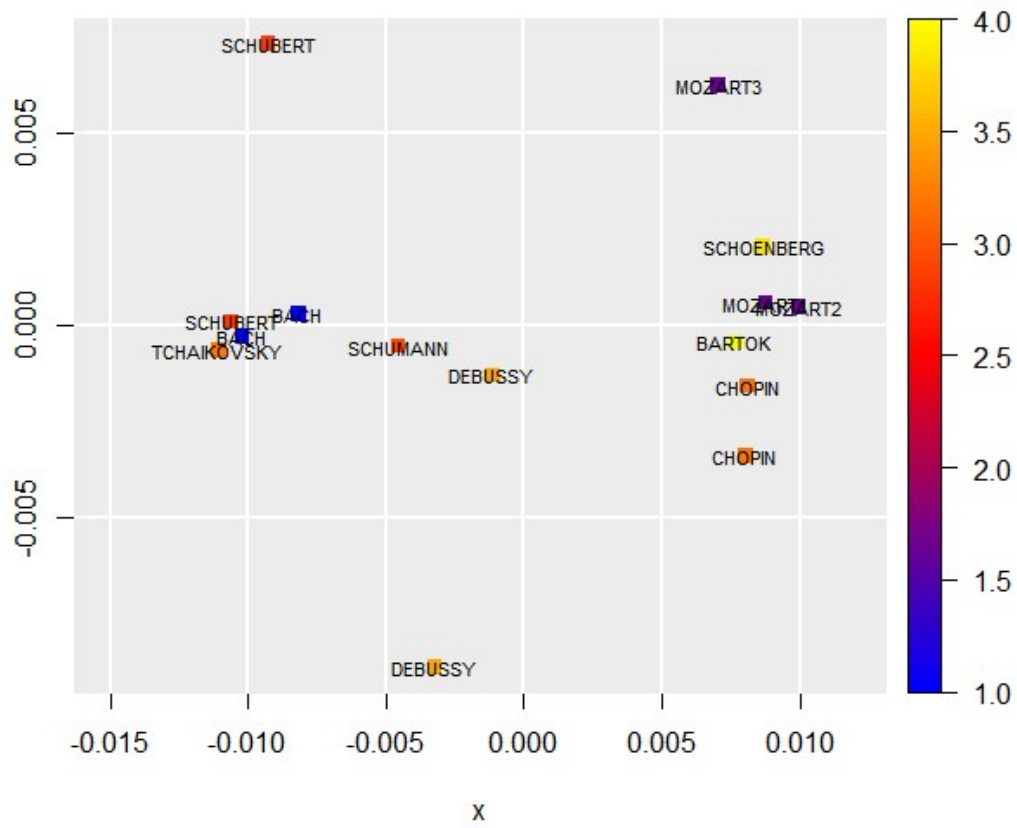


Figura 4.68: Scaling in 2D, soluzione “naive”

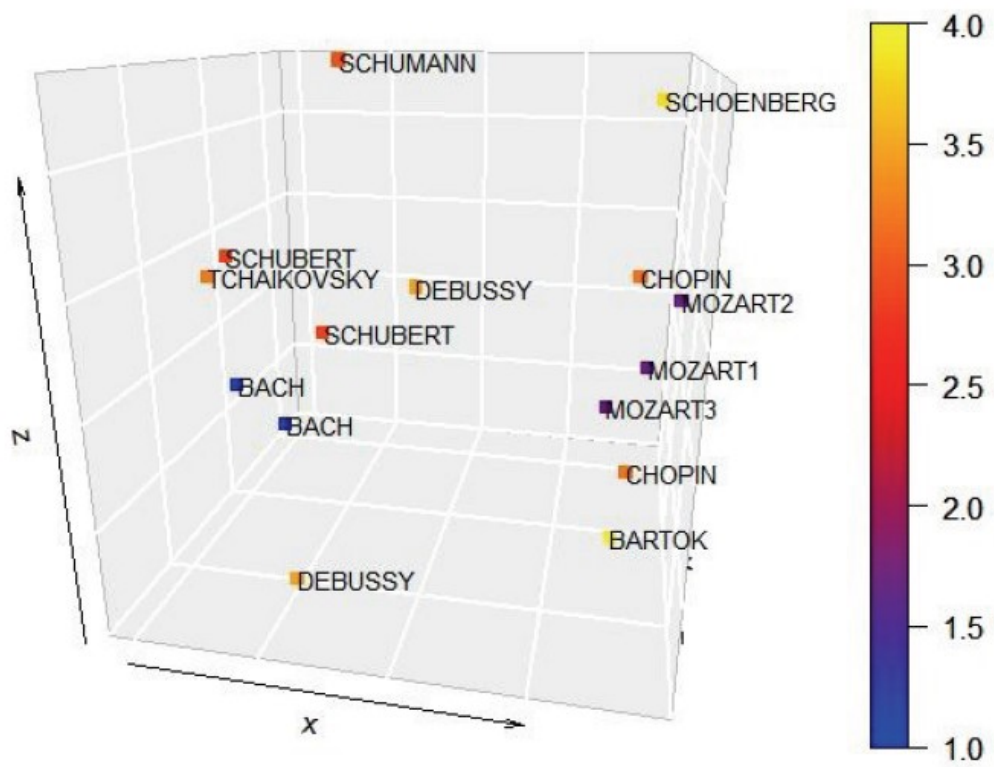


Figura 4.69: Scaling in 3D, soluzione “naive”

dei Romantici Schubert e Schumann (i cui modelli adattati hanno 9,11 e 3 stati nascosti), che nel grafico precedente si collocavano in modo sparso.

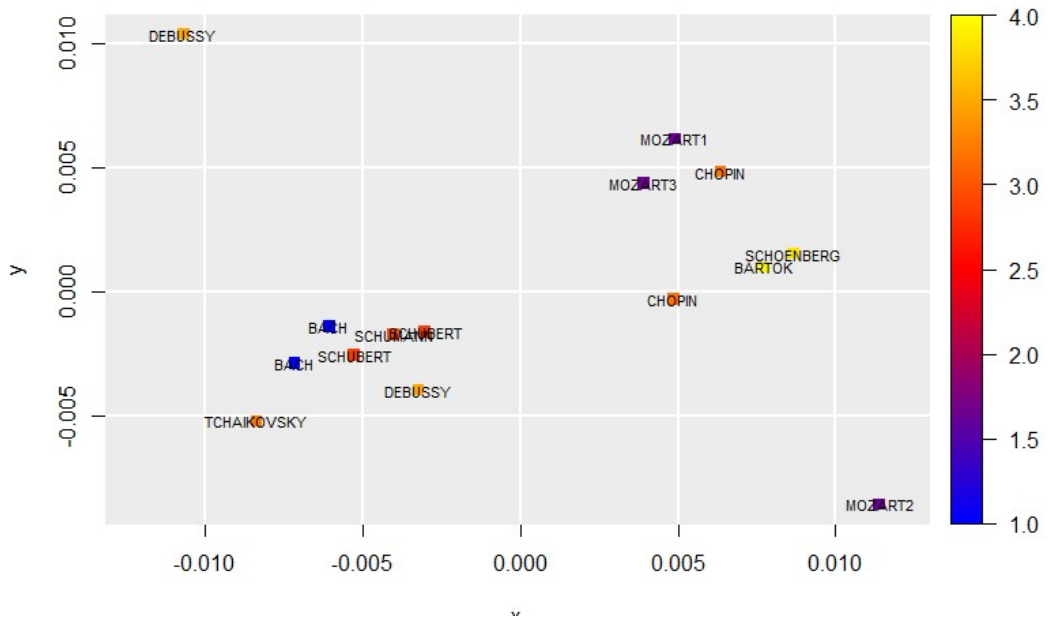


Figura 4.70: Scaling in 2D, a partire dalle divergenze calcolate con la penalizzazione del valore della probabilità

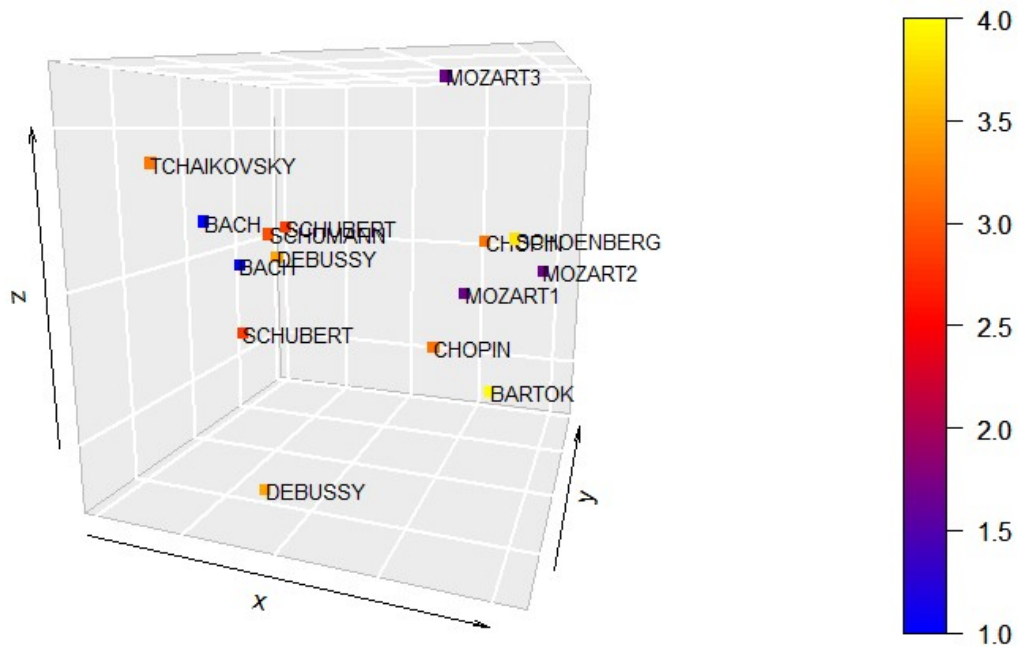


Figura 4.71: Scaling in 3D, a partire dalle divergenze calcolate con la penalizzazione del valore della probabilità

4.6 Un confronto con t-SNE

In ultima istanza, è stato effettuato un tentativo con la tecnica t-SNE, molto potente, quando l'obiettivo dell'analisi è ridurre la dimensione dei dati e proiettarli in uno spazio "ridotto" per poter notare analogie e similarità. Questa tecnica non nasce però per l'analisi di sequenze temporali, né per di dati categoriali, (si veda il Capitolo 3), tuttavia viene utilizzata per il riconoscimento di pattern, per esempio nelle immagini. I nostri dati sono categoriali, ma codificati con numeri, pertanto, in via esplorativa, per quanto inadatta, e solo per poter aver un confronto con i risultati ottenuti sopra, si tenta di forzare l'algoritmo e di testarne l'efficacia (anche in letteratura ci sono degli esemp in cui questo, con alcune modifiche viene applicato a serie temporali [13]). Rispetto alle altre tecniche di analisi multivariata, come le componenti principali (in cui vengono utilizzate delle combinazioni che danno peso diverso alle variabili concomitanti), uno dei vantaggi di t-SNE sta nel fatto che per ogni unità statistica, ciascuno dei record contribuisce a definire la configurazione dell'insieme nello spazio ridotto. Nell'ordine, sono stati generati da ciascun modello 10 nuovi brani, di questi è stata presa in considerazione la serie degli stati nascosti, dopodiché è stato richiesto all'algoritmo di trovare una rappresentazione prima in due e poi in tre dimensioni, per le unità statistiche. È stato scelto di simulare dei nuovi dati, e non di utilizzare solo quelli a disposizione, perché in questo modo non si hanno sequenze con dati mancanti, né è necessario stimare la sequenza latente. Con questo metodo, il risultato non è del tutto soddisfacente: brani generati dal medesimo modello si trovano vicini, tuttavia la disposizione sembra risentire unicamente del numero di stati nascosti stimati nei modelli originali. Nei grafici delle Figure 4.72 e 4.73 le sequenze generate dal medesimo modello sono indicate con la stessa lettera. Le lettere a e g fanno riferimento alle simulazioni da i due brani di Bach, la lettera b è usata per le simulazioni da Tchaikovsky, la c, la d e la e per i brani di Mozart, la lettera h fa riferimento a Schoenberg, la i e la j a Schubert, la lettera k è usata per le simulazioni dal brano di Schumann, l e m per i Notturmi di Chopin, n e o per Debussy e p per Bartok. Fanno parte del cluster di forma circolare-sferica tutte le simulazioni dai modelli con un solo stato nascosto. Inoltre brani dello stesso autore non sono sempre connessi (si osservi il caso delle simulazioni dai brani di Bach).

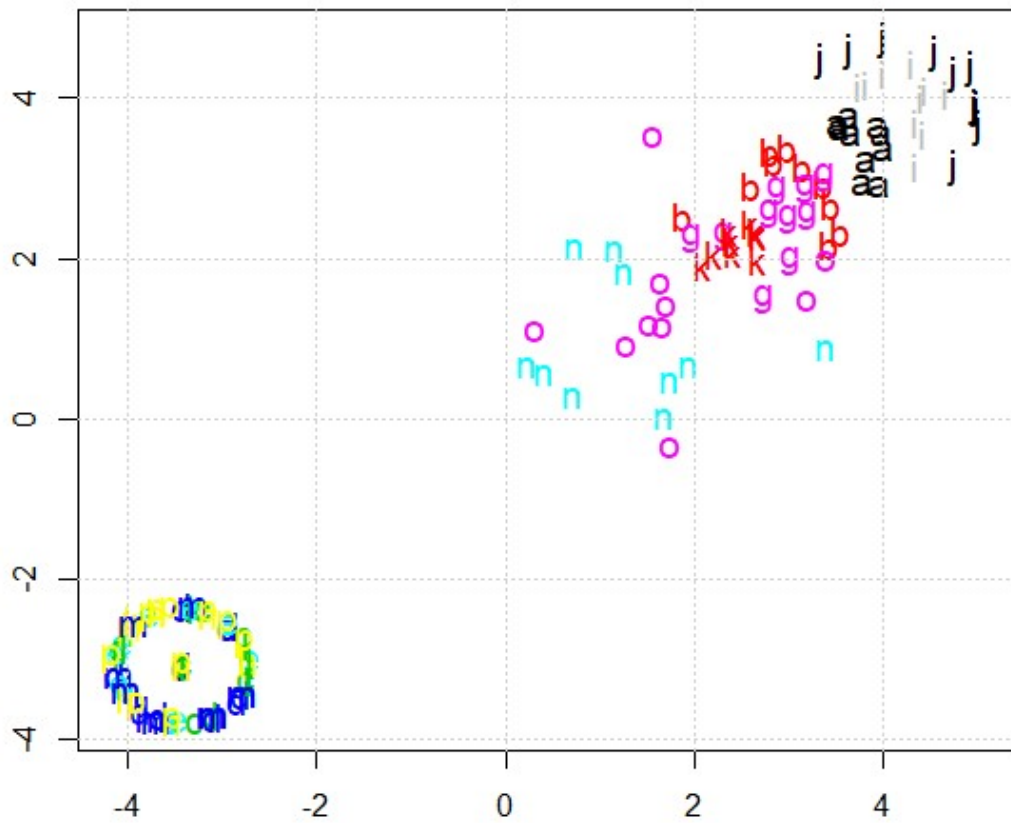


Figura 4.72: Proiezione dei risultati dell'algorithm t-SNE in uno spazio bidimensionale

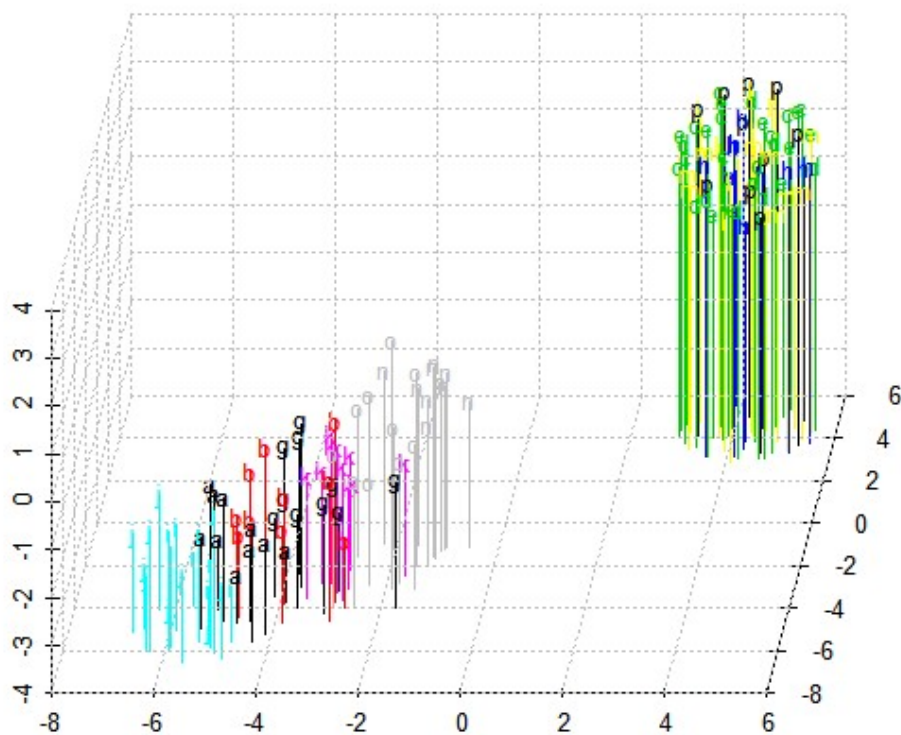


Figura 4.73: Proiezione dei risultati dell'algoritmo t-SNE in uno spazio tridimensionale

Capitolo 5

Conclusioni

In sintesi, gli obiettivi iniziali sembrano essere stati raggiunti, il metodo consente di mettere in relazione le unità statistiche in modo coerente con quanto ci si poteva aspettare. Rimangono tuttavia alcune questioni in sospeso. In particolare, meriterebbero un approfondimento il comportamento della funzione di verosimiglianza e del criterio di Akaike, rispetto alla lunghezza delle sequenze. Come visto, l'impatto della penalità inserita nell'AIC, sembra essere più forte sui brani brevi. Per ricalibrare i modelli, tenendo conto di questo aspetto può essere opportuno considerare, in alternativa al criterio di Akaike, quello di informazione Bayesiano (BIC). Sarebbe interessante osservare come variano le stime della divergenza di Kullback-Leibler in funzione delle nuove scelte. Potrebbe essere il caso, altrimenti, di stimare i modelli su segmenti di brani di lunghezza paragonabile. Altri aspetti che per mancanza di tempo non sono stati sviluppati sono, ad esempio, l'utilizzo di processi di ordine superiore, accennati all'Capitolo 2; in questo caso è necessario usare delle sequenze sufficientemente ampie, per il gran numero di parametri da stimare. O ancora si potrebbe prendere in considerazione la possibilità di includere delle covariare nel modello (la variabile "velocity", codificata nei file MIDI, una variabile che indichi se il percorso della melodia sta salendo o scendendo, oppure la durata dei suoni). L'analisi poi potrebbe vertere anche sulla serie delle successioni accordali, in modo congiunto. Si pensa infine che si potrebbe ottenere un miglioramento sostanziale nel metodo, se fosse possibile recuperare anche l'informazione sull'ordinamento naturale dei suoni: con un modello di Markov nascosto che basato su una distribuzione circolare discreta per i dati osservati. Inoltre, sarebbe utile, al fine di verificare la stabilità dei risultati, introdurre nel campione un numero maggiore di brani. In questo lavoro se ne sono considerati soltanto quindici: non avendo a disposizione pezzi di codice già pronti e dovendo partire da zero, è stato un procedimento abbastanza lungo dispendioso all'inizio soltanto capire come poter ottenere dei dati e ricavare dai brani le sequenze distinte. Alcuni accorgimenti che possono essere presi in con-

siderazione se si volesse ampliare il *corpus* dei brani sono legati al modo in cui si possono separare le tracce, operando quando queste sono ancora in formato .wav. Si potrebbero provare infine altre tecniche diverse dallo scaling classico, come i metodi per matrici asimmetriche.

Appendice A

Codice R utilizzato

Estrapolazione del contenuto dei file MIDI, adattamento dei modelli markoviani, realizzazione dei grafici

```
#import delle librerie

library(tuneR)
library(seqHMM)
library(circular)
library(plotrix)
library(tibble)
library(ggplot2)

A=readMidi(file.choose())##Cantata BWV140 Bach
noteA=getMidiNotes(A)
T=length(unique(noteA[,1]))
matrA=noteA[,c(1,2,3,4,5)]
attach(noteA)
tempo=unique(time)
v1=v2=v3=v4=v5=v6=v7=v8=rep(NA,T)
for (j in 1:length(time)){
  for (i in 1:T){
    idx=tempo[i]
    if(noteA[j,1]==idx){
      if(noteA[j,3]=='2'){
        v1[i]=noteA[j,5]
        break}
      if(noteA[j,3]=='3'){
        v2[i]=noteA[j,5]
        break}
      if(noteA[j,3]=='4'){
```

```

        v3[i]=noteA[j,5]
        break}
    if(noteA[j,3]=='5'){
        v4[i]=noteA[j,5]
        break}
    if(noteA[j,3]=='6'){
        v5[i]=noteA[j,5]
        break}
    if(noteA[j,3]=='7'){
        v6[i]=noteA[j,5]
        break}
    if(noteA[j,3]=='8'){
        v7[i]=noteA[j,5]
        break}
    if(noteA[j,3]=='9'){
        v8[i]=noteA[j,5]
        break}
    }
    i=i+1
  }
  j=j+1}
matriceossA=cbind(tempo,v1,v2,v3,v4,v5,v6,v7,v8)
matriceossA[,2:9]=matriceossA[,2:9]%%12 #MODULO 12
seqdef(matriceossA[,2])->SEQ1
seqdef(matriceossA[,3])->SEQ2
seqdef(matriceossA[,4])->SEQ3
seqdef(matriceossA[,5])->SEQ4
seqdef(matriceossA[,6])->SEQ5
seqdef(matriceossA[,7])->SEQ6
seqdef(matriceossA[,8])->SEQ7
seqdef(matriceossA[,9])->SEQ8
#adattamento con numero diverso di stati
loglikhidden=rep(NA,12)
aic=rep(NA,12)
for (u in 1:12){
  HU<-build_hmm(observations = list(SEQ1,SEQ2,SEQ3,
  SEQ4,SEQ5,SEQ6,SEQ7,SEQ8), u,n_channels(8))
  fitted3<-fit_model(HU, em_step=F,global_step= T,
  local_step = F, threads=4)
  loglikhidden[u] =fitted3$logLik
  aic[u]=AIC(fitted3$model)
}
q=seq(1:12)

```

```

par(mfrow=c(1,2))
plot(q, (loglikhidden), ylab="log verosimiglianza",
     xlab="numero di stati", type='l')
points(q, (loglikhidden), pch=20)
plot(q, aic, ylab="AIC", xlab="numero di stati", type='l')
points(q, aic, pch=20)

#scelta di 6 stati
H<-build_hmm(observations = list(SEQ1,SEQ2,SEQ3,SEQ4,
SEQ5,SEQ6,SEQ7,SEQ8), 6, n_channels(8))
fittedA<-fit_model(H, em_step = F, global_step = T,
local_step = F)
# massimizzazione numerica globale (
#Grafici
plot(matrA[,1], matrA[,5], col=(noteA[,3]),
     xlab="tempo", ylab="altezza")
plot(matrA[,1], matrA[,5]\%\%12, col=(noteA[,3]),
     xlab="tempo", ylab="altezza")
s=as.vector(table(matrA[,5]\%\%12)/
sum(table(matrA[,5]\%\%12)))
barplot(table(matrA[,5]\%\%12)/
sum(table(matrA[,5]\%\%12)))
data=data.frame(
  id=seq(11,0),
  individual=paste(" ", seq(11,0), sep=""),
  value=rev.default(s)*100)
label_data=data
p = ggplot(data, aes(x=rev.default(as.factor(id)),
y=(value))) +
geom_bar(stat="identity", fill=alpha("blue",0.24))+
ylim(-10,40) +
theme_minimal() +
theme(
  axis.text = element_blank(),
  axis.title = element_blank(),
  panel.grid = element_blank(),
  plot.margin = unit(rep(-2,4), "cm")
) +
coord_polar(start = 90)+
geom_text(data=label_data, aes(x=rev.default(id)+1,
y=1,20, label=individual), color="black", fontface="bold"
, alpha=0.99, size=5.5, inherit.aes=FALSE)
plot(p)

```

```

circ=as.circular(table(matrA[,5]\%\%12))
mean.circular(circ)
CircularBoxplot(circ)
seqdef(matriceossA[,2:9])->SEQ
ssplot(SEQ, type = "I")
ssplot(SEQ)

#Esempio in cui le sequenze vanno divise
C=readMidi(file.choose())##Pseudo Mozart1
noteC=getMidiNotes(C)
matrC=noteC[,c(1,2,3,4,5)]
matrCbis=cbind((matrC[,1])^0.30,matrC[,5]^1.2)
km=kmeans((matrCbis), centers=3)
plot(matrC[,1], matrC[,5], col=km$cluster)
v1=v2=v3=rep(NA,T)
for(j in 1:length(time)){#ogniriga del dataset viene
  #associata ad un valore tempo e una voce,
  for (i in 1:T){
    idx=tempo[i]
    if(noteC[j,1]==idx){
      if(km$cluster[j]==1){
        v1[i]=noteC[j,5]
        break}
      if(km$cluster[j]==2){
        v2[i]=noteC[j,5]
        break}
      if(km$cluster[j]==3){
        v3[i]=noteC[j,5]
        break}
    }
    i=i+1
  }
  j=j+1
}

head(matrC)
matriceossC=cbind(tempo,v1,v2,v3)
matriceossC[,2:4]=matriceossC[,2:4]\%\%12
seqdef(matriceossC[,2])->SEQ1
seqdef(matriceossC[,3])->SEQ2
seqdef(matriceossC[,4])->SEQ3
[...]
#da questo punto in poi

```

```

#si procede come nel caso precedente
[...]
#la procedura va ripetuta per ciascun brano

#brani trasportari alla stessa tonalit\`a
matriceossAA=(matriceossA+0)\%\%12
matriceossBB=(matriceossB+8)\%\%12
matriceossCC=(matriceossC+1)\%\%12
matriceossDD=(matriceossD+1)\%\%12
matriceossEE=(matriceossE+0)\%\%12
matriceossGG=(matriceossG+8)\%\%12
matriceossHH=(matriceossH+0)\%\%12
matriceossII=(matriceossI+3)\%\%12
matriceossJJ=(matriceossJ+0)\%\%12
matriceossKK=(matriceossK+3)\%\%12
matriceossLL=(matriceossL+9)\%\%12
matriceossMM=(matriceossM+0)\%\%12
matriceossNN=(matriceossN+1)\%\%12
matriceossOO=(matriceossO+9)\%\%12
matriceossPP=(matriceossP+11)\%\%12

seqdef(matriceossAA[,2])->SEQ1
seqdef(matriceossAA[,3])->SEQ2
seqdef(matriceossAA[,4])->SEQ3
seqdef(matriceossAA[,5])->SEQ4
seqdef(matriceossAA[,6])->SEQ5
seqdef(matriceossAA[,7])->SEQ6
seqdef(matriceossAA[,8])->SEQ7
seqdef(matriceossAA[,9])->SEQ8
#
H<-build_hmm(observations= list(SEQ1,SEQ2,SEQ3,
SEQ4,SEQ5,SEQ6,SEQ7,SEQ8), 6,n_channels(8))
#
fittedAA=fit_model(H,em_step=F, global_step= T,
local_step=F,threads=4,control_global(maxeval=10^10))
seqdef(matriceossBB[,2])->SEQ1
seqdef(matriceossBB[,3])->SEQ2
seqdef(matriceossBB[,4])->SEQ3
seqdef(matriceossBB[,5])->SEQ4
seqdef(matriceossBB[,6])->SEQ5
seqdef(matriceossBB[,7])->SEQ6
seqdef(matriceossBB[,8])->SEQ7
seqdef(matriceossBB[,9])->SEQ8

```

```

seqdef(matriceossBB[,10])->SEQ9
seqdef(matriceossBB[,11])->SEQ10
seqdef(matriceossBB[,12])->SEQ11
seqdef(matriceossBB[,13])->SEQ12
seqdef(matriceossBB[,14])->SEQ13
seqdef(matriceossBB[,15])->SEQ14
seqdef(matriceossBB[,16])->SEQ15
seqdef(matriceossBB[,17])->SEQ16
seqdef(matriceossBB[,18])->SEQ17

H<-build_hmm(observations = list(SEQ1,SEQ2,SEQ3,
SEQ4,SEQ5,SEQ6,SEQ7,SEQ8,SEQ9,SEQ10
,SEQ11, SEQ12,SEQ13,SEQ14,SEQ15,SEQ16,
SEQ17), 6,n_channels(17))
fittedBB=fit_model(H, em_step = F, global_step = T,
local_step=F,threads=4,control_global(maxeval=10^4))

seqdef(matriceossCC[,2])->SEQ1
seqdef(matriceossCC[,3])->SEQ2
seqdef(matriceossCC[,4])->SEQ3
#
H<-build_hmm(observations = list(SEQ1, SEQ2, SEQ3),
1,n_channels(3))
#stima
fittedCC<-fit_model(H, em_step = F, global_step = T,
local_step = F, threads = 4)

seqdef(matriceossDD[,2])->SEQ1
seqdef(matriceossDD[,3])->SEQ2
seqdef(matriceossDD[,4])->SEQ3
H<-build_hmm(observations=list(SEQ1, SEQ2, SEQ3),1,
n_channels(3))
fittedDD<-fit_model(H,em_step= F,global_step=T,
local_step = F,threads = 4)

seqdef(matriceossEE[,2])->SEQ1
seqdef(matriceossEE[,3])->SEQ2
seqdef(matriceossEE[,4])->SEQ3
H<-build_hmm(observations=list(SEQ1, SEQ2, SEQ3),1,
n_channels(3))
fittedEE=fit_model(H,threads=4,em_step=F,global_step=T
, local_step = F)
[...]
```



```
#cos`i anche per gli altri brani

#controllo che gli algoritmi non si siano fermati in
#massimi locali, confrontando il valore della
#log-verosimiglianza massimizzata
c(fittedA$logLik, fittedAA$logLik, fittedB$logLik,
  fittedBB$logLik, fittedC$logLik, fittedCC$logLik,
  fittedD$logLik, fittedDD$logLik, fittedE$logLik,
  fittedEE$logLik, fittedG$logLik, fittedGG$logLik,
  fittedH$logLik, fittedHH$logLik , fittedI$logLik,
  fittedII$logLik, fittedJ$logLik, fittedJJ$logLik,
  fittedK$logLik , fittedKK$logLik, fittedL$logLik,
  fittedLL$logLik, fittedM$logLik, fittedMM$logLik,
  fittedN$logLik, fittedNN$logLik, fittedO$logLik,
  fittedOO$logLik, fittedP$logLik , fittedPP$logLik)
```

Divergenza di Kullback-Leibler e scaling multidimensionale

```
#import delle librerie
library(plot3D)

#Funzione che calcola via simulazione
#la divergenza per coppia di modelli
KL.SIMUL= function(B,MP,MQ,BESTP,BESTQ,LEN) {
  #B REPLICAZIONI
  #MP e MQ = MODELLI STIMATI,
  # BESTP BESTQ= stati nascosti "ideali" per l' AIC
  areaKL=0 #inizializzo il valore
  for (i in 1:B) {
    seed=trunc(runif(1,1,1000000))
    #per ogni replicazione fisso un seme,
    #infatti per ottenere sequenze multivariate
    #devo generare tante sequenze univariate
    #ma la sequenza di stati nascosti deve essere
    #la medesima
    channelP=MP$n_channels #seq parallele del modelloP
    channelQ=MQ$n_channels #seq parallele del modelloQ

    transP=MP$transition_probs #matr di transiz modelloP
    transQ=MQ$transition_probs #matr di transiz modelloQ
    iniP=MP$initial_probs #vettore di prob iniziali modelloP
    iniQ=MQ$initial_probs #vettore di prob iniziali modelloQ
    lista=list(NA) #conterrÃ le sequenze generate
```

```

for(j in 1:channelP){
  set.seed(seed)      #fissato il seme
  #utilizzo le diverse matrici di emissione
  #genero con la stessa sequenza di stati nascosti
  SP=simulate_hmm(n_sequences=1,transition_probs=transP,
  emission_probs=MP$emission_probs[j],initial_probs=iniP,
  sequence_length = LEN)
  SP=seqdef(as.numeric(unlist(as.matrix(SP$observations))))
  lista[[j]] <- SP #creo un oggetto "sequenza"
  #a partire dalle osservazioni generate
  }
#ricostruisco un modelloP,
#con i param. iniziali e le nuove osservazioni
simulP=build_hmm(lista,BESTP,transition_probs=transP,
emission_probs = MP$emission_probs[1:channelP],
initial_probs = iniP)
simulQ=build_hmm(lista,BESTQ,transition_probs=transQ,
emission_probs = MQ$emission_probs[1:channelQ],
initial_probs = iniQ)
llp=as.numeric(logLik(simulP, threads=4)-
(BESTP^2+channelP*BESTP))/(LEN*channelP)
#normalizzata per il n di osservazioni,
###soluzione naive
llq=as.numeric(logLik(simulQ,threads=4))/
(LEN*channelP)
###soluzione con penalit\`a per il n di parametri
# llq=as.numeric(logLik(simulQ,threads=4)-
#(BESTQ^2+BESTQ*channelP))/(LEN*channelP)

TMP= exp(llp)*(llp-llq)
areaKL=areaKL+TMP
#sommo il contributo alla per la simulazione corrente
print(i)
#stampo il numero dell'iterazione
}
areaKL=areaKL/B      #ottengo la media
return(areaKL)
}

ac=KL.SIMUL(10, fittedAA$model,fittedCC$model,6,1,LEN = 400)
ca=KL.SIMUL(10, fittedCC$model,fittedAA$model,1,6,LEN = 400)
ab=KL.SIMUL(10, fittedAA$model,fittedBB$model,6,6,LEN = 400)
ba=KL.SIMUL(10, fittedBB$model,fittedAA$model,6,6,LEN = 400)

```

```

ad=KL.SIMUL(10, fittedAA$model, fittedDD$model, 6, 1, LEN = 400)
da=KL.SIMUL(10, fittedDD$model, fittedAA$model, 1, 6, LEN = 400)
ae=KL.SIMUL(10, fittedAA$model, fittedEE$model, 6, 1, LEN = 400)
ea=KL.SIMUL(10, fittedEE$model, fittedAA$model, 1, 6, LEN = 400)
ag=KL.SIMUL(10, fittedAA$model, fittedGG$model, 6, 5, LEN = 400)
ga=KL.SIMUL(10, fittedGG$model, fittedAA$model, 5, 6, LEN = 400)

ah=KL.SIMUL(10, fittedAA$model, fittedHH$model, 6, 1, LEN = 400)
ha=KL.SIMUL(10, fittedHH$model, fittedAA$model, 1, 6, LEN = 400)
ai=KL.SIMUL(10, fittedAA$model, fittedII$model, 6, 9, LEN = 400)
ia=KL.SIMUL(10, fittedII$model, fittedAA$model, 9, 6, LEN = 400)
aj=KL.SIMUL(10, fittedAA$model, fittedJJ$model, 6, 11, LEN= 400)
ja=KL.SIMUL(10, fittedJJ$model, fittedAA$model, 11, 6, LEN= 400)
ak=KL.SIMUL(10, fittedAA$model, fittedJJ$model, 6, 3, LEN = 400)
ka=KL.SIMUL(10, fittedJJ$model, fittedAA$model, 3, 6, LEN = 400)
al=KL.SIMUL(10, fittedLL$model, fittedBB$model, 6, 1, LEN = 400)
la=KL.SIMUL(10, fittedAA$model, fittedLL$model, 1, 6, LEN = 400)
am=KL.SIMUL(10, fittedAA$model, fittedMM$model, 6, 1, LEN = 400)
ma=KL.SIMUL(10, fittedMM$model, fittedAA$model, 1, 6, LEN = 400)
an=KL.SIMUL(10, fittedAA$model, fittedNN$model, 6, 2, LEN = 400)
na=KL.SIMUL(10, fittedNN$model, fittedAA$model, 2, 6, LEN = 400)
ao=KL.SIMUL(10, fittedAA$model, fittedOO$model, 6, 2, LEN = 400)
oa=KL.SIMUL(10, fittedOO$model, fittedAA$model, 2, 6, LEN = 400)
ap=KL.SIMUL(10, fittedAA$model, fittedPP$model, 6, 1, LEN = 400)
pa=KL.SIMUL(10, fittedPP$model, fittedAA$model, 1, 6, LEN = 400)

bc=KL.SIMUL(10, fittedBB$model, fittedCC$model, 6, 1, LEN = 400)
cb=KL.SIMUL(10, fittedCC$model, fittedBB$model, 1, 6, LEN = 400)
bd=KL.SIMUL(10, fittedBB$model, fittedDD$model, 6, 1, LEN = 400)
db=KL.SIMUL(10, fittedDD$model, fittedBB$model, 1, 6, LEN = 400)
be=KL.SIMUL(10, fittedBB$model, fittedEE$model, 6, 1, LEN = 400)
eb=KL.SIMUL(10, fittedEE$model, fittedBB$model, 1, 6, LEN = 400)
bg=KL.SIMUL(10, fittedBB$model, fittedGG$model, 6, 5, LEN = 400)
gb=KL.SIMUL(10, fittedGG$model, fittedBB$model, 5, 6, LEN = 400)
bh=KL.SIMUL(10, fittedBB$model, fittedHH$model, 6, 1, LEN = 400)
hb=KL.SIMUL(10, fittedHH$model, fittedBB$model, 1, 6, LEN = 400)
bi=KL.SIMUL(10, fittedBB$model, fittedII$model, 6, 9, LEN = 400)
ib=KL.SIMUL(10, fittedII$model, fittedBB$model, 9, 6, LEN = 400)
bj=KL.SIMUL(10, fittedBB$model, fittedJJ$model, 6, 11, LEN= 400)
jb=KL.SIMUL(10, fittedJJ$model, fittedBB$model, 11, 6, LEN= 400)
bk=KL.SIMUL(10, fittedBB$model, fittedKK$model, 6, 3, LEN = 400)
kb=KL.SIMUL(10, fittedKK$model, fittedBB$model, 3, 6, LEN = 400)
bl=KL.SIMUL(10, fittedBB$model, fittedLL$model, 6, 1, LEN = 400)

```

```

lb=KL.SIMUL(10, fittedLL$model, fittedBB$model, 1, 6, LEN = 400)
bm=KL.SIMUL(10, fittedBB$model, fittedMM$model, 6, 1, LEN = 400)
mb=KL.SIMUL(10, fittedMM$model, fittedBB$model, 1, 6, LEN = 400)
bn=KL.SIMUL(10, fittedBB$model, fittedNN$model, 6, 2, LEN = 400)
nb=KL.SIMUL(10, fittedNN$model, fittedBB$model, 2, 6, LEN = 400)
bo=KL.SIMUL(10, fittedBB$model, fittedOO$model, 6, 2, LEN = 400)
ob=KL.SIMUL(10, fittedOO$model, fittedBB$model, 2, 6, LEN = 400)
bp=KL.SIMUL(10, fittedBB$model, fittedPP$model, 6, 1, LEN = 400)
pb=KL.SIMUL(10, fittedPP$model, fittedBB$model, 1, 6, LEN = 400)

cd=KL.SIMUL(10, fittedCC$model, fittedDD$model, 1, 1, LEN = 400)
dc=KL.SIMUL(10, fittedDD$model, fittedCC$model, 1, 1, LEN = 400)
ce=KL.SIMUL(10, fittedCC$model, fittedEE$model, 1, 1, LEN = 400)
ec=KL.SIMUL(10, fittedEE$model, fittedCC$model, 1, 1, LEN = 400)

[...] #per tutte le combinazioni

lm=KL.SIMUL(10, fittedLL$model, fittedMM$model, 1, 1, LEN = 400)
ml=KL.SIMUL(10, fittedMM$model, fittedLL$model, 1, 1, LEN = 400)
ln=KL.SIMUL(10, fittedLL$model, fittedNN$model, 1, 2, LEN = 400)
nl=KL.SIMUL(10, fittedNN$model, fittedLL$model, 2, 1, LEN = 400)
lo=KL.SIMUL(10, fittedLL$model, fittedOO$model, 1, 2, LEN = 400)
ol=KL.SIMUL(10, fittedNN$model, fittedLL$model, 2, 1, LEN = 400)
lp=KL.SIMUL(10, fittedLL$model, fittedPP$model, 1, 1, LEN = 400)
pl=KL.SIMUL(10, fittedPP$model, fittedLL$model, 1, 1, LEN = 400)

mn=KL.SIMUL(10, fittedMM$model, fittedNN$model, 1, 2, LEN = 400)
nm=KL.SIMUL(10, fittedNN$model, fittedMM$model, 2, 1, LEN = 400)
mo=KL.SIMUL(10, fittedMM$model, fittedOO$model, 1, 2, LEN = 400)
om=KL.SIMUL(10, fittedOO$model, fittedMM$model, 2, 1, LEN = 400)
mp=KL.SIMUL(10, fittedMM$model, fittedPP$model, 1, 2, LEN = 400)
pm=KL.SIMUL(10, fittedPP$model, fittedMM$model, 2, 1, LEN = 400)

no=KL.SIMUL(10, fittedNN$model, fittedOO$model, 2, 2, LEN = 400)
on=KL.SIMUL(10, fittedOO$model, fittedNN$model, 2, 2, LEN = 400)
np=KL.SIMUL(10, fittedNN$model, fittedPP$model, 2, 1, LEN = 400)
pn=KL.SIMUL(10, fittedPP$model, fittedNN$model, 1, 2, LEN = 400)

op=KL.SIMUL(10, fittedOO$model, fittedPP$model, 2, 1, LEN = 400)
po=KL.SIMUL(10, fittedPP$model, fittedOO$model, 1, 2, LEN = 400)

#Matrice delle distanze
w0=0

```

```

matricetutte=
matrix(c(w0,ab,ac,ad,ae,ag,ah,ai,aj,ak,al,am,an,ao,ap,
        ba,w0,bc,bd,be,bg,bh,bi,bj,bk,bl,bm,bn,bo,bp,
        ca,cb,w0,cd,ce,cg,ch,ci,cj,ck,cl,cm,cn,co,cp,
        da,db,dc,w0,de,dg,dh,di,dj,dk,dl,dm,dn,do,dp,
        ea,eb,ec,ed,w0,eg,eh,ei,ej,ek,el,em,en,eo,ep,
        ga,gb,gc,gd,ge,w0,gh,gi,gj,gk,gl,gm,gn,go,gp,
        ha,hb,hc,hd,he,hg,w0,hi,hj,hk,hl,hm,hn,ho,hp,
        ia,ib,ic,id,ie,ig,ih,w0,ij,ik,il,im,inN,io,ip,
        ja,jb,jc,jd,je,jg,jh,ji,w0,jk,jl,jm,jn,jo,jp,
        ka,kb,kc,kd,ke,kg,kh,ki,kj,w0,kl,km,kn,ko,kp,
        la,lb,lc,ld,le,lg,lh,li,lj,lk,w0,lm,ln,lo,lp,
        ma,mb,mc,md,me,mg,mh,mi,mj,mk,ml,w0,mn,mo,mp,
        na,nb,nc,nd,ne,ng,nh,ni,nj,nk,nl,nm,w0,no,np,
        oa,ob,oc,od,oe,og,oh,oi,oj,ok,ol,om,on,w0,op,
        pa,pb,pc,pd,pe,pg,ph,pi,pj,pk,pl,pm,po,w0),
        nrow=15, byrow = TRUE)
MATRICEKLTUTTE=(t(abs(matricetutte))+abs(matricetutte))/2
#Matrice delle distanze (simmetrica)
#Heatmap
heatmap(MATRICEKLTUTTE, labRow = c(1,5,2,3,4,
6,7,8,9,10,11,12,13,14,15), labCol = c(1,5,2,3,4,
6,7,8,9,10,11,12,13,14,15))

#Scaling e grafici
mds=cmdscale(as.matrix(MATRICEKLTUTTE), eig=TRUE)
mds$GOF

labels=c("BACH","TCHAIKOVSKY","MOZART3","MOZART1",
"MOZART2","BACH","SCHOENBERG","SCHUBERT",
"SCHUBERT","SCHUMANN","CHOPIN","CHOPIN",
"DEBUSSY","DEBUSSY","BARTOK")
scatter2D(mds$points[,1],mds$points[,2],xlim=c(-0.012,0.01i)
,colvar=c(1,3.23,1.662,1.662,1.662,1,3.874,2.853,
2.853,2.99,3.1853,3.1853,3.514,3.514,4),
col = ramp.col(c("blue", "red", "yellow")),
cex=1.2,bty = "g", add = FALSE, pch=15)
text2D(mds$points[,1], mds$points[,2], labels,
colvar = NULL, col = NULL, add = TRUE, cex=0.68)
mds3d=cmdscale(as.matrix(MATRICEKLTUTTE),k=3, eig=TRUE)
mds3d$GOF
scatter3D(theta = 20, phi = 11.2,mds3d$points[,1],
mds3d$points[,2],mds3d$points[,3],

```

```
colvar=c(1,3.23,1.662,1.662,1.662,1,3.874,2.853,
2.853,2.99,3.1853,3.1853,3.514,3.514,4),
col = ramp.col(c("blue", "red", "yellow")),
cex=1,bty = "g", add = FALSE, pch=15)
text3D(theta=20, phi=11.2,mds3d$points[,1],
mds3d$points[,2],mds3d$points[,3], labels,
colvar = NULL,cex=0.8, add = TRUE)
```

Simulazioni e analisi con t-SNE

```
\small
#import delle librerie
library(digest)
library(tsne)
library(scatterplot3d)
source('http://www.sthda.com/sthda/RDoc/
functions/addgrids3d.r')

#Simulazioni delle sequenze degli stati nascosti
#dal brano ``Cantata BWV 884'' di Bach
simA=simulate_hmm(10,
  initial_probs=fittedAA$model$initial_probs,
  transition_probs = fittedAA$model$transition_probs,
  emission_probs = fittedAA$model$emission_probs,
  sequence_length = 400)
simA=(as.vector(unlist(simA$states)))
for (i in 1:length(simA)){
  if (simA[i]=="State 1") simA[i]=1
  if (simA[i]=="State 2") simA[i]=2
  if (simA[i]=="State 3") simA[i]=3
  if (simA[i]=="State 4") simA[i]=4
  if (simA[i]=="State 5") simA[i]=5
  if (simA[i]=="State 6") simA[i]=6
  if (simA[i]=="State 7") simA[i]=7
  if (simA[i]=="State 8") simA[i]=8
  if (simA[i]=="State 9") simA[i]=9
  if (simA[i]=="State 10") simA[i]=10
  if (simA[i]=="State 11") simA[i]=11
  if (simA[i]=="State 12") simA[i]=12
}
sa1=as.numeric(simA[1:400])
sa2=as.numeric(simA[401:800])
sa3=as.numeric(simA[801:1200])
```

```

sa4=as.numeric(simA[1201:1600])
sa5=as.numeric(simA[1601:2000])
sa6=as.numeric(simA[2001:2400])
sa7=as.numeric(simA[2401:2800])
sa8=as.numeric(simA[2801:3200])
sa9=as.numeric(simA[3201:3600])
sa10=as.numeric(simA[3601:4000])

#B Simulazioni delle sequenze degli stati nascosti
#dal brano ``Danza Russa'' di Tchaikovsky
simB=simulate_hmm(10,
  initial_probs = fittedBB$model$initial_probs,
  transition_probs = fittedBB$model$transition_probs,
  emission_probs = fittedBB$model$emission_probs,
  sequence_length = 400)
simB=(as.vector(unlist(simB$states)))
for (i in 1:length(simB)){
  if (simB[i]=="State 1") simB[i]=1
  if (simB[i]=="State 2") simB[i]=2
  if (simB[i]=="State 3") simB[i]=3
  if (simB[i]=="State 4") simB[i]=4
  if (simB[i]=="State 5") simB[i]=5
  if (simB[i]=="State 6") simB[i]=6
  if (simB[i]=="State 7") simB[i]=7
  if (simB[i]=="State 8") simB[i]=8
  if (simB[i]=="State 9") simB[i]=9
  if (simB[i]=="State 10") simB[i]=10
  if (simB[i]=="State 11") simB[i]=11
  if (simB[i]=="State 12") simB[i]=12
}
sb1=as.numeric(simB[1:400])
sb2=as.numeric(simB[401:800])
sb3=as.numeric(simB[801:1200])
sb4=as.numeric(simB[1201:1600])
sb5=as.numeric(simB[1601:2000])
sb6=as.numeric(simB[2001:2400])
sb7=as.numeric(simB[2401:2800])
sb8=as.numeric(simB[2801:3200])
sb9=as.numeric(simB[3201:3600])
sb10=as.numeric(simB[3601:4000])

##Simulazioni delle sequenze degli stati nascosti
#dal brano ``Pseudo Mozart 3''

```

```

simC=....

[...]

#Creazione di una matrice con tutte le sequenze
x=as.data.frame(rbind(
  sa1,sa2,sa3,sa4,sa5,sa6,sa7,sa8,sa9,sa10,
  sb1,sb2,sb3,sb4,sb5,sb6,sb7,sb8,sb9,sb10,
  sc1,sc2,sc3,sc4,sc5,sc6,sc7,sc8,sc9,sc10,
  sd1,sd2,sd3,sd4,sd5,sd6,sd7,sd8,sd9,sd10,
  se1,se2,se3,se4,se5,se6,se7,se8,se9,se10,
  sg1,sg2,sg3,sg4,sg5,sg6,sg7,sg8,sg9,sg10,
  sh1,sh2,sh3,sh4,sh5,sh6,sh7,sh8,sh9,sh10,
  si1,si2,si3,si4,si5,si6,si7,si8,si9,si10,
  sj1,sj2,sj3,sj4,sj5,sj6,sj7,sj8,sj9,sj10,
  sk1,sk2,sk3,sk4,sk5,sk6,sk7,sk8,sk9,sk10,
  sl1,sl2,sl3,sl4,sl5,sl6,sl7,sl8,sl9,sl10,
  sm1,sm2,sm3,sm4,sm5,sm6,sm7,sm8,sm9,sm10,
  sn1,sn2,sn3,sn4,sn5,sn6,sn7,sn8,sn9,sn10,
  so1,so2,so3,so4,so5,so6,so7,so8,so9,so10,
  sp1,sp2,sp3,sp4,sp5,sp6,sp7,sp8,sp9,sp10))
x=(as.matrix(x))

tsnex=tsne(x,k = 2,whiten = F)
tsnex3=tsne(x,k = 3,whiten = F)
plot(tsnex,cex=1.299, col=c(rep(1,10),
  rep(2,10), rep(3,10),rep(4,10),rep(5,10),
  rep(6,10), rep(7,10),rep(8,10), rep(9,10),
  rep(10,10), rep(11,10),rep(12,10),
  rep(13,10), rep(14,10),rep(15,10)),
  pch=c(rep("a",10),rep("b",10),
  rep("c",10),rep("d",10),rep("e",10),
  rep("g",10), rep("h",10),rep("i",10),
  rep("j",10),rep("k",10),rep("l",10),
  rep("m",10),rep("n",10),rep("o",10),rep("p",10)))
grid()
scatterplot3d(tsnex3, grid = F,box =F ,type = "h",
  lty.axis = 1,xlab = "",ylab="",zlab="",
  cex.symbols = 0.82, angle=80,
#colori 1 preclassico 2 classico 3 romantico 4 post
  pch=c(rep("a",10),rep("b",10),
  rep("c",10), rep("d",10),rep("e",10),
  rep("g",10), rep("h",10),rep("i",10),

```



```
rep("j",10),rep("k",10),rep("l",10),  
rep("m",10),rep("n",10),rep("o",10),  
rep("p"10)), color=c(rep(1,10),  
rep(2,10),rep(3,10), rep(3,10),  
rep(3,10), rep(1,10), rep(4,10),  
rep(5,10), rep(5,10), rep(6,10),  
rep(7,10),rep(7,10), rep(8,10),  
rep(8,10), rep(9,10))  
source('http://www.sthda.com/sthda/RDoc/  
functions/addgrids3d.r')  
addgrids3d(tsnex3,grid=c("xy","xz","yz"),  
lty.grid=9,angle=80)
```


Bibliografia

- [1] I. Bent, W. Drabkin, and C. Annibaldi. *Analisi musicale*. Biblioteca di cultura musicale: I Manuali. EDT, 1990.
- [2] J. Beran. *Statistics in Musicology*. Taylor & Francis, 2003.
- [3] Wei Chai and Barry Vercoe. Folk music classification using hidden markov models. 04 2018.
- [4] Tom Collins, Robin Laney, Alistair Willis, and Paul H. Garthwaite. Chopin, mazurkas and markov. *Significance*, 8(4):154–159.
- [5] Buttarazzi Davide, Pandolfo Giuseppe, and Porzio Giovanni C. A boxplot for circular data. *Biometrics*, 0(0).
- [6] B. Everitt and T. Hothorn. *An Introduction to Applied Multivariate Analysis with R*. Use R! Springer New York, 2011.
- [7] Markus Falkhausen, Herbert Reininger, and Dietrich Wolf. Calculation of distance measures between hidden markov models. 1995, 06 1996.
- [8] Jouni Helske and Satu Helske. *Mixture hidden Markov models for sequence data: the seqHMM package in R*, 2017. R package version 1.0.8.
- [9] Satu Helske and Jouni Helske. Mixture hidden Markov models for sequence data: the seqHMM package in R. *Accepted to Journal of Statistical Software*, 2017. Available at ArXiv.
- [10] B.H Juang and Lawrence Rabiner. A probabilistic distance measure for hidden markov models. 64, 02 1985.
- [11] Uwe Ligges, Sebastian Krey, Olaf Mersmann, and Sarah Schnackenberg. *tuneR: Analysis of music*, 2016.
- [12] Altman Rachel MacKay and Petkau A. John. Application of hidden markov models to multiple sclerosis lesion count data. *Statistics in Medicine*, 24(15):2335–2344.

- [13] Minh Nguyen, Sanjay Purushotham, Hien To, and Cyrus Shahabi. m-tsne: A framework for visualizing high-dimensional multivariate time series. *CoRR*, abs/1708.07942, 2017.
- [14] Rolf Turner. Direct maximization of the likelihood of a hidden markov model. 52:4147–4160, 05 2008.
- [15] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [16] C. Weihs, D. Jannach, I. Vatulkin, and G. Rudolph. *Music Data Analysis: Foundations and Applications*. Chapman & Hall/CRC Computer Science & Data Analysis. CRC Press, 2016.
- [17] I. Xenakis. *Musique formelles: nouveaux principes formels de composition musicale*. Revue musicale. Richard-Masse, 1963.

Ringraziamenti

Ringrazio la Professoressa Brazzale per il tempo dedicatomi, per avermi dato l'opportunità di sviluppare questo tema, di conoscere e sperimentare nuovi metodi. Un grazie va anche al Professor Finesso, per la disponibilità e per gli esperti pareri.